# Revisiting the Security of Approximate FHE with Noise-Flooding Countermeasures

Flavio Bergamaschi 1, Anamaria Costache 2\*, Dana Dachman-Soled 3\*\*, Hunter Kippen 3\* \* \*, Lucas LaBuff 3, and Rui Tang 3

Abstract. Approximate fully homomorphic encryption (FHE) schemes, such as the CKKS scheme (Cheon, Kim, Kim, Song, ASIACRYPT '17), are among the leading schemes in terms of efficiency and are particularly suitable for Machine Learning (ML) tasks. Although efficient, approximate FHE schemes have some inherent risks: Li and Micciancio (EUROCRYPT '21) demonstrated that while these schemes achieved the standard notion of CPA-security, they failed against a variant, IND-CPA<sup>D</sup>, in which the adversary is given limited access to the decryption oracle. Subsequently, Li, Micciancio, Schultz, and Sorrell (CRYPTO '22) proved that with noise-flooding countermeasures which add Gaussian noise of sufficiently high variance before outputting the decrypted value, the CKKS scheme is secure. However, the variance required for provable security is very high, inducing a large loss in message precision. In this work, we consider a broad class of attacks on CKKS with noise-flooding countermeasures, which we call "semi-honest" attacks, in which an adversary may submit only correctly distributed ciphertexts to the decryption oracle. The ciphertexts submitted for decryption can be fresh ciphertexts, or can be ciphertexts resulting from the homomorphic evaluation of some circuit on fresh and independent ciphertexts. Our motivation is to model an internal threat scenario where an adversary can passively access the internal randomness of the system.

We analyze the concrete security of CKKS with various levels of noise-flooding in the face of such attacks. The aim of this work is to outline and precisely quantify the various trade-offs between the number of allowed decryptions before refreshing the keys, noise-flooding levels, and the concrete security of the scheme after a number of decryptions have been observed by the adversary.

Due to the large dimension and modulus in typical FHE parameter sets, previous techniques even for estimating the concrete runtime of such attacks – such as those in (Dachman-Soled, Ducas, Gong, Rossi, CRYPTO '20) – become computationally infeasible, since they involve high dimensional and high precision matrix multiplication and inversion. We therefore develop new techniques that allow us to perform fast security estimation, even for FHE-size parameter sets.

<sup>\*</sup> Supported in part by Intel through the Intel Labs Soteria Research Collaboration

<sup>\*\*</sup> Supported in part by NSF grant #CNS-2154705 and by Intel through the Intel Labs Crypto Frontiers Research Center.

<sup>\*\*\*</sup> Supported in part by the Clark Doctoral Fellowship from the Clark School of Engineering, University of Maryland, College Park.

## 1 Introduction

The notion of "approximate FHE" – fully homomorphic encryption schemes that guarantee only approximate correctness of decryption – was proposed by Cheon, Kim, Kim, and Song [12]. Their scheme, henceforth referred to as CKKS, is one of the leading schemes in terms of efficiency and Single Instruction/Multiple Data (SIMD) parallelization opportunities, and is particularly suitable for Machine Learning (ML) tasks. Although efficient, approximate FHE schemes have some inherent risks: Li and Micciancio [25] demonstrated what while these schemes achieved the standard notion of CPA-security, they failed against a variant, IND-CPA $^D$ , in which the adversary is given limited access to the decryption oracle. In the same work [25], the authors showed that for exact schemes (such as BGV, BFV and TFHE), the notions of IND-CPA $^D$  and IND-CPA are equivalent.

Noise-flooding techniques have been suggested as a practical countermeasure against  $\mathsf{IND-CPA}^D$  attacks [1]. These techniques add noise (from a Gaussian distribution) to the message obtained by decrypting a ciphertext, before it is returned to the adversary. Such countermeasures were formally analyzed in the work of Li, Micciancio, Schultz, and Sorrell [26], and it was shown that when the noise-flooding level is sufficiently high, they are indeed provably secure. Nevertheless, the amount of noise required for provable  $\mathsf{IND-CPA}^D$  security remains quite high, and as a result, severely limits the message precision that CKKS can handle (8 or 16 bits of precision for parameter sets deemed "reasonable").

There are two main reasons for the large noise required for provable IND-CPA $^D$  security. First, a worst-case noise analysis is needed to determine the amount of noise already present in a ciphertext prior to decryption and the noise flooding must then scale with this worst-case noise. The reason is that average-case noise analysis assumes that input ciphertexts to homomorphic circuits are independent and identically distributed. When a circuit computation is performed on correlated inputs (a simple example is adding a ciphertext to itself  $\ell$  times instead of adding  $\ell$  independent ciphertexts), the average-case analysis will fail to output the correct noise estimation. This particular correlated input attack has been exploited in [21]. A similar attack exploiting decryption failures in exact FHE schemes was presented in [10]. Second, the current techniques for proving IND-CPA $^D$  show that the decryptions obtained in the two games of the indistinguishability definition are statistically close, whereas computational indistinguishability would be sufficient to achieve the security notion.

In this work, we introduce a formal security model that captures *semi-honest attackers with access to a decryption oracle*. As a result of enforcing semi-honest behavior, *average-case noise analysis* is sufficient to accurately estimate the noise present in a ciphertext submitted for decryption, so the noise-flooding level can scale with the *average-case* noise, as opposed to the worst-case noise. We then investigate the *concrete computational security* achieved when decryption is augmented with different levels of noise flooding. In particular, we investigate the concrete runtime and success probability of the state-of-the-art key recovery attacks when incorporating the additional information obtained from decryption. Thus, our goal is to provide insights into the concrete security of CKKS with various noise-flooding levels in the semi-honest setting.

We consider internal threats, which are common in deployment scenarios, and argue that it is conceivable that adversaries can access the internal randomness of the system. Our new model, which we call IND-CPA<sup>DSH</sup> (see Section 5 for the formal security definition), captures semi-honest attackers who do not hold the secret key, but may access the decryption oracle to obtain noisy decryptions. In particular, our model enforces that public keys and ciphertexts are created honestly, and that fresh and evaluated ciphertexts follow the "correct" average-case noise distributions. Our model only allows the adversary to request evaluations on classes of admissible circuits. Our model captures an adversary who passively corrupts a party within the system, and thus can observe their entire state, and is incomparable to IND-CPA<sup>D</sup>. On the

 $<sup>^4</sup>$  In a recent work [10], this is called into question, as the authors point out that the proof of equivalence between IND-CPA $^D$  and IND-CPA does not take into account the decryption failure probability of an exact scheme. The authors of [10] exploit the fact that this decryption failure probability is rather high in implementations of exact schemes to run an IND-CPA $^D$  attack on the BFV scheme, and remark that their attack also applies to BGV and TFHF

<sup>&</sup>lt;sup>5</sup> In this work, admissible circuits will correspond to either identity, Class 1 or Class 2 circuits and will be defined subsequently.

one hand, it is stronger since the adversary gets to observe the internal randomness of the corrupted party. On the other hand, our model is weaker since it enforces that the noise distribution of ciphertexts submitted for decryption follows the expected distribution for the corresponding class of admissible circuits.

One may ask why we disallow the adversary from querying the decryption oracle with malformed or incorrectly distributed ciphertexts. Indeed, when such queries are allowed, IND-CPA<sup>D</sup> attacks against average-case noise-flooding techniques are known [21,10], as mentioned above. We note that in such attacks the attacker actively modifies the distribution of ciphertexts to be decrypted so that it deviates from the correct distribution. At the same time, since FHE schemes (even exact schemes [10]) are inherently insecure against CCA attacks, external measures must be put in place in any particular deployment to ensure that an adversary cannot request decryptions of incorrectly distributed ciphertexts. For example, to ensure the integrity of the computation, as well as the well-formedness of the ciphertexts and relevant keys, Verifiable Computation (VC), and Zero-Knowledge (ZK) proofs for FHE schemes or even the use of enclaves should be considered [8,18,19]. The implementation of these measures are outside the scope of our work; we simply observe that if such measures are in place, then adversaries are restricted to be semi-honest as described above.

Once we have established our security model, we investigate the best attacks in this model, for various levels of noise flooding. We consider the concrete security degradation of the CKKS scheme in the presence of t decryptions, with noise flooding of some variance  $\rho^2$ . Our starting point is noise-flooding equal to the noise level computed by the "average-case" noise analysis. Here, the decryption of a ciphertext is noise-flooded by the variance of the noise already present in the ciphertext<sup>6</sup>. This noise-flooding setting is optimal, in the sense that only 1 bit of message precision is lost. On the other side of the spectrum is setting  $\rho^2$  as large as the variance needed for provable, statistical security. We investigate settings of  $\rho^2$  that fall between these two extremes. Our aim is to present tradeoffs among (1) the number of allowed decryptions before the secret/public key must be refreshed, (2) the variance of the noise-flooding added to the decryption (which determines the loss of precision), and (3) the concrete security of the scheme after a number of decryptions have been observed by the adversary (e.g. a drop of 10 or 15 bits in security for a 256-bit parameter set may still be acceptable). We stress that the aim of our work is not to provide any definite conclusion on the concrete level of noise-flooding to apply when deploying CKKS. The conclusions of our experiments should rather be viewed as informing choices such as choosing parameter sizes and key refreshing policies.

In Section 2, we give a technical overview that includes a description of the classes of admissible circuits we consider, as well as our techniques for obtaining concrete hardness estimates. We emphasize that prior methods for concrete hardness estimation that are applicable in this setting, such as [16], require performing expensive matrix operations on the covariance matrix representing the conditional distribution of the LWE secret/error. For FHE-scale parameter sets, the covariance matrix can have dimension as high as  $256K \times 256K$ , so several hundred terabytes are required to naively store the values (assuming 64-bit precision, whereas in our experimental results in Section 10.2, we find that up to 2,000 bit precision is required for meaningful results). Therefore, one of our main technical contributions is developing new tools to provide fast and accurate estimates that do not require these high-dimensional matrix operations.

## 2 Technical Overview and Related Work

We consider three types of admissible circuits: *Identity, Class 1*, and *Class 2* circuits, which are defined below. For each type of circuit, we consider an adversary who requests t evaluations of circuits of this type on fresh ciphertexts, and then obtains t noisy decryptions of these evaluations. Importantly, for each circuit type, the information obtained by the adversary from decryption will correspond to a noisy linear system of equations on the LWE secret and error underlying the CKKS public key. This means that the view of the adversary is equivalent to obtaining the public key  $p\mathbf{k} = (-as + e, a) \pmod{q}$ , for some ciphertext modulus q, along with a multivariate Gaussian distribution  $\mathcal{N}(\mu', \Sigma')$  representing the joint *conditional* distribution on the secret and error (s, e).

<sup>&</sup>lt;sup>6</sup> As predicted by an average-case noise analysis [14].

#### 2.1 Admissible Circuits

Decryption Queries on Identity Circuits. We start by considering an attacker who submits a number t of fresh ciphertexts for decryption, or equivalently, requesting t decryptions of ciphertexts obtained from the evaluation of the identity circuit on a fresh encryption. The adversary receives the noise-flooded output of the decryption, where the noise is a centered Gaussian of some variance  $\rho^2$ . This is a natural circuit class to consider since in the original paper of Li and Micciancio [25], attacks using only identity circuits were shown to allow full key-recovery against CKKS when there is no noise added during decryption.

Decryption Queries on Class 1 and Class 2 Circuits. We then extend our analysis to broader classes of circuits, beyond identity circuits (see Section 9 for formal definitions of these classes). Briefly, Class 1 circuits are circuits that consist of  $\ell$  independent subcircuits  $C_1, \ldots, C_\ell$ . These circuits can be completely arbitrary as long as they all have the same multiplicative depth  $d \geq 1$  and they each end in a multiplication with rescale operation. The final circuit consists of the addition of the outputs of these subcircuits. Intuitively, we require addition of  $\ell$  ciphertexts so that the noise coefficients, which are individually uniformly random between [-0.5, 0.5], can be well-approximated by independent Gaussian distributions. Class 2 circuits are circuits whose output corresponds to the multiplication without rescale of the outputs of two independent Class 1 circuits. Our motivation for considering Class 2 circuits is that in practice, a rescale is typically not performed in the final multiplication gate of the circuit, in order to reduce the size of the top-level modulus. We recall the noise terms for a multiplication with and without rescale in Section 4. At a high level, the difference between the two is that the noise in the former is dominated by a rounding noise, whereas the latter contains more terms, including a quadratic equation in the secret decryption key.

For circuits in Class 1 and 2, we note that our attacker does not need to know the internal randomness used by the encryption process, and thus the attack is valid even in a weaker adversarial model. The analysis in this case is facilitated by the fact that it was shown in prior work [15,7] that after a **rescale** step, the rounding noise (which can be publicly computed) dominates the noise present in the ciphertext. We note that the same assumption was made in [26] and refer the reader to the paper for a further discussion. Upon decryption, the information obtained by the adversary corresponds to an approximate linear equation on the secret, which induces a conditional Gaussian distribution on the secret. Thus, the information obtained is in fact a special case of the information obtained by decryptions of the identity circuit, which corresponds to noisy linear equations on both the LWE secret and error.

#### 2.2 Key Recovery Attacks

We consider three types of key recovery attacks for each of the three classes of admissible circuits described above. The attacks reduce the instance observed by the adversary to a unique-SVP (u-SVP) instance, which is the *same* approach used to determine the FHE parameter sets for varying levels of bit security in the first place! Our analysis differs in that we determine the effect of incorporating the knowledge that the LWE secret and error are jointly distributed as the multivariate Gaussian distribution  $\mathcal{N}(\mu', \Sigma')$  (capturing the conditional distribution on the LWE secret and error) on the concrete runtime of these attacks.

Lattice Reduction Attacks. Here we assume that the adversary embeds the original LWE instance and the distribution  $\mathcal{N}(\mu', \Sigma')$  into a Distorted Bounded Distance Decoding (DBDD) instance (introduced by [16]). Specifically, the resulting DBDD instance will consist of a tuple  $(\Lambda, \mu', \Sigma')$ , where  $\Lambda$  is the lattice obtained by performing Kannan's embedding on the LWE instance  $pk = ([-as + e]_{q_L}, a)$  obtained from the CKKS public key (see Section 4.6 for more details). As shown by Dachman-Soled et al. [16], a DBDD instance can be reduced to a u-SVP instance, and solved using the state-of-the-art BKZ-algorithm. Using the terminology of [16], the information obtained by the adversary from decryption is denoted as "hints," and as discussed previously, these hints consist of noisy linear equations on the LWE secret/error, where the noise is sampled from a Gaussian distribution. Therefore, the conditional distribution on the LWE secret/error, given the hints, remains a Gaussian distribution and a closed-form formula for the new distribution can be obtained from known techniques. Thus, the steps to integrate the hints and transform the DBDD instance to a u-SVP instance follow those given in [16] for the case of conditional, full-dimensional, approximate hints. Upon obtaining the resulting u-SVP instance, the adversary then uses the BKZ algorithm to recover the shortest

vector which corresponds to the LWE secret/error. As shown in [16], the time required by the BKZ algorithm in terms of bikz (i.e. BKZ- $\beta$ ) to solve the final u-SVP instance, can be accurately estimated given only the volume and dimension of the final u-SVP instance.

Importantly, although the attack template proceeds as the one outlined in [16], our analysis of the attack runtime differs. To obtain concrete security estimates for the runtime via full-dimensional approximate hints as in [16], one would need to compute the determinant of a  $2n \times 2n$  dimensional matrix that depends on the t ciphertexts submitted for decryption and the outputs observed by the adversary. For n=256 and t=16, our experiments showed that this computation takes roughly a week on a supercomputer (See Section 10.2). In contrast, typical FHE parameters sets can have dimension up to  $\log_2(n)=17$ . Thus, to provide fast estimates, we analyze the distribution of the resulting  $2n \times 2n$  dimensional matrix arising from the outlined attack. We provide a closed-form expression for the expected determinant of a matrix drawn from this distribution (See Section 6 and Lemma 6.1). We verify experimentally (See Section 10.2) that the predicted and actual expected determinant match closely. Further, to the best of our knowledge, ours is the first concrete analysis to crucially take into account the ring structure of the full-dimensional approximate "hints" obtained by the adversary. We believe this type of analysis is a crucial component for allowing concrete hardness estimates for FHE-size parameters.

Guessing Attacks. Here the attacker keeps track of the conditional multivariate Gaussian distribution on the LWE secret/error after integrating the t hints. When the variance of individual secret/error coordinates becomes small enough, the adversary rounds the coordinate of the mean of the multivariate Gaussian distribution to the nearest integer. At some point, the adversary can guess n out of 2n coordinates correctly with high probability, in which case it can solve the original LWE system to obtain the remaining n coordinates. Similarly to the lattice reduction case, actually keeping track of the covariance matrix of the multivariate Gaussian distribution requires a  $2n \times 2n$  matrix inversion and is highly computationally intensive for FHE-scale parameters. Since we know the distribution of the matrix, we are able to derive bounds that hold with high probability on the trace and eigenvalues of the matrix, which in turn can be used to bound the success probability of the guessing attack, using the Gaussian correlation inequality [24] (See Section 7 and Lemma 7.1 for the case of identity circuits, and Section 9.3 for the case of Class 1 and Class 2 circuits).

Hybrid Attacks. Here the attacker guesses g < n number of coordinates as above, but cannot guess n of them with sufficiently high probability. The attacker integrates these g guesses as "perfect hints" into the DBDD instance and finally obtains a new u-SVP instance, which it then solves using lattice reduction. After integrating the guesses, the information known to the adversary corresponds to a principal submatrix of the covariance matrix, whose determinant we need to compute in order to estimate hardness. As before, we do not compute the actual  $2n \times 2n$  covariance matrix for the instance, which is highly computationally intensive, but rather use the fact that the distribution of the covariance matrix is known. We use the Eigenvalue Interlacing Theorem (see e.g. [22]) and bounds on the eigenvalues that hold w.h.p. in order to bound the determinant of the principal submatrix, given the determinant of the entire matrix (See Section 8 and Lemma 8.1 for the case of identity circuits, and Section 9.4 for the case of Class 1 and Class 2 circuits).

#### 2.3 Summary of Experimental Results

We performed extensive experimentation for a wide range of parameter sets proposed by the homomorphicencryption.org standards [2], as well as a larger parameter set with a ring dimension of  $\log_2 n = 17$  [27]. In Section 10, we provide experimental validation of Lemma 6.1, and in Appendix B we provide tables detailing the effectiveness of each of the three attack types on fresh ciphertexts (identity circuits) at various noise-flooding levels:  $\rho_{\rm circ}^2$ —the noise variance already present in a ciphertext— $100 \cdot \rho_{\rm circ}^2$ , and  $t \cdot \rho_{\rm circ}^2$ , where t is the number of decryptions the attacker may observe. See Figures 4, 5, 6. We present the data for the analogous experiments on Class 1 and 2 circuits in our supplementary material, see Appendix B Figures 7 - 12.

We note that our lemma statements involve complicated mathematical expressions for quantities such as the determinant or trace of the covariance matrix, and the implications for concrete security may not

<sup>&</sup>lt;sup>7</sup> Here circ denotes the circuit type that is being evaluated.

be immediately clear from these statements. The reason for this complexity is that in this work we strive for concrete (and in some cases achieve exact) values of the expected determinant or guessing probability, as opposed to asymptotic or approximate values. Further, our results are tailored to the ring-LWE setting which is crucially required by the CKKS scheme, and this setting introduces additional complexity as the entries of the matrices representing the noisy linear transformations of the secret and error are correlated instead of i.i.d. In order to obtain concrete estimates from the lemma and theorem statements, we ran scripts that used the expressions in the lemma and theorem statements, along with a BKZ-estimator, to compute the concrete hardness for various parameter sets and noise-flooding levels. As referenced above, we report our findings extensively in Figures 4-12.

In Section 11, we provide a graphical representation of our results and highlight our key findings. Most notably, we find that with noise-flooding levels of  $\rho_{\rm circ}^2$  and  $100 \cdot \rho_{\rm circ}^2$ , full guessing attacks are feasible after observing a sufficient number of decryption queries (at most  $\sim 100 {\rm K}$  needed), for all parameter sets and types of circuits considered. On the other hand, for noise level of  $t \cdot \rho_{\rm circ}^2$ , lattice reduction attacks are the only effective attacks.

Rephrasing the above, we investigate noise-flooding by  $x \cdot \rho_{\text{circ}}^2$ , where x ranges from 1 to t, where t is the number of decryption queries. We recall that  $\rho_{\text{circ}}^2$  corresponds to the variance of the average-case noise that is already present in the ciphertext. It follows that noise-flooding by  $x \cdot \rho_{\text{circ}}^2$  incurs an additional loss of  $\frac{1}{2}\log_2(x+1)$  bits in the message precision. This is in contrast to using the noise-flooding levels in [26], which incur a loss of an additional  $\log_2(\sigma) + 1$  bits of precision (beyond the worst-case noise already present in the ciphertext), where  $\sigma = 8\sqrt{tn}2^{\kappa/2}$ ,  $\kappa$  is the security parameter, and n is the dimension (see Definition 18 and Theorem 3 of [26]).

#### 2.4 Related Work

The inherent noise already present in a CKKS ciphertext was analyzed closely in [14]. We rely on their average-case analysis in our work in order to calibrate the noise-flooding noise and determine how much message precision is lost via the noise-flooding countermeasure.

The tools of incorporating side information on the LWE secret/error into a lattice reduction attack were developed in [16] via an introduction of an intermediate problem known as Distorted Bounded Distance Decoding (DBDD). Their framework allows the incorporation of "hints" into DBDD instances, which are finally converted to u-SVP instances via homogenization/isotropization, and can be applied to analyze the concrete security of the CKKS scheme with noise-flooding countermeasures. However, in practice, keeping track of the intermediate DBDD instance is not feasible for FHE-scale parameters. The security estimation for the LWE problem was revisited in [17], but those techniques similarly do not scale to FHE-size parameter sets.

The work of Kim, Lee, Seo, and Song [23] considered the provable security of the Hint-LWE problem, and it can be observed that the information obtained from noisy decryptions of fresh ciphertexts can be viewed as an instance of Hint-LWE. Theorem 1 in [23] provides a security reduction from a spherical LWE instance to Hint-LWE. However, because the conditional Gaussian distribution arising from the Hint-LWE problem is ellipsoidal (not spherical), the reduction is not tight (additional noise is added to convert from the spherical to ellipsoidal distribution). This is in contrast to our approach, which provides an attack that first converts the Hint-LWE instance to a DBDD instance. Importantly, a DBDD instance with an ellipsoidal distribution is equivalent to another DBDD instance with a spherical distribution, and there is no loss in this reduction. Thus, our concrete security estimates are tighter, but only apply to certain classes of attack strategies. We also note that reduction in Theorem 1 of [23] is for decisional LWE, whereas our attacks are for the search LWE problem, making the two results somewhat incomparable.

Two recent works [21,10] present a key-recovery attack on the schemes CKKS and the exact FHE schemes, respectively. Both attacks rely on the following observation: an average-case noise analysis models all noise terms as independent Gaussians. When that assumption fails, the noise predicted by an average-case noise analysis will underestimate the actual noise observed. Indeed both works successfully run a key-recovery attack by using correlated inputs. We note that, while that research direction is interesting, this does not affect our setting. In particular, in all circuits we consider (the identity circuit, and the classes C1 and C2),

the noise terms remain independent. We note that a recent work [5] argues that those attacks amount to incorrect estimation of the underlying ciphertext noise, as the heuristics specifically assume that inputs are independent, but [21,10] heavily rely on correlated inputs. The authors of [5] therefore define the notion of application-aware homomorphic encryption that can precisely counter these types of attacks. Our work therefore fits well within their model.

The work of Cheon, Hong and Kim [11] suggests noise-flooding countermeasures but does not delve deeply into the practical implications on the CKKS scheme's performance. Our work extends this by evaluating the practicality and efficiency of these countermeasures. Specifically, in our analysis, we examine the trade-offs between the number of allowed decryptions, noise-flooding levels, and concrete security. We show that while high levels of noise-flooding provide provable security, they significantly degrade message precision, making CKKS less practical for real-world applications (see Section 4.5).

The work of Bootle, Delaplace, Espitau, Fouque and Tibouchi [9] discusses LWE problems under certain conditions, relevant to our discussion on lattice problems post-noise flooding. Specifically, their techniques are most relevant to the analysis of our "guessing" attacks. They consider the case in which noisy linear equations (without reduction modulo q) are released on the LWE secret and error and provide bounds on the ratio of the noise versus coefficients of the linear equation needed to prevent guessing attacks. This corresponds to the setting of "approximate hints" that we use in this work to model the information learned by the adversary during decryption. Our results differ in that our analysis crucially takes into account the ring structure and distribution of the "hints" specific to the CKKS + noise-flooding setting. This is in contrast to [9] which assumes the hint vectors and the noise are independently drawn from distributions with known variances. Further, our goal is to provide a concrete, as opposed to asymptotic analysis. In particular, for a given CKKS + noise-flooding parameter set and a given target success probability, our analysis allows one to compute a concrete number of decryptions that are sufficient for the guessing attack to succeed with the target probability. Finally, our work derives the distribution of the "approximate hints" specific to the CKKS + noise-flooding setting and for the particular circuit classes we consider, whereas the prior work focused on the BLISS signature scheme.

The work of May and Nowakowski [28] shows a faster incorporation of hints into LWE problems, compared to that of Dachman-Soled et al. [16]. The reason we do not directly compare our efficiency to that of [28] is that their algorithms are only for modular and perfect hints (hints that correspond to noiseless linear equations modulo q or over the integers), whereas the hints required for the analysis in this work are  $approximate\ hints$ , which require computing the mean and covariance of a conditional Gaussian distribution.

Finally, Glaser, May, and Nowakowski [20] should be compared with our proposed technique, and we acknowledge that while it offers an efficient guessing method, our focus is on the practical complexity and concrete security estimates of such attacks in the context of CKKS. Our analysis includes the impact of noise-flooding on the effectiveness of guessing attacks and provides detailed estimates for the success probabilities of these attacks under various noise-flooding levels (see Section 7).

## 3 Preliminaries and Notation

Notation. We use bold lower case letters to denote vectors, and bold upper case letters to denote matrices. We use row notation for vectors, and denote by  $\mathbf{I}_d$  the identity matrix of dimension d. We denote by  $\{\mathbf{e}_i\}_{i\in[n]}$  the standard basis vectors in dimension n.

We use the notation  $R_q$  to denote the ring  $\mathbb{Z}[x]/(\Phi_m(x),q)$ , where  $\Phi_m(x)=x^n+1$ , and  $n=\phi(m)$  is a power of two. We denote ring elements by lowercase, non-bolded letters. When we employ a particular vector representation of a ring element in the coefficient or canonical embedding, we use vector notation.  $[\cdot]_q$  denotes modular reduction (mod q) (usually centered around 0).

We will make use of the canonical embedding and the subspace  $H \subseteq \mathbb{C}^{\mathbb{Z}_m^*}$  defined as follows:

$$H = \{\mathbf{x} = (x_i)_{i \in \mathbb{Z}_m^*} \in \mathbb{C}^n : x_i = \overline{x_{-i}}, \forall i \in \mathbb{Z}_m^* \}.$$

H is isomorphic to  $\mathbb{R}^n$  as an inner product space via the unitary transformation

$$B = \begin{pmatrix} \frac{1}{\sqrt{2}} \mathbf{I} & \frac{i}{\sqrt{2}} \mathbf{J} \\ \frac{1}{\sqrt{2}} \mathbf{J} & \frac{-i}{\sqrt{2}} \mathbf{I} \end{pmatrix}$$

where **I** is the identity matrix of size n/2 and **J** is its reversal matrix.

The canonical embedding of  $a \in \mathbb{Q}[x]/\Phi_m(x)$  into  $\mathbb{C}^n$  is the vector of evaluations of a at the roots of  $\Phi_m(x)$ . Specifically  $\sigma(a) = [a(\zeta^j)_{j \in \mathbb{Z}_m^*}]$ , where  $\zeta$  is a primitive m-th root of unity. Due to the conjugate pairs,  $\sigma$  maps into the subspace H. When a is represented as a vector of coefficients  $\mathbf{a}$ , we can express the canonical embedding transformation as a linear transformation  $\mathbf{aV}$ .

We denote by  $\mathcal{N}(\mu, \Sigma)$  the multivariate Gaussian with mean  $\mu$  and covariance  $\Sigma$ . We note that a multivariate Gaussian is fully determined by its mean and covariance. Thus, when the covariance of a dim dimensional multivariate Gaussian is a multiple of  $\mathbf{I}_{\text{dim}}$ , the dim variables are all independent.

DBDD and concrete hardness estimates. A DBDD instance (defined in [16]) consists of a tuple  $(\Lambda, \mu, \Sigma)$ , where  $\Lambda$  is a lattice, and  $(\mu, \Sigma)$  are viewed as the mean and covariance of a Gaussian distribution. Informally, the DBDD problem asks to find the unique vector in the lattice  $\Lambda$  that is contained in the ellipsoid defined by  $(\mu, \Sigma)$  (for the formal definition see [16]). The prior work of [16] showed how to transform a DBDD instance into a u-SVP instance with lattice  $\Lambda'$  using the homogenization and isotropization steps, and further showed that the secret vector of this u-SVP instance has expected squared norm  $||\mathbf{s}||^2 = \dim(\Lambda')$ . Thus, standard techniques can be used to estimate the hardness of the resulting u-SVP instance, where hardness is measured in terms of the "bikz" or BKZ- $\beta$  required to find the unique solution. In particular, following [3,6,16],  $\beta$  can be estimated as the minimum integer that satisfies

$$\sqrt{\beta} \le \delta_{\beta}^{2\beta - \dim(\Lambda') - 1} \text{Vol}(\Lambda')^{1/\dim(\Lambda')} \tag{1}$$

for a lattice  $\Lambda'$  where  $\delta$  is the root-Hermite-Factor of BKZ- $\beta$ .

The CKKS scheme. See Appendix 4 for a detailed description of the CKKS encryption scheme as well as a derivation of the error terms present in the message when decrypting a fresh ciphertext, and when decrypting after one or more multiplication steps (with or without a rescale operation). Following [14], we also present the noise variance in a fresh CKKS ciphertext, and in a ciphertext resulting from a multiplication and rescale operation (See Appendix 4.5).

## 4 The CKKS Scheme [12]

Let  $\chi$  be a discrete Gaussian of standard deviation  $\sigma = 3.2$ . We denote by  $\mathcal{ZO}(\rho)$  the distribution where 0 is sampled with probability  $\rho$ , and  $\pm 1$  are sampled with probability  $\rho/2$ . We denote the secret key distribution by S. We assume this to be the uniform ternary distribution.

We assume a sequence of moduli  $q_L, \ldots, q_0$ . After  $\ell$  levels of multiplication, we obtain level  $\ell$  ciphertexts with moduli  $q_{\overline{\ell}}$ , where  $\overline{\ell} = L - \ell$ . We let P be a relinearization modulus (typically P itself is a product of primes). We note that, although we present encryption as being performed at the "top" level L, it can be performed at any level  $\overline{\ell}$ . Finally, for readability, we omit the encoding and decoding algorithms, and refer the reader to the original CKKS paper [12].

SecretKeyGen( $\lambda$ ): Sample  $s \leftarrow S$  and output  $\mathtt{sk} = (1, s)$ .

PublicKeyGen(sk): For sk = (1, s), sample  $a \leftarrow R_{q_L}$  uniformly at random and  $e \leftarrow \chi$ . Output pk =  $([-as + e]_{q_L}, a)$ .

EvaluationKeyGen(sk, w): Set s = sk. Sample  $a' \leftarrow R_Q$ ,  $(Q = Pq_L)$  uniformly at random and  $e' \leftarrow \chi$ . Output  $\text{evk} = ([-a's + e' + Ps^2]_Q, a')$ .

Encrypt(pk, m): For the message  $m \in R$ . Let pk =  $(p_0, p_1)$ , sample  $v \leftarrow S$  and  $e_0, e_1 \leftarrow \chi$ . Output ct =  $([m + p_0v + e_0]_q, [p_1v + e_1]_q)$ .

Decrypt(sk,ct): Let ct =  $(c_0, c_1)$ . Output  $m' = [c_0 + c_1 s]_q$ .

 $\texttt{Add}(\texttt{ct}_0,\texttt{ct}_1) \texttt{:} \ \text{Given two level} \ \ell \ \text{ciphertexts, output} \ \texttt{ct} = ([\texttt{ct}_0[0] + \texttt{ct}_1[0]]_{q_{\overline{\ell}}}, [\texttt{ct}_0[1] + \texttt{ct}_1[1]]_{q_{\overline{\ell}}}).$ 

Pre-Multiply(ct<sub>0</sub>, ct<sub>1</sub>): Given two level  $\bar{\ell}$  ciphertexts, set

$$\begin{split} d_0 &= \left[ \mathsf{ct}_0[0] \mathsf{ct}_1[0] \right]_{q_{\overline{\ell}}} \\ d_1 &= \left[ \mathsf{ct}_0[0] \mathsf{ct}_1[1] + \mathsf{ct}_0[1] \mathsf{ct}_1[0] \right]_{q_{\overline{\ell}}} \\ d_2 &= \left[ \mathsf{ct}_0[1] \mathsf{ct}_1[1] \right]_{q_{\overline{\ell}}} \end{split}$$

Output  $ct = (d_0, d_1, d_2)$ .

Relinearize(ct, evk, P): Given level a level  $\bar{\ell}$  ciphertext as input, let  $\mathsf{ct}[0] = d_0$ ,  $\mathsf{ct}[1] = d_1$  and  $\mathsf{ct}[2] = d_2$ . Let  $\mathsf{evk}[0] = -a's + e' + Ps^2$  and  $\mathsf{evk}[1] = a'$ . Set

$$c'_0 = [d_0 + \lfloor P^{-1} \cdot d_2 \cdot (-a's + e' + Ps^2)]]_{q_{\overline{\ell}}}$$
  
$$c'_1 = [d_1 + \lfloor P^{-1} \cdot d_2 \cdot a']]_{q_{\overline{\ell}}}$$

Output  $ct' = (c'_0, c'_1)$ .

Rescale(ct,  $\Delta$ ): Given level a level  $\ell$  ciphertext as input, let ct =  $(c_0, c_1)$ . Set  $c_0' = \left[\left\lfloor \frac{c_0}{\Delta}\right\rfloor\right]_{q_{\overline{\ell}-1}}$  and  $c_1' = \left[\left\lfloor \frac{c_1}{\Delta}\right\rfloor\right]_{q_{\overline{\ell}}}$ . Output ct =  $(c_0', c_1')$ .

## 4.1 Decrypting a Fresh Ciphertext

Let ct be a fresh ciphertext encrypted under the public key pk, where we have  $pk = ([-as + e]_{q_L}, a)$ . Then, decrypting ct yields

$$\begin{aligned} \mathtt{Decrypt}(\mathtt{ct},\mathtt{sk}) &= c_0 + sc_1 \pmod{q_L} \\ &= m + p_0v + e_0 + svp_1 + se_1 \\ &= m + ve + e_0 + se_1. \end{aligned}$$

Recall that  $e, e_0, e_1 \leftarrow \chi$ . The ephemeral key v here is drawn from the same distribution as the secret key S, but sometimes it can be sampled from a slightly different distribution. This can for example be the distribution  $\mathcal{ZO}(\rho)$ .

## 4.2 Decrypting a Multiplication, No Rescale

Let  $\mathsf{ct} = (c_0, c_1)$  and  $\mathsf{ct}' = (c_0', c_1')$  be two level  $\ell$  ciphertexts that decrypt as follows.

$$c_0 + sc_1 = \frac{m^{2^{\ell}}}{\Delta^{2^{\ell} - 1}} + E$$
$$c'_0 + sc'_1 = \frac{m^{2^{\ell}}}{\Delta^{2^{\ell} - 1}} + E'.$$

Then, the output of Pre-Mult is

$$(d_0, d_1, d_2) = (c_0 c'_0, c_0 c'_1 + c'_0 c_1, c_1 c'_1).$$

Note that this decrypts as

$$d_0 + sd_1 + s^2 d_2 = (c_0 + c_1 s)(c'_0 + sc'_1)$$
$$= \frac{m^{2^{\ell+1}}}{\Lambda^{2^{\ell+1}-2}} + \tilde{E},$$

for some error  $\tilde{E}$ . Recall that the evaluation key is  $evk = ([-a's + e' + Ps^2]_Q, a')$ . Then, the output of Relinearize is

$$C_{0} = d_{0} + \lfloor P^{-1} \cdot d_{2} \cdot (-a's + e' + Ps^{2}) \rceil$$

$$= d_{0} + P^{-1} \cdot d_{2} \cdot (-a's + e' + Ps^{2}) + \epsilon_{0}$$

$$C_{1} = d_{1} + \lfloor P^{-1} \cdot d_{2} \cdot a' \rceil$$

$$= d_{1} + P^{-1} \cdot d_{2} \cdot a' + \epsilon_{1},$$

where  $\epsilon_i$  are rounding errors. Decrypting this yields

$$C_0 + sC_1 = d_0 + P^{-1}d_2(-a's + e' + Ps^2) + \epsilon_0 + sd_1 + sP^{-1}d_2a' + s\epsilon_1$$

$$= d_0 + sd_1 + s^2d_2 + (\epsilon_0 + \epsilon_2 s) + P^{-1}d_2e'$$

$$= \frac{m^{2^{\ell+1}}}{\Delta^{2^{\ell+1}-2}} + \tilde{E} + (\epsilon_0 + \epsilon_1 s) + P^{-1}d_2e'.$$

It has been shown that for all the FHE parameter sets we consider, the error above is dominated by  $\tilde{E} = E \cdot \frac{m^{2^{\ell}}}{\Delta^{2^{\ell}-1}} + E' \cdot \frac{m^{2^{\ell}}}{\Delta^{2^{\ell}-1}}$  [15,7].

## 4.3 Decrypting a Multiplication, With Rescale

From the previous subsection, we have that the noise after a Pre-Mult and a Relin is

$$C_0 + sC_1 = \frac{m^{2^{\ell+1}}}{\Lambda^{2^{\ell+1}-2}} + E + (\epsilon_0 + \epsilon_1 s) + P^{-1} d_2 e'.$$

We are going from level  $\ell$  to level  $\ell+1$  and from modulus  $q_{\overline{\ell}}$  to modulus  $q_{\overline{\ell}-1}$ . Following the notation of the previous subsection, we have the ciphertext

$$(C_0,C_1) = \mathtt{Relin}(\mathtt{Pre-Mult}(\mathtt{ct},\mathtt{ct}')).$$
 Let  $(C_0',C_1') = \mathtt{Rescale}(C_0,C_1) = (\left[\left\lfloor\frac{C_0}{\Delta}\right\rceil\right]_{q_{\overline{\ell}-1}},\left\lfloor\frac{C_1}{\Delta}\right\rceil]_{q_{\overline{\ell}-1}}).$  Then

$$\begin{split} \operatorname{Dec}((C_0',C_1'),\operatorname{sk}) &= C_0' + sC_1' \\ &= \left\lfloor \frac{C_0}{\Delta} \right\rfloor + s \left\lfloor \frac{C_1}{\Delta} \right\rfloor \\ &= \frac{C_0}{\Delta} + s\frac{C_1}{\Delta} + \delta_0 + s\delta_1 \\ &= \frac{1}{\Delta}(C_0 + sC_1) + \delta_0 + s\delta_1 \\ &= \frac{1}{\Delta}(\frac{m^{2^{\ell+1}}}{\Delta^{2^{\ell+1}-2}} + E + (\epsilon_0 + \epsilon_1 s) + P^{-1}d_2e') + \delta_0 + s\delta_1 \\ &= \frac{m^{2^{\ell+1}}}{\Delta^{2^{\ell+1}-1}} + \frac{1}{\Delta}(E + (\epsilon_0 + \epsilon_1 s) + P^{-1}d_2e') + \delta_0 + s\delta_1, \end{split}$$

where  $\delta_i$  are rounding errors, and we omit a reduction modulo  $q_{\overline{\ell}-1}$  throughout.

It was observed in [15,7] that the error above is typically dominated by  $\delta_0 + s\delta_1$  for most parameter sets. When the resulting ciphertext  $(C'_0, C'_1)$  is a level  $\ell'$  ciphertext, we denote the error as  $E_{\ell'}$ . Note that if an adversary knows the evaluation key evk, then the adversary can compute  $\delta_0$  and  $\delta_1$  on its own. Further, each element of  $\delta_0$  and  $\delta_1$  can be assumed to be independently and uniformly distributed between [-0.5, 0.5].

If the adversary does not know the evaluation key evk, then it will be unable to gain information about the values of  $\delta_0$  and  $\delta_1$  as evk[1] = a' is sampled uniformly at random. If the adversary knows

 $\operatorname{evk}[0] = [-a's + e' + Ps^2]_Q$  but not  $\operatorname{evk}[1]$ , then it is able to calculate  $\delta_0$  exactly by computing  $C_0$ . However, by the LWE assumption, it is unable to learn a' and thus cannot determine  $\delta_1$ . Similarly, knowing only  $\operatorname{evk}[1]$  allows the adversary to compute  $\delta_1$  exactly but learn nothing about  $\delta_0$ . In our attack model, as is standard, we will assume the adversary knows  $\operatorname{evk}$ .

## 4.4 Two or More Multiplications, With No Final Rescale

Recall that our chain of ciphertext moduli are formed as follows. Let  $q_0, \ldots, q_L$  be primes of roughly equal size. We recall that the size of the scaling parameter  $\Delta$  is also roughly equal to each  $q_i$ . Then, for any level i, the ciphertext modulus  $Q_i$  is  $Q_i = \prod_{j=0}^i q_j$ . We encrypt "at the top" level  $Q_L$ , and go "down" one level after each multiplication.

Let  $\operatorname{ct_0}$  and  $\operatorname{ct_1}$  be two ciphertexts encrypting the same message  $m^{2^\ell}$  at level  $\ell$ , where  $\ell > 0$ . Note that this implies that the re-scale operation has been performed on  $\operatorname{ct_0}$  and  $\operatorname{ct_1}$  and so the error in each of these ciphertexts is  $E_{\ell,0}, E_{\ell,1}$ . From the previous subsections, we know that these errors are dominated by  $\delta_{0,0} + s\delta_{1,0}$ , and  $\delta_{0,1} + s\delta_{1,1}$  respectively, for most parameter sets. Further, each element of  $\delta_{0,0}, \delta_{1,0}, \delta_{0,1}$  and  $\delta_{1,1}$  can be assumed to be independently and uniformly distributed between [-0.5, 0.5].

$$\mathsf{Dec}(\mathsf{ct}_0, \mathsf{sk}) = \frac{m^{2^\ell}}{\Delta^{2^\ell-1}} + E_{\ell,0} \pmod{q_{\overline{\ell}}}$$

$$\mathsf{Dec}(\mathsf{ct}_1, \mathsf{sk}) = \frac{m^{2^\ell}}{\Delta^{2^\ell - 1}} + E_{\ell, 1} \pmod{q_{\overline{\ell}}}.$$

Re-using the analysis from Section 4.2, we have that the error after multiplication without rescale is dominated by:

$$B = E_{\ell,0} \frac{m^{2^{\ell}}}{\Delta^{2^{\ell}-1}} + E_{\ell,1} \frac{m^{2^{\ell}}}{\Delta^{2^{\ell}-1}} + E_{\ell,0} E_{\ell,1}.$$

#### 4.5 CKKS Error Estimation

The following formulas are taken from [14] and will be useful in our work.

**4.5.1** Fresh Ciphertext The variance of the error of a fresh ciphertext with error distribution  $\mathcal{N}(0, \sigma_e^2 I_n)$  and ternary secret distribution (over domain  $\{-1, 0, 1\}$ ) of variance 2/3 is approximated as

$$\rho_{fresh}^2 = (\frac{4}{3}n + 1)\sigma^2.$$

**4.5.2** Multiplication With Rescale Multiplication of two ciphertexts with rescale results in a ciphertext with error of the following form

$$B_{final\ error} = \Delta^{-1}(B_{mult} + B_{ks}) + B_{round}.$$
 (2)

For the parameter sets we consider,  $B_{round}$  dominates the error, where  $B_{round}$  has variance

$$\rho_{mult\ error}^2 = \frac{n}{18} + \frac{1}{12}.$$

#### 4.6 Modeling Noisy Decryptions as Hints

For the case of identity circuits (we will deal with attacks on Class 1 and Class 2 circuits in Section 9), we concretely consider an adversary who obtains a CKKS public key  $\mathtt{pk} = ([-as+e]_q, a)$  and t independently sampled encryptions, and then asks for t decryptions of the constructed ciphertexts. Thus, for each  $j \in [t]$ , the adversary obtains the (noisy) polynomial  $e_1^j \cdot s + v^j \cdot e \approx \gamma^j$ , where multiplication is over the ring  $R_q$ . The adversary knows  $e_1^j$  and  $v^j$  whose coefficients are modeled as independent Gaussians with 0 mean and variance  $\sigma_{h_s}^2$  and  $\sigma_{h_e}^2$ , respectively. (s||e) corresponds to the LWE secret/error used to construct the public

key. Since we assume that all the polynomials involved have small magnitude, there is actually no wraparound modulo q. In this case, we can view the multiplication and addition as over the ring of integers  $\mathbb{Z}[x]/\Phi_m(x)$ , where  $\Phi_m(x)$  is the m-th cyclotomic polynomial of degree  $n = \phi(m)$ , and n is a power of two. Using the well-known representation of polynomial multiplication in  $\mathbb{Z}[x]/\Phi_m(x)$  as matrix-vector multiplication with vectors in  $\mathbb{Z}^n$ , we note that decryptions correspond to "hints," or noisy linear systems of equations with respect to secret/error vectors ( $\mathbf{s}||\mathbf{e}|$ ). The matrices corresponding to these systems of linear equations can be combined into a single matrix denoted as  $\mathbf{H}$  and referred to as the "hint matrix."

Since the original LWE secret/error distribution is (well-approximated) by a multivariate Gaussian (with mean  $\mathbf{0}$  and covariance  $\mathbf{\Sigma}$ ), and since the information obtained from decryption corresponds to noisy linear systems of equations on the LWE secret/error (where the noise is Gaussian), the information of the adversary after observing decryptions, is captured by a tuple  $(pk, \mu', \mathbf{\Sigma}')$ , where  $pk = ([-as + e]_{q_L}, a)$  is an LWE instance, and  $(\mu', \mathbf{\Sigma}')$  are the mean and covariance matrix of a multivariate Gaussian distribution corresponding to the *conditional distribution* of the LWE secret and error, given the information learned by the adversary during decryption (see, for example, Lemma 6 in [16]).

For purposes of our key recovery attacks, will further consider the related DBDD instance  $(\Lambda, \mu', \Sigma')$ , where  $\Lambda$  is the lattice obtained from the LWE instance via Kannan's embedding (see Section 3 and [16] for more details on the DBDD problem). Since the unique solution of this DBDD instance corresponds to the LWE secret and error, it is sufficient for our key recovery attacker to solve this DBDD instance. To estimate the concrete hardness of the DBDD instance  $(\Lambda, \mu', \Sigma')$ , it remains to compute  $\det(\Sigma')^{-1}$ . In Section 6, we will not compute this quantity directly, but instead compute its expected value. While our result makes the simplifying assumption that the coordinates of  $\mathbf{e}_1^j$  and  $\mathbf{v}^j$  are Gaussian (as opposed to ternary distributions or discrete Gaussians), we crucially take into account the **ring-LWE** setting, which gives rise to the algebraic structure of the hint matrix  $\mathbf{H}$ . Indeed, the expected determinant would significantly differ in the standard LWE setting, where  $\mathbf{H}$  would be modeled as a matrix whose entries are independent Gaussians. In our case, the entries of  $\mathbf{H}$  are *correlated*, making the analysis more delicate.

#### 5 Adversarial Model

Let us first examine the  $\mathsf{IND\text{-}CPA}^D$  adversarial model introduced by Li and Micciancio [25]. In their setting, the adversary had access to an additional (restricted) decryption oracle, which allowed it to launch a key recovery attack in the setting of approximate schemes, but which is equivalent to standard  $\mathsf{IND\text{-}CPA}$  in the case of exact schemes.

**Definition 5.1** (IND-CPA<sup>D</sup> Security [25]). Let  $\mathcal{E} = (\textit{KeyGen}, \textit{Encrypt}, \textit{Decrypt}, \textit{Eval})$  be a public-key homomorphic, approximate encryption scheme with plaintext space  $\mathcal{M}$  and ciphertext space  $\mathcal{C}$ . We define an experiment  $\mathsf{Expr}_b^{\mathsf{indcpa}^D}[\mathcal{A}]$ , parametrized by a bit  $b \in \{0,1\}$  and involving an efficient adversary  $\mathcal{A}$  that is given access to the following oracles, sharing a common state  $S \in (\mathcal{M} \times \mathcal{M} \times \mathcal{C})^*$  consisting of a sequence of message-message-ciphertext triplets:

- An encryption oracle  $Encrypt(pk, m_0, m_1)$  that, given a pair of plaintext messages  $m_0, m_1$ , computes  $ct \leftarrow Encrypt(pk, m_b)$ , extends the state

$$S := [S; (m_0, m_1, ct)]$$

with one more triplet, and returns the ciphertext ct to the adversary.

- An evaluation oracle  $H(\mathbf{evk}, g, J)$  that, given a function  $g : \mathcal{M}^k \to \mathcal{M}$  and a sequence of indices  $J = (j_1, \ldots, j_k) \in \{1, \ldots, |S|\}^k$ , computes the ciphertext  $\mathbf{ct} \leftarrow \mathbf{Eval}(\mathbf{evk}, g, S[j_1].\mathbf{ct}, \ldots, S[j_k].\mathbf{ct})$ , extends the state

$$S := [S; (q(S[j_1].m_0, \dots, S[j_k].m_1), q(S[j_1].m_1, \dots, S[j_k].m_1), ct)]$$

with one more triplet and returns the ciphertext ct to the adversary.

- A decryption oracle  $\operatorname{Decrypt}(\operatorname{sk}, j)$  that, given an index  $j \leq |S|$ , checks whether  $S[j].m_0 = S[j].m_1$ , and, if so, returns  $\operatorname{Decrypt}(\operatorname{sk}, S[j].\operatorname{ct})$  to the adversary.

The experiment is defined as

$$\begin{split} \mathsf{Expr}_b^{\mathsf{indcpa}^D}[\mathcal{A}](1^\kappa) : & (sk, pk, evk) \leftarrow \mathit{KeyGen}(1^\kappa) \\ S := [] \\ b' \leftarrow \mathcal{A}^{\mathit{Encrypt}(pk, \cdot, \cdot), \mathit{H}(evk, \cdot, \cdot), \mathit{Decrypt}(sk, \cdot)}(1^\kappa, pk, evk) \\ & \mathsf{return}(b') \end{split}$$

The advantage of adversary A against the IND-CPA security of the scheme is

$$\mathsf{Adv}_{\mathsf{indcpa}^D}[\mathcal{A}](\kappa) = \left| \Pr[\mathsf{Expr}_0^{\mathsf{indcpa}^D}[\mathcal{A}](1^\kappa) = 1] - \Pr[\mathsf{Expr}_1^{\mathsf{indcpa}^D}[\mathcal{A}](1^\kappa) = 1] \right|.$$

The scheme  $\mathcal{E}$  is IND-CPA<sup>D</sup>-secure if for any efficient (probabilistic polynomial time)  $\mathcal{A}$ , the advantage  $\mathsf{Adv}_{\mathsf{indcpa}^D}[\mathcal{A}]$  is negligible in  $\kappa$ .

In this work, we introduce a modification of the  $\mathsf{IND\text{-}CPA}^D$  that captures semi-honest attacks, in which the attacker passively corrupts a user in the system and obtains its view. We call our new notion  $\mathsf{IND\text{-}CPA}^{DSH}$ , where SH stands for Semi-Honest.

**Definition 5.2** (IND-CPA<sup>DSH</sup> Security with respect to admissible set  $\mathcal{G}$ ). Let  $\mathcal{E} = (KeyGen, Encrypt, Decrypt, Eval)$  be a public-key homomorphic, approximate encryption scheme with plaintext space  $\mathcal{M}$ , ciphertext space  $\mathcal{C}$ , randomness space  $\mathcal{R}$ . Let the set  $\mathcal{D}$  correspond to the image of Decrypt. We define an experiment  $\mathsf{Expr}_b^{\mathsf{indcpa}^{DSH}}[\mathcal{A},\mathcal{G}]$ , parametrized by a bit  $b \in \{0,1\}$  and involving an efficient adversary  $\mathcal{A}$ , given access to the following oracles. The oracles share a common state  $S \in (\mathcal{M} \times \mathcal{M} \times \mathcal{C} \times (\mathcal{R} \cup \{\bot\}) \times (\mathcal{D} \cup \{\bot\}), \{0,1\})^*$  consisting of a sequence of tuples. Each tuple consists of two messages, a ciphertext, the randomness used to generate the ciphertext or  $\bot$ , the decryption of the ciphertext or  $\bot$ , and a bit indicating whether the ciphertext is a fresh ciphertext that has not yet been included in an evaluation. The experiment is also parametrized by a set  $\mathcal{G}$  that consists of admissible tuples (S, g, J), where S is a valid state, g is a function  $g: \mathcal{M}^k \to \mathcal{M}$ , and J is a sequence of indices  $J = (j_1, \ldots, j_k) \in \{1, \ldots, |S|\}^k$ .

- An encryption oracle  $\textit{Encrypt}(\textit{pk}, m_0, m_1)$  that, given a pair of plaintext messages  $m_0, m_1$ , computes  $r \leftarrow \mathcal{R}$ ,  $\textit{ct} = \textit{Encrypt}(\textit{pk}, m_b; r)$ , and sets  $d = \perp$ , u = 0. If  $m_0 \neq m_1$ , it sets  $\rho = \perp$  and if  $m_0 = m_1$ , it sets  $\rho = r$ . It extends the state

$$S := [S; (m_0, m_1, ct, \rho, d, u)].$$

- An evaluation oracle H(evk, g, J) that, given a function  $g: \mathcal{M}^k \to \mathcal{M}$  and a sequence of indices  $J = (j_1, \ldots, j_k) \in \{1, \ldots, |S|\}^k$ , checks whether (S, g, J) is admissible by checking whether  $(S, g, J) \in \mathcal{G}$ . If so, the evaluation oracle computes the ciphertext  $ct \leftarrow \textbf{Eval}(\textbf{evk}, g, S[j_1].ct, \ldots, S[j_k].ct)$ , and extends the state

$$S := [S; (g(S[j_1].m_0, \ldots, S[j_k].m_0), g(S[j_1].m_1, \ldots, S[j_k].m_1), ct, \bot, \bot, \bot)].$$

Additionally, for  $\ell \in [k]$ , it sets the  $j_{\ell}$ -th tuple as follows

$$S[j_\ell] := (S[j_\ell].m_0, S[j_\ell].m_1, S[j_\ell].\mathsf{ct}, S[j_\ell].\rho, S[j_\ell].d, 1).$$

- A decryption oracle  $\operatorname{Decrypt}(\operatorname{sk},j)$  that, given an index  $j \leq |S|$ , checks whether  $S[j].m_0 = S[j].m_1$ . If so, the decryption sets  $d^* = \operatorname{Decrypt}(\operatorname{sk}, S[j].\operatorname{ct})$  and sets the j-th tuple of S as follows:

$$S[j] := (S[j].m_0, S[j].m_1, S[j].ct, S[j].\rho, d^*, 1).$$

At any point, the adversary can query its oracles with a special symbol ★. When this occurs, the entire state S is returned to the adversary. After this point, no further queries can be made to any of the oracles.

The experiment is defined as

$$\begin{split} \mathsf{Expr}_b^{\mathsf{indcpa}^{DSH}}[\mathcal{A},\mathcal{G}](1^\kappa) : & (\mathit{sk}, \mathit{pk}, \mathit{evk}) \leftarrow \mathit{KeyGen}(1^\kappa) \\ S := [] \\ b' \leftarrow \mathcal{A}^{\mathit{Encrypt}(\mathit{pk},\cdot,\cdot),\mathit{H}(\mathit{evk},\cdot,\cdot),\mathit{Decrypt}(\mathit{sk},\cdot)}(1^\kappa, \mathit{pk}, \mathit{evk}) \\ & \mathsf{return}(b') \end{split}$$

The advantage of adversary  $\mathcal A$  with respect to admissible set  $\mathcal G$  against the IND-CPA  $^{DSH}$  security of the scheme is

$$\begin{split} \mathsf{Adv}_{\mathsf{indcpa}^{DSH}}[\mathcal{A},\mathcal{G}](\kappa) &= \left| \Pr[\mathsf{Expr}_0^{\mathsf{indcpa}^D}[\mathcal{A},\mathcal{G}](1^\kappa) = 1] \right. \\ &\left. - \Pr[\mathsf{Expr}_1^{\mathsf{indcpa}^{DSH}}[\mathcal{A},\mathcal{G}](1^\kappa) = 1] \right|. \end{split}$$

The scheme  $\mathcal E$  is IND-CPA -secure with respect to admissible set  $\mathcal G$  if for any efficient (probabilistic polynomial time)  $\mathcal A$ , the advantage  $\mathsf{Adv}_{\mathsf{indcpa}^D}[\mathcal A,\mathcal G]$  is negligible in  $\kappa$ .

We now explain how our attacks on identity circuits and Class 1/Class 2 circuits can be viewed as  $\mathsf{IND-CPA}^{DSH}$  attacks.

The attack on identity circuits. Recall that our attacker simply asks for decryptions of fresh CKKS ciphertexts, and, given the internal randomness of the fresh ciphertexts and the noisy decryptions, runs a key recovery attack. We now formalize how this attack can be viewed as a IND-CPA<sup>DSH</sup> attack. Our attacker will query the Encrypt oracle t times with  $m_0 = m_1 = 0$  and once with  $m_0 \neq m_1$ . It will then query the Eval oracle t times with the identity function for each of the first t ciphertexts. Checking whether (S, g, J) is admissible corresponds to checking that g is the identity function, that J consists of a single index j and that S[j].u = 0. The adversary will make a decryption query for each of these t evaluated ciphertexts. The decryption oracle computes  $\text{Decrypt}(sk, ct) + \mathcal{N}(0, \sigma_{\epsilon}^2)$ , for some noise-flooding variance  $\sigma_{\epsilon}^2$ . Our adversary will then query with  $\bigstar$  to obtain the entire state S. and will use the information to run a full key recovery attack. Once it knows the key, it can trivially break indistinguishability by using the recovered key to decrypt the (t+1)-st ciphertext obtained from the encryption oracle and determine whether it encrypts  $m_0$  or  $m_1$ .

The attack on Class 1 or Class 2 circuits. Recall that our attacker asks for decryptions of Class 1 or Class 2 circuits evaluated on fresh CKKS ciphertexts, and given the noisy decryptions, runs a key recovery attack. We now formalize how this attack can be viewed as a IND-CPA attack. The attacker first generates fresh ciphertexts corresponding to the inputs for t evaluations of the Class 1 or Class 2 circuit. It does this by querying the  $\text{Encrypt}(pk,\cdot,\cdot)$  oracle sufficiently many times, where each call sets  $m_0 = m_1$ . It queries the  $\text{Encrypt}(pk,\cdot,\cdot)$  a final time with  $m_0 \neq m_1$ . The  $\text{H}_{\text{evk}}(\cdot,\cdot)$  oracle is then called with functions  $g:\mathcal{M}^k\to\mathcal{M}$  in Class 1 or Class 2 and with input indices  $J=(j_1,\ldots,j_k)$ . Checking whether (S,g,J) is admissible corresponds to checking that g is in Class 1 or Class 2 and that for all  $j\in J$ , S[j].u=0. Decryption queries are then made with ciphertexts  $\mathsf{ct}$  corresponding to the output of calls to  $\mathsf{H}(\mathsf{evk},\cdot,\cdot)$  as described above. The decryption oracle computes  $\mathsf{Decrypt}(\mathsf{sk},\mathsf{ct})+\mathcal{N}(0,\sigma_\epsilon^2)$ , for some noise-flooding variance  $\sigma_\epsilon^2$ . Our adversary will then query with  $\bigstar$  to obtain the entire state S. and will use the information to run a full key recovery attack. Once it knows the key, it can trivially break indistinguishability by using the recovered key to decrypt the last ciphertext obtained from the encryption oracle and determine whether it encrypts  $m_0$  or  $m_1$ .

## 6 Security Loss under a Lattice Reduction Attack

Recall that the matrix  $\Sigma$  corresponds to the original covariance matrix for the LWE secret and error. Formally, let  $\Sigma$  be an  $2n \times 2n$  diagonal matrix with the first n diagonal entries set to  $\sigma_s^2$ , the second n diagonal entries set to  $\sigma_e^2$ . The matrix  $\Sigma_{\varepsilon}$  corresponds to the covariance of the noise in the set of linear equations obtained on the LWE secret  $\mathbf{s}$  from decrypting a ciphertext. Formally,  $\Sigma_{\varepsilon} = \sigma_{\varepsilon}^2 \cdot \mathbf{I}_{tn}$ .  $\gamma = \gamma^1 || \cdots || \gamma^t$  corresponds to the obtained outputs.

First, note that for  $j \in [t]$ ,

$$e_1^j \cdot s = \mathbf{sVBP}\left(\mathbf{M}(e_1^j)\right)\mathbf{P}^{-1}\mathbf{B}^{-1}\mathbf{V}^{-1},$$

where **V** is the canonical embedding transformation into  $\mathbb{C}^n$ , **B** is the matrix corresponding to the isomorphism between  $H \subset \mathbb{C}^n$  and  $\mathbb{R}^n$ , **P** is a permutation matrix, and  $\mathbf{A}_1^j := \mathbf{M}(e_1^j)$  is a block diagonal matrix with n/2 blocks, each of dimension  $2 \times 2$ , where the *i*-th block is

$$\mathbf{A}_{1,i}^{j} := \begin{bmatrix} 1/\sqrt{2}w_{i,h_s}^{j} & 1/\sqrt{2}w_{n-i,h_s}^{j} \\ -1/\sqrt{2}w_{n-i,h_s}^{j} & 1/\sqrt{2}w_{i,h_s}^{j}, \end{bmatrix}$$

and  $\mathbf{w}_{h_s}^j = (w_{1,h_s}^j, \dots, w_{n,h_s}^j)$  is equal to  $\mathbf{w}_{h_s}^j = \mathbf{e}_1^j \mathbf{V} \mathbf{B}$ . Since  $\mathbf{V} \mathbf{B}$  is an isometry (an orthogonal matrix scaled by  $\sqrt{n}$ ), we have that  $\sigma_{h_s}^2(\mathbf{V} \mathbf{B})(\mathbf{V} \mathbf{B})^T = n\sigma_{h_s}^2 \cdot \mathbf{I}_n$ . So the random variables  $[w_{i,h_s}^j, w_{n-i,h_s}^j]_{j \in [t], i \in [n/2]}$  are distributed as independent Gaussians with variance  $n\sigma_{h_s}^2$ . Note that  $\mathbf{R} = (\mathbf{V} \mathbf{B} \mathbf{P})$  is a real matrix, even though  $\mathbf{V}$  and  $\mathbf{B}$  themselves are complex.

Similarly, for  $j \in [t]$ ,

$$v^j \cdot e = \mathbf{eVBP}\left(\mathbf{M}(\mathbf{v}^j)\right)\mathbf{P}^{-1}\mathbf{B}^{-1}\mathbf{V}^{-1}.$$

In this case,  $\mathbf{A}_2^j := \mathbf{M}(v^j)$  is a block diagonal matrix with n/2 blocks, each of dimension  $2 \times 2$ , where the *i*-th block is

$$\mathbf{A}_{2,i}^{j} := \begin{bmatrix} 1/\sqrt{2}w_{i,h_{e}}^{j} & 1/\sqrt{2}w_{n-i,h_{e}}^{j} \\ -1/\sqrt{2}w_{n-i,h_{e}}^{j} & 1/\sqrt{2}w_{i,h_{e}}^{j}, \end{bmatrix}$$

and  $\mathbf{w}_{h_e}^j = (w_{1,h_e}^j, \dots, w_{n,h_e}^j)$  is equal to  $\mathbf{w}_{h_e}^j = \mathbf{v}^j \mathbf{V} \mathbf{B}$ . Now for each  $j \in [t], i \in [n/2], w_{i,h_e}^j$  and  $w_{n-i,h_e}^j$  are random variables distributed as independent Gaussians with variance  $n\sigma_{h_e}^2$ .

Thus, if there are t decryption queries we can represent the hint matrix  $\mathbf{H}$  as:

$$\mathbf{H} = \begin{bmatrix} \mathbf{R} & 0 \\ 0 & \mathbf{R} \end{bmatrix} \begin{bmatrix} \mathbf{A}_1^1 & \mathbf{A}_1^2 & \dots & \mathbf{A}_1^t \\ \mathbf{A}_2^1 & \mathbf{A}_2^2 & \dots & \mathbf{A}_2^t \end{bmatrix} \begin{bmatrix} \mathbf{R}^{-1} & & & \\ & \ddots & & \\ & & \mathbf{R}^{-1} \end{bmatrix},$$

where **R** is an orthogonal matrix scaled by  $\sqrt{n}$ .

Applying the approximate hints of [16], the transformed covariance matrix  $\Sigma'$  and mean  $\mu'$  are as follows (the dimension and lattice of the DBDD instance remain unchanged from the instance described in Section 4.6):

$$\Sigma' = \Sigma - \Sigma \mathbf{H} (\mathbf{H}^T \Sigma \mathbf{H} + \Sigma_{\varepsilon})^{-1} \mathbf{H}^T \Sigma$$
(3)

$$\mu' = \gamma (\mathbf{H}^T \mathbf{\Sigma} \mathbf{H} + \mathbf{\Sigma}_{\varepsilon})^{-1} \mathbf{H}^T \mathbf{\Sigma}. \tag{4}$$

Our goal is to find  $\det(\Sigma')$ . Given this, we can estimate the hardness of the new DBDD instance under a lattice reduction attack. However, instead of computing  $\Sigma'$  and then  $\det(\Sigma')$  exactly, which requires inversion of a  $2n \times 2n$  matrix, we will instead compute the expected value of  $\det(\Sigma')$ , where the expectation is taken over the choice of the hint matrix  $\mathbf{H}$ .

Using a generalization of the Matrix Determinant Lemma, we obtain:

$$\mathbb{E}[\det((\mathbf{\Sigma}')^{-1})] = \mathbb{E}\left[\frac{\det(\mathbf{H}^T\mathbf{\Sigma}\mathbf{H} + \mathbf{\Sigma}_{\varepsilon})}{\det(\mathbf{\Sigma}_{\varepsilon})\det(\mathbf{\Sigma})}\right]. \tag{5}$$

Since  $\Sigma_{\varepsilon}$  and  $\Sigma$  are diagonal matrices whose entries depend on the parameters of the FHE cryptosystem, their determinants are constants and are easy to compute. Thus, it remains to compute  $\mathbb{E}[\det(\mathbf{H}^T\Sigma\mathbf{H} + \Sigma_{\varepsilon})]$ , which can then be plugged into (5).

**Lemma 6.1.** Let  $\mathbf{H}, \mathbf{R}, [\mathbf{A}_1^j = \mathbf{M}(e_1^j), \mathbf{A}_2^j = \mathbf{M}(v^j)]_{j \in [t]}$  be as described above. Then

$$\begin{split} \mathbb{E}[\det(\mathbf{H}^T \mathbf{\Sigma} \mathbf{H} + \mathbf{\Sigma}_{\epsilon})] &= \\ \left( \sigma_s^4 \sigma_e^4 \sigma_{\epsilon}^{4t-8} \left( \frac{7}{4} t (t-1) n^4 \sigma_{h_s}^4 \sigma_{h_e}^4 + t n^2 \sigma_{\epsilon}^4 \left( \frac{\sigma_{h_s}^4}{\sigma_e^4} + \frac{\sigma_{h_e}^4}{\sigma_s^4} \right) \right. \\ &+ \left. \left( t (t-1) n^2 \sigma_{h_s}^2 \sigma_{h_e}^2 + t n \sigma_{\epsilon}^2 \left( \frac{\sigma_{h_s}^2}{\sigma_s^2} + \frac{\sigma_{h_e}^2}{\sigma_s^2} \right) + \frac{\sigma_{\epsilon}^4}{\sigma_s^2 \sigma_s^2} \right)^{\frac{n}{2}}, \end{split}$$

where the expectation is taken over choice of  $\mathbf{e}_1^j \sim \mathcal{N}(0, \sigma_{h_s}^2)^n$  and  $\mathbf{v}^j \sim \mathcal{N}(0, \sigma_{h_e}^2)^n$  for all  $j \in [t]$ .

*Proof.* We use the fact that if **A** is an invertible n-by-n matrix and  $\mathbf{U}, \mathbf{V}$  are n-by-m matrices, then

$$\det\left(\mathbf{A} + \mathbf{U}\mathbf{V}^{\top}\right) = \det\left(\mathbf{I}_{\mathbf{m}} + \mathbf{V}^{\top}\mathbf{A}^{-1}\mathbf{U}\right)\det(\mathbf{A}),$$

and the definition of **H** and  $\Sigma$  to rewrite det  $(\mathbf{H}^T \Sigma \mathbf{H} + \Sigma_{\epsilon})$  as

$$\det \left( \mathbf{H}^{T} \mathbf{\Sigma} \mathbf{H} + \mathbf{\Sigma}_{\epsilon} \right)$$

$$= \det \left( \mathbf{I}_{2n} + \frac{1}{\sigma_{\epsilon}^{2}} \mathbf{\Sigma}^{1/2} \mathbf{H} \mathbf{H}^{T} \mathbf{\Sigma}^{1/2} \right) \det (\mathbf{\Sigma}_{\epsilon})$$

$$= \det \left( \mathbf{I}_{2n} + \frac{1}{\sigma_{\epsilon}^{2}} \begin{bmatrix} \sigma_{s}^{2} \mathbf{B}_{1,1} & \sigma_{s} \sigma_{e} \mathbf{B}_{1,2} \\ \sigma_{s} \sigma_{e} \mathbf{B}_{2,1} & \sigma_{e}^{2} \mathbf{B}_{2,2} \end{bmatrix} \right) \det (\mathbf{\Sigma}_{\epsilon})$$

$$= \det \left( \begin{bmatrix} \frac{\sigma_{s}^{2}}{\sigma_{\epsilon}^{2}} \sum_{j=1}^{t} \mathbf{A}_{1}^{j} (\mathbf{A}_{1}^{j})^{T} + \mathbf{I}_{n} & \frac{\sigma_{s} \sigma_{e}}{\sigma_{\epsilon}^{2}} \sum_{j=1}^{t} \mathbf{A}_{1}^{j} (\mathbf{A}_{2}^{j})^{T} \\ \frac{\sigma_{s} \sigma_{e}}{\sigma_{\epsilon}^{2}} \sum_{j=1}^{t} \mathbf{A}_{2}^{j} (\mathbf{A}_{1}^{j})^{T} & \frac{\sigma_{e}^{2}}{\sigma_{\epsilon}^{2}} \sum_{j=1}^{t} \mathbf{A}_{2}^{j} (\mathbf{A}_{2}^{j})^{T} + \mathbf{I}_{n} \end{bmatrix} \right) \det (\mathbf{\Sigma}_{\epsilon})$$

$$= \det (\star) \det (\mathbf{\Sigma}_{\epsilon}),$$

where  $\mathbf{B}_{k,l} := \mathbf{R} \left( \frac{1}{n} \sum_{j=1}^{t} \mathbf{A}_{k}^{j} (\mathbf{A}_{l}^{j})^{T} \right) \mathbf{R}^{T}$ . Exchanging two rows and two columns of  $\star$  at a time, which does not change the determinant, we obtain

$$\det(\star)\det(\mathbf{\Sigma}_{\epsilon}) = \det\begin{bmatrix} \mathbf{S}_1 & 0 & \dots & 0 & 0 \\ 0 & \mathbf{S}_2 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \mathbf{S}_{\frac{n}{2}-1} & 0 \\ 0 & 0 & \dots & 0 & \mathbf{S}_{\frac{n}{2}} \end{bmatrix} \det(\mathbf{\Sigma}_{\epsilon}) = \det(\circledast)$$

where

$$\det \mathbf{S}_{i} = \det \begin{bmatrix} a_{i} & 0 & c_{i} - d_{i} \\ 0 & a_{i} & d_{i} & c_{i} \\ c_{i} & d_{i} & b_{i} & 0 \\ -d_{i} & c_{i} & 0 & b_{i} \end{bmatrix} = \det \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \det(AD - BC), \tag{6}$$

$$\det(AD - BC) = \det\left(\begin{bmatrix} a_i b_i & 0\\ 0 & a_i b_i \end{bmatrix} - \begin{bmatrix} c_i & -d_i\\ d_i & c_i \end{bmatrix} \begin{bmatrix} c_i & d_i\\ -d_i & c_i \end{bmatrix}\right) = (a_i b_i - c_i^2 - d_i^2)^2,$$

$$\begin{split} a_i &= \frac{\sigma_s^2}{2\sigma_\epsilon^2} \left( \sum_{j=1}^t \left( (w_{i,h_s}^j)^2 + (w_{n-i,h_s}^j)^2 \right) + \frac{2\sigma_\epsilon^2}{\sigma_s^2} \right), \\ b_i &= \frac{\sigma_e^2}{2\sigma_\epsilon^2} \left( \sum_{j=1}^t \left( (w_{i,h_e}^j)^2 + (w_{n-i,h_e}^j)^2 \right) + \frac{2\sigma_\epsilon^2}{\sigma_e^2} \right), \\ c_i &= \frac{\sigma_s \sigma_e}{2\sigma_\epsilon^2} \sum_{j=1}^t \left( w_{i,h_s}^j w_{i,h_e}^j + w_{n-i,h_s}^j w_{n-i,h_e}^j \right), \\ d_i &= \frac{\sigma_s \sigma_e}{2\sigma_\epsilon^2} \sum_{j=1}^t \left( w_{i,h_s}^j w_{n-i,h_e}^j - w_{i,h_e}^j w_{n-i,h_s}^j \right). \end{split}$$

Note that (6) holds because if the blocks A, B, C, D are square matrices of the same size and, for example, C and D commute (i.e., CD = DC), then it holds that

$$\det\begin{pmatrix} A & B \\ C & D \end{pmatrix} = \det(AD - BC)$$

We therefore have that

$$\begin{split} &\det(\circledast) = \det(\Sigma_{\epsilon}) \prod_{i=1}^{n/2} \det(\mathbf{S}_{i}) \\ &= \sigma_{\epsilon}^{2tn} \prod_{i=1}^{n/2} \frac{\sigma_{s}^{4} \sigma_{e}^{4}}{16\sigma_{\epsilon}^{8}} \bigg( \left( \sum_{j=1}^{t} \left( (w_{i,h_{s}}^{j})^{2} + (w_{n-i,h_{s}}^{j})^{2} \right) + \frac{2\sigma_{\epsilon}^{2}}{\sigma_{s}^{2}} \right) \left( \sum_{j=1}^{t} \left( (w_{i,h_{e}}^{j})^{2} + (w_{n-i,h_{e}}^{j})^{2} \right) + \frac{2\sigma_{\epsilon}^{2}}{\sigma_{e}^{2}} \right) \\ &- \left( \sum_{j=1}^{t} \left( w_{i,h_{s}}^{j} w_{i,h_{e}}^{j} + w_{n-i,h_{s}}^{j} w_{n-i,h_{e}}^{j} \right) \right)^{2} - \left( \sum_{j=1}^{t} \left( w_{i,h_{s}}^{j} w_{n-i,h_{e}}^{j} - w_{i,h_{e}}^{j} w_{n-i,h_{s}}^{j} \right) \right)^{2} \bigg)^{2} \\ &= \left( \frac{\sigma_{s}^{4} \sigma_{e}^{4}}{16\sigma_{\epsilon}^{8-4t}} \right)^{\frac{n}{2}} \prod_{i=1}^{\frac{n}{2}} \left( \sum_{1 \leq j \neq k \leq t} \left( \left( w_{i,h_{s}}^{j} w_{i,h_{e}}^{k} \right)^{2} + \left( w_{i,h_{s}}^{j} w_{n-i,h_{e}}^{k} \right)^{2} + \left( w_{n-i,h_{s}}^{j} w_{n-i,h_{e}}^{k} \right)^{2} + \left( w_{n-i,h_{s}}^{j} w_{n-i,h_{e}}^{k} \right)^{2} - \left( w_{i,h_{s}}^{j} w_{i,h_{e}}^{k} w_{i,h_{e}}^{k} + w_{n-i,h_{s}}^{j} w_{n-i,h_{e}}^{k} w_{n-i,h_{s}}^{k} w_{n-i,h_{e}}^{k} \right)^{2} + \left( w_{n-i,h_{s}}^{j} w_{n-i,h_{e}}^{k} \right)^{2} + \left( w_{n-i,h_{e}}^{j} w_{n-i,h_{e}}^{k} w_{n-i,h_{e}}^{k} \right)^{2} + \left( \left( w_{i,h_{s}}^{j} w_{n-i,h_{e}}^{k} w_{n-i,h_{e}}^{k} \right)^{2} + \left( \left( w_{i,h_{s}}^{j} w_{n-i,h_{e}}^{k} w_{n-i,h_{e}}^{k} \right)^{2} + \left( \left( w_{i,h_{s}}^{j} w_{n-i,h_{e}}^{k} w_{n-i,h_{e}}^{k} \right)^{2} \right)^{2} + \left( \left( w_{n-i,h_{s}}^{j} w_{n-i,h_{e}}^{k} w_{n-i,h_{e}}^{k} \right)^{2} + \left( \left( w_{i,h_{s}}^{j} w_{n-i,h_{e}}^{k} w_{n-i,h_{e}}^{k} \right)^{2} + \left( \left( w_{i,h_{s}}^{j} w_{n-i,h_{e}}^{k} w_{n-i,h_{e}}^{k} \right)^{2} \right)^{2} + \left( \left( w_{i,h_{s}}^{j} w_{n-i,h_{e}}^{k} w_{n-i,h_{e}}^{k} w_{n-i,h_{e}}^{k} w_{n-i,h_{e}}^{k} w_{n-i,h_{e}}^{k} w_{n-i,h_{e}}^{k} w_{n-i,h_{e}}^{k} w_{n-i,h_{e}}^{k} \right)^{2} + \left( \left( w_{i,h_{s}}^{j} w_{n-i,h_{e}}^{k} w_{n-i,h_{e}}^{k} w_{n-i,h_{e}}^{k} \right)^{2} + \left( \left( w_{i,h_{s}}^{j} w_{n-i,h_{e}}^{k} w_{n-i,h_{e}}^{k} w_{n-i,h_{e}}^{k} w_{n-i,h_{e}}^{k} w_{n-i,h_{e}}^{k} w_{n-i,h_{e}}^{k} w_{n-i,h_{e}}^{k} \right)^{2} + \left( \left( w_{i,h_{s}}^{j} w_{n-i,h_{e}}^{k} w_{n-i,h_{e}}^{k} w_{n-i,h_{e}}^{k} w_{n-i,h_{e}}^{k} w_{n-i,h_{e}}^{k} w_{n-i,h_{e}}^{k} w_{n-i,h_{e}}^{k} w_$$

Now, to analyze the expectation  $\mathbb{E}[\det(\circledast)]$  of the above expression, we identify

$$\begin{split} Y_i &:= \sum_{1 \leq j \neq k \leq t} \left( \left( w_{i,h_s}^j w_{i,h_e}^k \right)^2 + \left( w_{i,h_s}^j w_{n-i,h_e}^k \right)^2 + \left( w_{n-i,h_s}^j w_{i,h_e}^k \right)^2 + \left( w_{n-i,h_s}^j w_{n-i,h_e}^k \right)^2 \\ &- \left( w_{i,h_s}^j w_{i,h_e}^j w_{i,h_s}^k w_{i,h_e}^k + w_{n-i,h_s}^j w_{n-i,h_e}^j w_{n-i,h_s}^k w_{n-i,h_e}^k \right. \\ &+ \left. w_{i,h_s}^j w_{n-i,h_e}^j w_{i,h_s}^k w_{n-i,h_e}^k + w_{n-i,h_s}^j w_{i,h_e}^j w_{n-i,h_s}^k w_{n-i,h_e}^k \right) \right) \\ &+ \frac{2\sigma_\epsilon^2}{\sigma_s^2} \sum_{j=1}^t \left( \left( \left( w_{i,h_e}^j \right)^2 + \left( \left( w_{n-i,h_e}^j \right)^2 \right) + \frac{2\sigma_\epsilon^2}{\sigma_e^2} \sum_{j=1}^t \left( \left( \left( w_{i,h_s}^j \right)^2 + \left( \left( w_{n-i,h_s}^j \right)^2 \right) + 4 \frac{\sigma_\epsilon^4}{\sigma_s^2 \sigma_e^2} \right) \right) \right] \\ &+ \frac{2\sigma_\epsilon^2}{\sigma_s^2} \sum_{j=1}^t \left( \left( \left( w_{i,h_e}^j \right)^2 + \left( \left( w_{n-i,h_e}^j \right)^2 \right) + 2 \frac{\sigma_\epsilon^2}{\sigma_e^2} \sum_{j=1}^t \left( \left( \left( w_{i,h_s}^j \right)^2 + \left( \left( w_{n-i,h_e}^j \right)^2 \right) + 2 \frac{\sigma_\epsilon^4}{\sigma_s^2 \sigma_e^2} \right) \right) \right) \\ &+ \frac{2\sigma_\epsilon^2}{\sigma_s^2} \sum_{j=1}^t \left( \left( \left( w_{i,h_e}^j \right)^2 + \left( \left( w_{n-i,h_e}^j \right)^2 \right) + 2 \frac{\sigma_\epsilon^2}{\sigma_e^2} \sum_{j=1}^t \left( \left( \left( w_{i,h_e}^j \right)^2 + \left( \left( w_{n-i,h_e}^j \right)^2 + \left( w_{n-i,h_e}^j \right)^2 + \left( \left( w_{n-i,h_e}^j$$

Since  $\{w_{i,h_s}^j\}_{j=1;i=1}^{t;n}$  and  $\{w_{i,h_e}^j\}_{j=1;i=1}^{t;n}$  are mutually independent, the expectation of the product is the product of each expectation,

$$\mathbb{E}[\det\left(\mathbf{H}^{T}\boldsymbol{\Sigma}\mathbf{H}+\boldsymbol{\Sigma}_{\epsilon}\right)] = \left(\frac{\sigma_{s}^{4}\sigma_{e}^{4}}{16\sigma_{\epsilon}^{8-4t}}\right)^{\frac{n}{2}}\prod_{i=1}^{\frac{n}{2}}\mathbb{E}Y^{2} = \left(\frac{\sigma_{s}^{4}\sigma_{e}^{4}}{16\sigma_{\epsilon}^{8-4t}}\right)^{\frac{n}{2}}\prod_{i=1}^{\frac{n}{2}}\left(\operatorname{Var}Y+\mathbb{E}^{2}Y\right),$$

where

$$VarY = 28t(t-1)n^{4}\sigma_{h_{s}}^{4}\sigma_{h_{e}}^{4} + 16tn^{2}\sigma_{\epsilon}^{4}(\frac{\sigma_{h_{s}}^{4}}{\sigma_{e}^{4}} + \frac{\sigma_{h_{e}}^{4}}{\sigma_{s}^{4}}),$$

$$\mathbb{E}^{2}Y = \left(4t(t-1)n^{2}\sigma_{h_{s}}^{2}\sigma_{h_{e}}^{2} + 4tn\sigma_{\epsilon}^{2}(\frac{\sigma_{h_{s}}^{2}}{\sigma_{\epsilon}^{2}} + \frac{\sigma_{h_{e}}^{2}}{\sigma_{\epsilon}^{2}}) + 4\frac{\sigma_{\epsilon}^{4}}{\sigma_{\epsilon}^{2}\sigma_{\epsilon}^{2}}\right)^{2}.$$

Finally, we obtain that

$$\begin{split} &\mathbb{E}[\det(\mathbf{H}^T\boldsymbol{\Sigma}\mathbf{H}+\boldsymbol{\Sigma}_{\epsilon})] = \\ &\left(\sigma_s^4\sigma_\epsilon^4\sigma_\epsilon^{4t-8}\Big(\frac{7}{4}t(t-1)n^4\sigma_{h_s}^4\sigma_{h_e}^4 + tn^2\sigma_\epsilon^4\big(\frac{\sigma_{h_s}^4}{\sigma_e^4} + \frac{\sigma_{h_e}^4}{\sigma_s^4}\big) + \Big(t(t-1)n^2\sigma_{h_s}^2\sigma_{h_e}^2 + tn\sigma_\epsilon^2\big(\frac{\sigma_{h_s}^2}{\sigma_e^2} + \frac{\sigma_{h_e}^2}{\sigma_s^2}\big) + \frac{\sigma_\epsilon^4}{\sigma_s^2\sigma_e^2}\Big)^2\Big)\right)^{\frac{n}{2}}. \end{split}$$

Obtaining the final hardness estimates. One can perform homogenization/isotropization of the DBDD instance (as in [16]) to obtain a u-SVP instance and then estimate the BKZ- $\beta$  for that instance. However, as described in [16], one can obtain the BKZ- $\beta$  estimates using only the dimension and volume of the lattice after homogenization/isotropization, and the lattice basis itself is not required. The lattice in our DBDD instance is a  $q_L$ -ary lattice and thus has log volume  $n \cdot \ln(q_L)$ . After homogenization/isotropization, the log volume of the lattice increases to  $n \ln(q_L) + \ln(\det((\Sigma')^{-1}))/2$ . Using (5) and Lemma 6.1, we use the expectation of  $\det((\Sigma')^{-1})$  in the above formula. The dimension remains unchanged after integrating hints. Thus, this information is sufficient for obtaining BKZ- $\beta$  estimates for the final u-SVP instance.

## 7 Key Recovery via Guessing

When  $\Sigma'$  in (3) has sufficiently small variance, then instead of running a lattice reduction attack, another strategy is to simply guess coordinates of the LWE secret/error by rounding the mean  $\mu'$  in (4) to the nearest integer. If n coordinates of these coordinates are guessed and all guesses are correct, then the entire LWE secret/error can be recovered by solving a linear system modulo q. To analyze the success of the above attack we begin with the following lemma:

**Lemma 7.1.** Let  $\Sigma'$  be defined as in (3). Then  $\operatorname{Tr}(\Sigma') \leq T = n \cdot \frac{\left(\frac{\sigma_s^2 \cdot \sigma_e^2 \cdot 2t \cdot n(\sigma_{h_s}^2 + \sigma_{h_e}^2)}{2 \cdot \sigma_e^2} + \sigma_s^2 + \sigma_e^2\right)}{B} + \frac{3\sqrt{2n \cdot V}}{B}$  with probability at least  $0.99 - 3n \cdot e^{-12.25}$  over choice of hint vectors, where

$$\begin{split} B &= \frac{\sigma_s^2 \cdot \sigma_e^2 \cdot (2t - 7\sqrt{2t})^2 \cdot n^2 \sigma_{h_s}^2 \cdot \sigma_{h_e}^2}{4 \cdot \sigma_\epsilon^4} + \frac{\sigma_s^2 \cdot (2t - 7\sqrt{2t}) \cdot n \sigma_{h_s}^2}{2 \cdot \sigma_\epsilon^2} \\ &\quad + \frac{\sigma_e^2 \cdot (2t - 7\sqrt{2t})(n \sigma_{h_e}^2)}{2 \cdot \sigma_\epsilon^2} + 1 - \frac{\sigma_s^2 \cdot \sigma_e^2 \cdot n^2 (\sigma_{h_s}^2 + \sigma_{h_e}^2)^2 (3.5\sqrt{2t} + 12.25)^2}{2 \cdot \sigma_\epsilon^4} \\ V &= \frac{\sigma_s^4 \cdot \sigma_e^4 \cdot (\mathbb{E}[R_1^2] + \mathbb{E}[R_2^2])}{4 \cdot \sigma_\epsilon^4} + 2\frac{\sigma_s^4 \cdot \sigma_e^4 \cdot \mathbb{E}[R_1] \cdot \mathbb{E}[R_2]}{2 \cdot \sigma_\epsilon^2} + (\sigma_s^2 + \sigma_e^2)^2 \\ &\quad + 2\frac{(\sigma_s^4 \cdot \sigma_e^2 + \sigma_s^2 \cdot \sigma_e^4) \cdot (\mathbb{E}[R_1] + \mathbb{E}[R_2])}{2 \cdot \sigma_\epsilon^2} + (\sigma_s^2 + \sigma_e^2)^2 \\ &\quad - \left(\frac{\sigma_s^2 \cdot \sigma_e^2 \cdot \mathbb{E}[R_1]}{2 \cdot \sigma_\epsilon^2} + \frac{\sigma_s^2 \cdot \sigma_e^2 \cdot \mathbb{E}[R_2]}{2 \cdot \sigma_\epsilon^2} + \sigma_s^2 + \sigma_e^2\right)^2 \\ \mathbb{E}[R_1] &= 2t \cdot n \sigma_{h_s}^2 \\ \mathbb{E}[R_2] &= 2t \cdot n \sigma_{h_e}^2 \\ \mathbb{E}[R_2] &= 4t n^2 \sigma_{h_e}^4 + 4t^2 n^2 \sigma_{h_e}^4 \\ \mathbb{E}[R_2^2] &= 4t n^2 \sigma_{h_e}^4 + 4t^2 n^2 \sigma_{h_e}^4 \end{split}$$

We note that up to parameter setting of n = 32768, the success probability in the above claim is at least 0.52.<sup>8</sup>

Proof. Recall that

$$\mathbf{\Sigma}' = \mathbf{\Sigma} - \mathbf{\Sigma} \mathbf{H} (\mathbf{H}^T \mathbf{\Sigma} \mathbf{H} + \mathbf{\Sigma}_{\varepsilon})^{-1} \mathbf{H}^T \mathbf{\Sigma}.$$

The eigenvalues of  $\Sigma'$  consist of the set of  $\alpha \in \mathbb{R}$  such that  $det(\Sigma' - \alpha \cdot \mathbf{I}) = 0$ . Equivalently,  $det((\Sigma - \alpha \cdot \mathbf{I}) - \Sigma \mathbf{H}(\mathbf{H}^T \Sigma \mathbf{H} + \Sigma_{\varepsilon})^{-1} \mathbf{H}^T \Sigma) = 0$ .

Using the generalization of the matrix determinant lemma, this is the same as finding  $\alpha$  such that  $det(\Sigma_{\varepsilon} + \mathbf{H}^{T}(\Sigma - \Sigma(\Sigma - \alpha \cdot \mathbf{I})^{-1}\Sigma)\mathbf{H}) = 0$ .

Let  $\tilde{\Sigma} = \tilde{\Sigma} - \Sigma (\tilde{\Sigma} - \alpha \cdot \tilde{\mathbf{I}})^{-1} \Sigma$ . Then we must find  $\alpha$  such that  $\det(\Sigma_{\varepsilon} + \mathbf{H}^{T} \tilde{\Sigma} \mathbf{H}) = 0$  Then  $\tilde{\Sigma}$  is a diagonal matrix with entries  $\frac{-\sigma_{s}^{2} \cdot \alpha}{\sigma_{s}^{2} - \alpha}$  in the first n positions and entries  $\frac{-\sigma_{e}^{2} \cdot \alpha}{\sigma_{e}^{2} - \alpha}$  in the last n positions. Using the analysis from the proof of Lemma 6.1, we have that

$$\det(\mathbf{\Sigma}_{\varepsilon} + \mathbf{H}^T \tilde{\mathbf{\Sigma}} \mathbf{H}) = \Pi_{i \in [n/2]} = \Pi_{i \in [n/2]} (a_i b_i - c_i^2 - d_i^2)^2, \tag{7}$$

where

$$a_i = \frac{-\sigma_s^2 \cdot \alpha}{2(\sigma_s^2 - \alpha)\sigma_\epsilon^2} R_{1,i} + 1,$$

$$b_i = \frac{-\sigma_e^2 \cdot \alpha}{2(\sigma_e^2 - \alpha)\sigma_\epsilon^2} R_{2,i} + 1,$$

$$c_i = \frac{-\sigma_s \cdot \sigma_3 \cdot \alpha}{2\sqrt{\sigma_s^2 - \alpha} \cdot \sqrt{\sigma_e^2 - \alpha}\sigma_\epsilon^2} R_{3,i},$$

$$d_i = \frac{-\sigma_s \cdot \sigma_3 \cdot \alpha}{2\sqrt{\sigma_s^2 - \alpha} \cdot \sqrt{\sigma_e^2 - \alpha}\sigma_\epsilon^2} R_{4,i}$$

<sup>&</sup>lt;sup>8</sup> For the parameter sets with n=131072, we increase 7 to 7.5, 3.5 to 3.75, 12.25 to 14.0625, and increase the probability to  $0.99-3n\cdot e^{-14}>0.66$ .

and

$$R_{1,i} = \sum_{j=1}^{t} (W_{i,h_s}^{j})^2 + (W_{n-i,h_s}^{j})^2$$

$$R_{2,i} = \sum_{j=1}^{t} (W_{i,h_e}^{j})^2 + (W_{n-i,h_e}^{j})^2$$

$$R_{3,i} = \sum_{j=1}^{t} W_{i,h_s}^{j} W_{i,h_e}^{j} + W_{n-i,h_s}^{j} W_{n-i,h_e}^{j}$$

$$R_{4,i} = \sum_{j=1}^{t} W_{i,h_s}^{j} W_{n-i,h_e}^{j} - W_{i,h_e}^{j} W_{n-i,h_s}^{j}.$$

So of the four eigenvalues  $(\alpha_{4i+1}, \alpha_{4i+2}, \alpha_{4i+3}, \alpha_{4i+4})$  corresponding to the *i*-th block, we have that  $\alpha_{4i+1} = \alpha_{4i+3}, \ \alpha_{4i+2} = \alpha_{4i+4}$ . Further, we can solve for  $\alpha_{4i+1}$  and  $\alpha_{4i+2}$  by finding the roots of the quadratic equation  $(a_ib_i - c_i^2 - d_i^2) = 0$ .  $\sum_{j \in [4]} \alpha_{4i+j}$  is then equal to the sum of those roots,  $\frac{-4q_{b,i}}{2q_{a,i}} = \frac{-2q_{b,i}}{q_{a,i}}$ , where

$$\begin{split} q_{a,i} &= \frac{\sigma_s^2 \cdot \sigma_e^2 \cdot R_{1,i} \cdot R_{2,i}}{4 \cdot \sigma_\epsilon^4} + \frac{\sigma_s^2 \cdot R_{1,i}}{2 \cdot \sigma_\epsilon^2} + \frac{\sigma_e^2 \cdot R_{2,i}}{2 \cdot \sigma_\epsilon^2} + 1 - \frac{\sigma_s^2 \cdot \sigma_e^2 \cdot R_{3,i}^2}{4 \cdot \sigma_\epsilon^4} - \frac{\sigma_s^2 \cdot \sigma_e^2 \cdot R_{4,i}^2}{4 \cdot \sigma_\epsilon^4} \\ q_{b,i} &= -\left(\frac{\sigma_s^2 \cdot \sigma_e^2 \cdot R_{1,i}}{2 \cdot \sigma_\epsilon^2} + \frac{\sigma_s^2 \cdot \sigma_e^2 \cdot R_{2,i}}{2 \cdot \sigma_\epsilon^2} + \sigma_s^2 + \sigma_e^2\right). \end{split}$$

Towards bounding  $\mathbb{E}\left[\frac{-q_{b,i}}{q_{a,i}}\right]$ , we first lower bound  $q_{a,i}$ . Using the fact that  $XY = 1/4(X+Y)^2 - 1/4(X-Y)^2$ , we can express  $R_{3,i}$  as

$$R_{3,i} = \sum_{j=1}^{2t} 1/4(X_j')^2 + \sum_{j=1}^{2t} 1/4(X_j'')^2$$

where  $X'_j$  and  $X''_j$  are a Gaussian random variable with variance  $n(\sigma^2_{h_s} + \sigma^2_{h_e}), X'_1, \dots, X'_{2t}$  are independent and  $X''_1, \dots, X''_{2t}$  are independent. The probability that either  $1/4 \sum_{j=1}^{2t} (X')_j^2 \notin \frac{2t \cdot n}{4} (\sigma^2_{h_s} + \sigma^2_{h_e}) \pm \frac{2 \cdot n(\sigma^2_{h_s} + \sigma^2_{h_e})(3.5 \sqrt{2t} + 12.25)}{4}$  or  $1/4 \sum_{j=1}^{2t} (X'')_j^2 \notin \frac{2t \cdot n}{4} (\sigma^2_{h_s} + \sigma^2_{h_e}) \pm \frac{2 \cdot n(\sigma^2_{h_s} + \sigma^2_{h_e})(3.5 \sqrt{2t} + 12.25)}{4}$  is at most  $2 \cdot e^{-12.25}$ . Thus,  $R_{3,i}$  and  $R_{4,i}$  are both in  $[-n(\sigma^2_{h_s} + \sigma^2_{h_e})(5\sqrt{2t} + 25), n(\sigma^2_{h_s} + \sigma^2_{h_e})(3.5\sqrt{2t} + 12.25)]$  with all but  $4 \cdot e^{-12.25}$  probability. Further,  $R^2_{3,i}$  (and similarly  $R^2_{4,i}$ ) is at most  $n^2(\sigma^2_{h_s} + \sigma^2_{h_e})^2(3.5\sqrt{2t} + 12.25)^2$ .

 $R_{1,i}$  can be expressed as the sum of 2t squares of Gaussians with variance  $n\sigma_{h_s}^2$ . So  $R_{1,i} \geq (2t-7\sqrt{2t}) \cdot n\sigma_{h_s}^2$  with probability  $1-e^{-12.25}$ . Similarly,  $R_{2,i} \geq (2t-7\sqrt{2t})(n\sigma_{h_e}^2)$  with probability  $1-e^{-12.25}$ .

Thus, we have that with all but  $1 - 6 \cdot e^{-12.25}$  probability,

$$\begin{split} q_{a,i} &= \frac{\sigma_s^2 \cdot \sigma_e^2 \cdot R_{1,i} \cdot R_{2,i}}{4 \cdot \sigma_\epsilon^4} + \frac{\sigma_s^2 \cdot R_{1,i}}{2 \cdot \sigma_\epsilon^2} + \frac{\sigma_e^2 \cdot R_{2,i}}{2 \cdot \sigma_\epsilon^2} + 1 \\ &- \frac{\sigma_s^2 \cdot \sigma_e^2 \cdot R_{3,i}^2}{4 \cdot \sigma_\epsilon^4} - \frac{\sigma_s^2 \cdot \sigma_e^2 \cdot R_{4,i}^2}{4 \cdot \sigma_\epsilon^4} \\ &\geq \frac{\sigma_s^2 \cdot \sigma_e^2 \cdot (2t - 7\sqrt{2t})^2 \cdot n^2 \sigma_{h_s}^2 \cdot \sigma_{h_e}^2}{4 \cdot \sigma_\epsilon^4} + \frac{\sigma_s^2 \cdot (2t - 7\sqrt{2t}) \cdot n \sigma_{h_s}^2}{2 \cdot \sigma_\epsilon^2} \\ &+ \frac{\sigma_e^2 \cdot (2t - 7\sqrt{2t})(n \sigma_{h_e}^2)}{2 \cdot \sigma_\epsilon^2} + 1 - \frac{\sigma_s^2 \cdot \sigma_e^2 \cdot n^2 (\sigma_{h_s}^2 + \sigma_{h_e}^2)^2 (3.5\sqrt{2t} + 12.25)^2}{2 \cdot \sigma_\epsilon^4} \\ &= B. \end{split}$$

Further, the above is true for all  $i \in [n/2]$  with probability at least  $1 - 3n \cdot e^{-12.25}$ . So we have that

$$\mathbb{E}\left[\frac{-2q_{b,i}}{q_{a,i}}\right] \leq \frac{\mathbb{E}[-2q_{b,i}]}{B}$$

$$= 2\frac{\left(\frac{\sigma_s^2 \cdot \sigma_e^2 \cdot \mathbb{E}[R_1]}{2 \cdot \sigma_e^2} + \frac{\sigma_s^2 \cdot \sigma_e^2 \cdot \mathbb{E}[R_2]}{2 \cdot \sigma_e^2} + \sigma_s^2 + \sigma_e^2\right)}{B}$$

$$= 2\frac{\left(\frac{\sigma_s^2 \cdot \sigma_e^2 \cdot 2t \cdot n\sigma_{h_s}^2}{2 \cdot \sigma_e^2} + \frac{\sigma_s^2 \cdot \sigma_e^2 \cdot 2t \cdot n\sigma_{h_e}^2}{2 \cdot \sigma_e^2} + \sigma_s^2 + \sigma_e^2\right)}{B}$$

We next bound the variance of  $-q_{b,i}$  (where the sum of  $\sum_{j\in[4]}\alpha_{4i+j}=-4q_{b,i}$ ). Note that  $\mathbb{E}[R_{1,i}]$  (resp.  $\mathbb{E}[R_{2,i}]$ ,  $\mathbb{E}[R_{2,i}]$ ,  $\mathbb{E}[R_{2,i}]$ ,  $\mathbb{E}[R_{2,i}]$ ) is the same for all  $i\in[n/2]$ . Therefore, we denote  $\mathbb{E}[R_1]=\mathbb{E}[R_{1,i}]$  (resp.  $\mathbb{E}[R_2]=\mathbb{E}[R_{2,i}]$ ,  $\mathbb{E}[R_2]=\mathbb{E}[R_{2,i}]$ ,  $\mathbb{E}[R_2]=\mathbb{E}[R_{2,i}]$ ,  $\mathbb{E}[R_2]=\mathbb{E}[R_{2,i}]$ ). We have:

$$\begin{split} \mathbb{E}[R_1] &= 2t \cdot n \sigma_{h_s}^2 \\ \mathbb{E}[R_2] &= 2t \cdot n \sigma_{h_e}^2 \\ \mathbb{E}[R_1^2] &= 4t n^2 \sigma_{h_s}^4 + 4t^2 n^2 \sigma_{h_s}^4 \\ \mathbb{E}[R_2^2] &= 4t n^2 \sigma_{h_e}^4 + 4t^2 n^2 \sigma_{h_e}^4 \end{split}$$

Further,  $R_1$  and  $R_2$  are independent.

$$\begin{split} V &= \mathbb{E}[(q_b)^2] - \mathbb{E}[q_b]^2 \\ &= \frac{\sigma_s^4 \cdot \sigma_e^4 \cdot (\mathbb{E}[R_1^2] + \mathbb{E}[R_2^2])}{4 \cdot \sigma_\epsilon^4} + 2 \frac{\sigma_s^4 \cdot \sigma_e^4 \cdot \mathbb{E}[R_1] \cdot \mathbb{E}[R_2]}{4 \cdot \sigma_\epsilon^4} \\ &+ 2 \frac{(\sigma_s^4 \cdot \sigma_e^2 + \sigma_s^2 \cdot \sigma_e^4) \cdot (\mathbb{E}[R_1] + \mathbb{E}[R_2])}{2 \cdot \sigma_\epsilon^2} + (\sigma_s^2 + \sigma_e^2)^2 \\ &- \left(\frac{\sigma_s^2 \cdot \sigma_e^2 \cdot \mathbb{E}[R_1]}{2 \cdot \sigma_\epsilon^2} + \frac{\sigma_s^2 \cdot \sigma_e^2 \cdot \mathbb{E}[R_2]}{2 \cdot \sigma_\epsilon^2} + \sigma_s^2 + \sigma_e^2\right)^2 \end{split}$$

Using Chebyshev, we therefore have that  $\Pr[|\mathbb{E}[\mathbf{Tr}(\mathbf{\Sigma}')] - \mathbf{Tr}(\mathbf{\Sigma}')| > 10^{\frac{\sqrt{2n \cdot V}}{B}}] \le 0.01$ . Thus, putting everything together, we have that with  $0.99 - 3n \cdot e^{-12.25}$  probability,

$$\mathbf{Tr}(\mathbf{\Sigma}') \leq n \cdot \frac{\left(\frac{\sigma_s^2 \cdot \sigma_e^2 \cdot 2t \cdot n(\sigma_{h_s}^2 + \sigma_{h_e}^2)}{2 \cdot \sigma_e^2} + \sigma_s^2 + \sigma_e^2\right)}{R} + \frac{10\sqrt{2n \cdot V}}{R}.$$

Given the above, we consider the distribution of  $\mathbf{e}||\mathbf{s} - \mu'$ , where  $\mu'$  is the mean from equation (4). The random variable  $\mathbf{e}||\mathbf{s} - \mu'$  is distributed as the multivariate Gaussian distribution  $\mathcal{N}(0, \mathbf{\Sigma}')$ .  $\mu'$  is the correct guess for  $\mathbf{e}||\mathbf{s}$  as long as for all  $i \in [n]$   $|e_i - \mu'_i| \leq 0.5$  and for all  $i \in [n]$   $|s_i - \mu'_i| \leq 0.5$ . The probability that the above occurs for each coordinate is the same as the probability weight of the hypercube corresponding to  $-0.5 \leq x_i \leq 0.5, i \in [n]$  under the multivariate Gaussian distribution  $\mathcal{N}(0, \mathbf{\Sigma}')$ . We use the following theorem to lower bound this probability weight:

Theorem 7.2 (Special case of the Gaussian Correlation Inequality [24]). Let X be an n-dimensional Gaussian random variable. Then for any  $t_1, \ldots, t_n > 0$ ,

$$\mathbb{P}(|X_1| \le t_1, \dots, X_n \le t_n] \ge \mathbb{P}(|X_1| \le t_1) \cdots \mathbb{P}(|X_n| \le t_n].$$

We instantiate the above theorem with **X** consisting of a subset S of size n of the coordinates of the conditional Gaussian distribution  $((\mathbf{s}||\mathbf{e}) - \mu') \sim \mathcal{N}(\mathbf{0}, \Sigma')$ , with  $t_i = 0.5, j \in S$  We thus have that

$$\mathbb{P}(|X_j| \le t_j, i \in S) \ge \prod_{j \in S} \mathbb{P}_{X_j \sim \mathcal{N}(0, \mathbf{e}_j \mathbf{\Sigma}' \mathbf{e}_i^T)}(|X_j| \le t_j], \tag{8}$$

where the  $\mathbf{e}_{i}$  are the standard basis vectors.

To analyze  $\Pr_{X_j \sim \mathcal{N}(0, \mathbf{e}_j \cdot \mathbf{\Sigma}' \cdot \mathbf{e}_j^T)}[X_j \leq 0.5]$ , we note that  $\sum_{i \in [2n]} \mathbf{e}_i \cdot \mathbf{\Sigma}' \cdot \mathbf{e}_i^T = \mathsf{Tr}(\mathbf{\Sigma}')$ . By Lemma 7.1, we have that  $\mathsf{Tr}(\mathbf{\Sigma}') \leq \mathsf{T}$  with 53% probability. Let  $S \subseteq [2n]$  of size n be the set of indices j corresponding to the n smallest values among  $\{\mathbf{e}_i \cdot \mathbf{\Sigma}' \cdot \mathbf{e}_i^T : i \in [2n]\}$  this set of minimum values. Using the analysis in Appendix A, we have that for each  $j \in S$ ,  $\mathbf{e}_j \cdot \mathbf{\Sigma}' \cdot \mathbf{e}_j^T \leq \frac{\mathsf{T}}{2n}$ . Therefore,

$$\Pr_{X_j \sim \mathcal{N}(0, \mathbf{e}_j \cdot \mathbf{\Sigma}' \cdot \mathbf{e}_j^T)}[|X_j| \le 0.5] \ge -\text{erf}\left(\frac{-0.5}{\sqrt{2 \cdot \frac{\mathsf{T}}{2n}}}\right) = \text{erf}\left(\frac{0.5}{\sqrt{2 \cdot \frac{\mathsf{T}}{2n}}}\right). \tag{9}$$

Finally, the attack is as follows: The adversary chooses to guess the values of  $\mathbf{e}_j$  or  $\mathbf{s}_j$  for these n smallest values (corresponding to the set S), and then use the LWE instance to solve for the remaining n variables. The probability that all of the adversary's guesses are correct is lower bounded by the probability weight on the hypercube corresponding to  $|X_j| \leq 0.5, j \in I$  when X is drawn from the multivariate Gaussian distribution  $X \sim \mathcal{N}(0, \Sigma')$ . Using (8) and (9), this is at most

$$\prod_{j \in S} - \mathrm{erf}\left(\frac{-0.5}{\sqrt{2 \cdot \mathbf{e}_j \cdot \mathbf{\Sigma}' \cdot \mathbf{e}_j^T}}\right) \geq \left(-\mathrm{erf}\frac{-0.5}{\sqrt{2 \cdot \frac{\mathsf{T}}{2n}}}\right)^n = \mathrm{erf}\left(\frac{0.5}{\sqrt{\frac{\mathsf{T}}{n}}}\right)^n.$$

The final success probability of the attack is:<sup>9</sup>

$$\operatorname{erf}\left(\frac{0.5}{\sqrt{\frac{1}{n}}}\right)^{n} - 3n \cdot e^{-12.25} - 0.01. \tag{10}$$

## 8 Hybrid Guessing/Lattice-Reduction Attacks

Recall the structure of the eigenvalues of  $\Sigma'$ : There are [n/2] blocks and for each  $i \in [n/2]$ , the eigenvalues  $(\alpha_{4i+1}, \alpha_{4i+2}, \alpha_{4i+3}, \alpha_{4i+4})$ , where  $\alpha_{4i+1} = \alpha_{4i+3}$ ,  $\alpha_{4i+2} = \alpha_{4i+4}$ . For each  $i \in [n/2]$ , we say that  $\{\alpha_{4i+1}, \alpha_{4i+2}\}$  and  $\{\alpha_{4i+3}, \alpha_{4i+4}\}$  are pairs. For each i, the adversary computes  $\mathbf{e}_i \Sigma' \mathbf{e}_i^T$  and guesses  $\mu_i$  for the g minimum values where g is the maximum value such that

$$\operatorname{erf}\left(\frac{0.5}{\sqrt{\frac{\mathsf{T}}{n}}}\right)^g \ge p,\tag{11}$$

for some threshold p. These guesses are made and incorporated as perfect hints. After this process, the covariance matrix is a principal submatrix of  $\Sigma'$  of dimension  $(2n-g) \times (2n-g)$ , which we denote by  $\Sigma''$ . We denote by  $\operatorname{PSub}_{2n-g}(\Sigma')$  the set of all principal submatrices of  $\Sigma'$  of dimension 2n-g. Similarly, the lattice reduces dimension by g and its volume remains the same. The following lemma gives a bound on the determinant of  $\Sigma''$ .

**Lemma 8.1.** Let  $g \in \{0, 1, ..., n\}$ . Let  $\Sigma'$  be defined as in (3). Let  $\Sigma'' = \operatorname{argmax}_{\widetilde{\Sigma} \in \mathsf{PSub}_{2n-g}(\Sigma')} \mathsf{Tr}(\widetilde{\Sigma})$ . With probability  $0.99 - 4n \cdot e^{-12.25}$  over choice of hint vectors, 10

$$\mathbf{Tr}(\boldsymbol{\Sigma}') \leq \mathsf{T} \quad \textit{and} \quad \mathsf{det}(\boldsymbol{\Sigma}'') \leq \frac{\mathsf{det}(\boldsymbol{\Sigma}')}{\left(\frac{L}{U}\right)^g},$$

<sup>&</sup>lt;sup>9</sup> And for n = 131072, we replace  $e^{-12.25}$  with  $e^{-14}$ .

<sup>&</sup>lt;sup>10</sup> For the parameter sets with n=131072, we increase 7 to 7.5, 24.5 to 28.125 and increase the probability to  $0.99-4n\cdot e^{-14}$ .

where T and B are defined as in Lemma 7.1, and

$$\begin{split} L &= \frac{G + \sqrt{G^2 - 4 \cdot B \cdot \sigma_s^2 \cdot \sigma_e^2}}{2 \cdot B} \\ U &= \frac{\sigma_s^2 \cdot \sigma_e^2}{B_{max}} \\ G &= \sigma_s^2 \cdot \sigma_e^2 (2t + 7\sqrt{2t} + 24.5) \cdot (n\sigma_{h_e}^2) 2 \cdot \sigma_\epsilon^2 \\ &+ \sigma_s^2 \cdot \sigma_e^2 (2t + 7\sqrt{2t} + 24.5) \cdot (n\sigma_{h_s}^2) 2 \cdot \sigma_\epsilon^2 + \sigma_s^2 + \sigma_e^2 \\ B_{max} &= \frac{\sigma_s^2 \cdot \sigma_e^2 \cdot (2t + 7\sqrt{2t} + 24.5)^2 \cdot n^2 \sigma_{h_s}^2 \cdot \sigma_{h_e}^2}{4 \cdot \sigma_\epsilon^4} + \frac{\sigma_s^2 \cdot (2t + 7\sqrt{2t} + 24.5) \cdot n\sigma_{h_s}^2}{2 \cdot \sigma_\epsilon^2} \\ &+ \frac{\sigma_e^2 \cdot (2t + 7\sqrt{2t} + 24.5)(n\sigma_{h_e}^2)}{2 \cdot \sigma_\epsilon^2} + 1. \end{split}$$

Proof. Lemma 7.1 showed that with probability  $0.99 - 3n \cdot e^{-12.25}$  over choice of hint vectors,  $\mathbf{Tr}(\mathbf{\Sigma}') \leq \mathsf{T}$ . Let  $\alpha_1, \ldots, \alpha_g$  be the g minimum eigenvalues of  $\mathbf{\Sigma}'$ . Using the Eigenvalue Interlacing Theorem [22], we have that  $\det(\mathbf{\Sigma}'') \leq \frac{\det(\mathbf{\Sigma}')}{\alpha_1 \cdots \alpha_g}$ . We therefore need a lower bound on  $\alpha_1 \cdots \alpha_g$ .

We consider  $\alpha'_1,\ldots,\alpha'_g$  such that for all  $i\in[g],\{\alpha_i,\alpha'_i\}$  are a pair. We show an upper bound U on  $\alpha'_i\leq U$  for all  $i\in[g]$ . We further show a lower bound L on all  $\alpha'_i\cdot\alpha_i\geq L$  for all  $i\in[g]$  with all but  $1-n\cdot e^{-12.25}$  probability. Finally, this allows us to obtain a lower bound  $\alpha_i\geq \frac{L}{\alpha'_i}\geq \frac{L}{U}$  for all  $i\in[g]$ .  $\alpha_1\cdots\alpha_g$  can then be lower bounded by  $\left(\frac{L}{U}\right)^g$ . which implies that

$$\mathsf{det}(\boldsymbol{\Sigma}'') \leq \frac{\mathsf{det}(\boldsymbol{\Sigma}')}{\left(\frac{L}{U}\right)^g}.$$

Specifically, assuming the bounds from the proof of Lemma 7.1, and assuming in addition the following upper bounds on  $R_{1,i}$ ,  $R_{2,i}$ , which occurs with  $1 - 2 \cdot e^{-12.25}$  probability, <sup>11</sup>

$$R_{1,i} \le (2t + 7\sqrt{2t} + 24.5) \cdot n\sigma_{h_s}^2$$
  $R_{2,i} \le (2t + 7\sqrt{2t} + 24.5) \cdot n\sigma_{h_e}^2$ , (12)

we have that

$$-q_{b_i} \leq \sigma_s^2 \cdot \sigma_e^2 (2t + 7\sqrt{2t} + 24.5) \cdot (n\sigma_{h_e}^2) 2 \cdot \sigma_\epsilon^2 + \sigma_s^2 \cdot \sigma_e^2 (2t + 7\sqrt{2t} + 24.5) \cdot (n\sigma_{h_s}^2) 2 \cdot \sigma_\epsilon^2 + \sigma_s^2 + \sigma_e^2 = G.$$

Thus,

$$\forall i \in [g], \alpha_i' = \frac{-q_{b,i} + \sqrt{q_{b,i}^2 - 4q_{a,i} \cdot q_{c,i}}}{2 \cdot q_{a,i}} \le \frac{G + \sqrt{G^2 - 4 \cdot B \cdot \sigma_s^2 \cdot \sigma_e^2}}{2 \cdot B} = L.$$

<sup>&</sup>lt;sup>11</sup> For the parameter sets with n=131072, we increase 7 to 7.5, 24.5 to 28.125 and increase the probability to  $1-2 \cdot e^{-14}$ .

Using the same upper bounds from (12) we also have the following upper bound on  $q_{a,i}$ :12

$$q_{a_{i}} = \frac{\sigma_{s}^{2} \cdot \sigma_{e}^{2} \cdot R_{1,i} \cdot R_{2,i}}{4 \cdot \sigma_{\epsilon}^{4}} + \frac{\sigma_{s}^{2} \cdot R_{1,i}}{2 \cdot \sigma_{\epsilon}^{2}} + \frac{\sigma_{e}^{2} \cdot R_{2,i}}{2 \cdot \sigma_{\epsilon}^{2}} + 1 - \frac{\sigma_{s}^{2} \cdot \sigma_{e}^{2} \cdot R_{3,i}}{4 \cdot \sigma_{\epsilon}^{4}} - \frac{\sigma_{s}^{2} \cdot \sigma_{e}^{2} \cdot R_{4,i}}{4 \cdot \sigma_{\epsilon}^{4}}$$

$$\leq \frac{\sigma_{s}^{2} \cdot \sigma_{e}^{2} \cdot R_{1,i} \cdot R_{2,i}}{4 \cdot \sigma_{\epsilon}^{4}} + \frac{\sigma_{s}^{2} \cdot R_{1,i}}{2 \cdot \sigma_{\epsilon}^{2}} + \frac{\sigma_{e}^{2} \cdot R_{2,i}}{2 \cdot \sigma_{\epsilon}^{2}} + 1$$

$$\leq \frac{\sigma_{s}^{2} \cdot \sigma_{e}^{2} \cdot (2t + 7\sqrt{2t} + 24.5)^{2} \cdot n^{2} \sigma_{h_{s}}^{2} \cdot \sigma_{h_{e}}^{2}}{4 \cdot \sigma_{\epsilon}^{4}} + \frac{\sigma_{s}^{2} \cdot (2t + 7\sqrt{2t} + 24.5) \cdot n \sigma_{h_{s}}^{2}}{2 \cdot \sigma_{\epsilon}^{2}}$$

$$+ \frac{\sigma_{e}^{2} \cdot (2t + 7\sqrt{2t} + 24.5)(n \sigma_{h_{e}}^{2})}{2 \cdot \sigma_{\epsilon}^{2}} + 1$$

$$= B_{max}. \tag{14}$$

Thus,

$$\begin{aligned} \forall i \in [g], \alpha_i' \cdot \alpha_i &= \frac{-q_{b,i} + \sqrt{q_{b,i}^2 - 4q_{a,i} \cdot q_{c,i}}}{2 \cdot q_{a,i}} \cdot \frac{-q_{b,i} - \sqrt{q_{b,i}^2 - 4q_{a,i} \cdot q_{c,i}}}{2 \cdot q_{a,i}} \\ &= \frac{q_{c,i}}{q_{a,i}} \\ &\geq \frac{\sigma_s^2 \cdot \sigma_e^2}{B_{max}} = U. \end{aligned}$$

Combining Lemma 7.1 with Theorem 7.2 as before, we estimate that with at least  $p - 4n \cdot e^{-12.25} - 0.01$  probability, all g number of guesses are correct, and

$$\det(\mathbf{\Sigma}'') \le \frac{\det(\mathbf{\Sigma}')}{\left(\frac{L}{U}\right)^g}.$$
 (15)

We note that for up to n=32768,  $4n\cdot e^{-12.5}\leq 0.63$ . <sup>13</sup> As before,  $\mathbb{E}[\det(\Sigma')]$  can be computed via Lemma 6.1. Thus, we can use (15) to obtain a bound on the expected value of  $\det(\Sigma'')$  (conditioned on events with probability at least  $0.99-4n\cdot e^{-12.25}$  occurring), compute the log-volume of the lattice after homogenization/isotropization as described in Section 4.6, and use the log-volume and dimension to estimate the hardness of the residual instance (after guesses) under a lattice reduction attack.

## 9 Extending to Larger Classes of Circuits

## 9.1 The First Class of Circuits and Lattice Reduction Attacks

In Figure 1a we present the first class of circuits we consider. The circuits  $C_1, \ldots, C_\ell$  that are depicted each consist of  $\log(r)$  levels of multiplications as well as any number of additions. The final gate in each of the circuits  $C_1, \ldots, C_\ell$  is a multiplication with rescale. Note that the noise after multiplication with rescale in circuit  $C_i$  is dominated by  $\delta_1^i \cdot s + \delta_0^i$  (see Section 4.5.2), where  $\delta_0^i, \delta_1^i$  are distributed as uniform random variables in the range [-0.5, 0.5].

The final gate of the entire circuit is an addition gate that adds the outputs of each of the  $C_i$  circuits. We require  $\ell$  subcircuits and a final addition gate in order to ensure that the linear coefficients of the noise polynomial (which are independent and uniform random in the range [-0.5, 0.5] for each of the  $\ell$  circuits) can be approximated by Gaussian random variables with mean 0 and variance  $\frac{\ell}{12}$ , the setting for which our Lemma 6.1 applies.

<sup>&</sup>lt;sup>12</sup> For the parameter sets with n = 131072, we increase 7 to 7.5, and 24.5 to 28.

<sup>&</sup>lt;sup>13</sup> And for n = 131072,  $4n \cdot e^{-14} < 0.44$ .

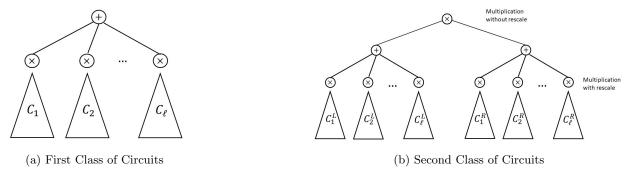


Fig. 1: A pictorial representation of the two classes of circuits we consider.

Specifically, the lattice reduction attack for circuits of this class can be analyzed by instantiating Lemma 6.1 with the following parameter settings.

- $\begin{array}{l} -\ \sigma_{h_s}^2 = \frac{\ell}{12} \\ -\ \sigma_{h_e}^2 = 0 \\ -\ \sigma_{\epsilon}^2 \ \ \mbox{is set to the noise-flooding noise. The variance of the noise already present in the ciphertext can \\ \end{array}$ be computed by taking the noise in each ciphertext before addition (which Section 4.5.2 provides) and multiplying by  $\ell$ .

#### The Second Class of Circuits and Lattice Reduction Attacks 9.2

In Figure 1b we present the second class of circuits we consider. The circuits  $C_1^L, \ldots, C_\ell^L, C_1^R, \ldots, C_\ell^R$  that are depicted each consist of  $\log(r)$  levels of multiplications as well as any number of additions. The final gate in each of the circuits  $C_1^L, \ldots, C_\ell^L, C_1^R, \ldots, C_\ell^R$  is a multiplication with rescale. Note that the noise after multiplication with rescale in circuit  $C_i^L$  (resp.  $C_i^R$ ) is dominated by  $\delta_1^{L,i} \cdot s + \delta_0^{L,i}$  (resp.  $\delta_1^{R,i} \cdot s + \delta_0^{R,i}$ ) (see Section 4.5.2), where  $\delta_0^{L,i}, \delta_1^{L,i}$  (resp.  $\delta_0^{R,i}, \delta_1^{R,i}$ ) are distributed as uniform random variables in the range [-0.5, 0.5]. Thus, after the summation gates on the second level from the top, the linear and constant coefficients of the noise corresponding to the left and right summations can be approximated by Gaussian random variables  $G_{L,1}, G_{L,0}, G_{R,1}, G_{R,0}$  with mean 0 and variance  $\frac{\ell}{12}$ .

These outputs are then multiplied via a multiplication without rescale gate. For most parameter settings, the dominating terms of the error after the final multiplication without rescale will correspond to  $\frac{m^r}{4r-1}$ .  $(G_{L,1}+G_{R,1})\cdot s$ . Further, the dominating linear coefficients of s are again (well approximated by) a Gaussian of variance  $\sigma_{h_s}^2 = \frac{\ell}{6} \cdot (\frac{m^r}{\Delta^{r-1}})^2$ . Since the error term does not include information about e, we can set  $\sigma_{h_e}^2 = 0$ .

We compute the noise variance that is already present in the ciphertext, as a contribution of the following terms  $\frac{m^r}{\Delta^{r-1}} \cdot (G_{L,0} + G_{R,0}), \frac{m^r}{\Delta^{r-1}} \cdot (G_{L,1} + G_{R,1}) \cdot s, (G_{L,1} \cdot G_{R,1}) \cdot s^2, (G_{L,0} \cdot G_{R,1}) \cdot s, (G_{L,1} \cdot G_{R,0}) \cdot s, G_{L,0} \cdot G_{R,0}$ . Since the covariance of the above terms is 0, the total variance is the sum of the variances each term above. Recall that  $G_{L,1}, G_{L,0}, G_{R,1}, G_{R,0}$  are Gaussian random variables with mean 0 and variance  $\frac{\ell}{12}$ .

Since the arithmetic is coordinate-wise on the canonical space, it suffices to consider the arithmetic of i-th component of each vector in  $\mathbb{C}^{\mathbb{Z}_m^*}$ . Specifically, we compute the variance of the real and imaginary coordinate of each vector. Since all pairs among all the resulting real and imaginary coordinates have covariance of 0, and since there is an isometry (orthogonal transformation scaled by  $\frac{1}{\sqrt{n}}$ ) from the vector consisting of the real/imaginary parts of the canonical embedding multiplied by  $\sqrt{2}$ , we can obtain the coordinate-wise variance in the coefficient embedding by scaling the results we obtain by  $\frac{2}{n}$ 

Contribution of  $\frac{m^r}{\Delta^{r-1}} \cdot (G_{L,0} + G_{R,0})$ . The variance is immediately computed as  $2(\frac{m^r}{\Delta^{r-1}})^2 \cdot \frac{\ell}{12} \cdot \sigma_s^2$ .

Contribution of  $\frac{m^r}{\Delta^{r-1}} \cdot G_{L,1} \cdot s$ . Note that the contribution of  $\frac{m^r}{\Delta^{r-1}} \cdot G_{R,1} \cdot s$  is the same as the above. By symmetry, we only need to compute the variance of the real part of the multiplication.

$$\operatorname{Var}\left[G_{L,1,i,\operatorname{Re}}s_{i,\operatorname{Re}} - G_{L,0,i,\operatorname{Im}}s_{i,\operatorname{Im}}\right]$$
$$= 2 \cdot \frac{n}{2} \cdot \frac{\ell}{12} \cdot \frac{n\sigma_s^2}{2}$$

In the coefficient domain, the total contribution will be  $(\frac{m^r}{\Delta^{r-1}})^2 \cdot \frac{n}{4} \cdot \frac{\ell}{12} \cdot \sigma_s^2$ .

Contribution of  $(G_{L,1} \cdot G_{R,1}) \cdot s^2$ . By symmetry, we only need to compute the variance of the real part of the multiplication.

$$\begin{aligned} & \operatorname{Var} \bigg[ \left( G_{L,1,i,\operatorname{Re}} G_{R,1,i,\operatorname{Re}} - G_{L,1,i,\operatorname{Im}} G_{R,1,i,\operatorname{Im}} \right) \left( s_{i,\operatorname{Re}}^2 - s_{i,\operatorname{Im}}^2 \right) \\ & - 2 s_{i,\operatorname{Re}} s_{i,\operatorname{Im}} \left( G_{L,1,i,\operatorname{Re}} G_{R,1,i,\operatorname{Im}} + G_{L,1,i,\operatorname{Im}} G_{R,1,i,\operatorname{Re}} \right) \bigg] \\ & = \operatorname{Var} \big[ G_{L,1,i,\operatorname{Re}} G_{R,1,i,\operatorname{Re}} s_{i,\operatorname{Re}}^2 - G_{L,1,i,\operatorname{Re}} G_{R,1,i,\operatorname{Re}} s_{i,\operatorname{Im}}^2 - G_{L,1,i,\operatorname{Im}} G_{L,1,i,\operatorname{Im}} s_{i,\operatorname{Re}}^2 \\ & + G_{L,1,i,\operatorname{Im}} G_{L,1,i,\operatorname{Im}} s_{i,\operatorname{Im}}^2 - 2 s_{i,\operatorname{Re}} s_{i,\operatorname{Im}} G_{L,1,i,\operatorname{Re}} G_{R,1,i,\operatorname{Im}} - 2 s_{i,\operatorname{Re}} s_{i,\operatorname{Im}} G_{L,1,i,\operatorname{Re}} G_{R,1,i,\operatorname{Im}} G_{R,1,i,\operatorname{Im}} G_{R,1,i,\operatorname{Re}} \bigg] \\ & = 4 \mathbb{E} \left[ G_{L,1,i,\operatorname{Re}}^2 (G_{R,1,i,\operatorname{Re}})^2 s_{i,\operatorname{Re}}^4 \right] + 8 \mathbb{E} \left[ s_{i,\operatorname{Re}}^2 s_{i,\operatorname{Im}}^2 G_{L,1,i,\operatorname{Re}}^2 (G_{R,1,i,\operatorname{Im}})^2 \right] \\ & = 4 \left( \frac{n}{2} \cdot \frac{\ell}{12} \right)^2 \cdot \frac{3n^2}{4} \sigma_s^4 + 8 \left( \frac{n}{2} \sigma_s^2 \right)^2 \left( \frac{n}{2} \cdot \frac{\ell}{12} \right)^2 \\ & = \frac{5}{4} n^4 \left( \frac{\ell}{12} \right)^2 \sigma_s^4 \end{aligned}$$

In the coefficient domain, the contribution will be  $\frac{5}{2}n^3\left(\frac{\ell}{12}\right)^2\sigma_s^4$ .

Contribution of  $(G_{L,0} \cdot G_{R,1}) \cdot s$ . By symmetry, we only need to compute the variance of the real part of the multiplication.

$$\begin{aligned} \operatorname{Var} \left[ G_{L,0,i,\operatorname{Re}} G_{R,1,i,\operatorname{Re}} s_{i,\operatorname{Re}} - G_{L,0,i,\operatorname{Im}} G_{R,1,i,\operatorname{Im}} s_{i,\operatorname{Re}} \right. \\ & - G_{L,0,i,\operatorname{Im}} G_{R,1,i,\operatorname{Re}} s_{i,\operatorname{Im}} - G_{L,0,i,\operatorname{Re}} G_{R,1,i,\operatorname{Im}} s_{i,\operatorname{Im}} \right] \\ &= 4 \cdot \left( \frac{n}{2} \cdot \frac{\ell}{12} \right)^2 \cdot \frac{n \sigma_s^2}{2} \end{aligned}$$

In the coefficient domain, the contribution will be  $n^2 \left(\frac{\ell}{12}\right)^2 \sigma_s^2$ .

Contribution of  $G_{L,0} \cdot G_{R,0}$ . By symmetry, we only need to compute the variance of the real part of the multiplication.

$$\operatorname{Var}\left[G_{L,0,i,\operatorname{Re}}G_{R,0,i,\operatorname{Re}} - G_{L,0,i,\operatorname{Im}}G_{R,1,i,\operatorname{Im}}\right]$$
$$= 2 \cdot \left(\frac{n}{2} \cdot \frac{\ell}{12}\right)^{2}$$

In the coefficient domain, the contribution will be  $n \cdot \left(\frac{\ell}{12}\right)^2$ .

Total noise present. The total noise in the ciphertext has variance:

$$2\left(\frac{m^r}{\Delta^{r-1}}\right)^2 \cdot \frac{\ell}{12} \cdot \sigma_s^2 + \left(\frac{m^r}{\Delta^{r-1}}\right)^2 \cdot \frac{n \cdot \ell}{6} \cdot \sigma_s^2 + \frac{5}{2}n^3 \left(\frac{\ell}{12}\right)^2 \sigma_s^4 + 2n^2 \left(\frac{\ell}{12}\right)^2 \sigma_s^2 + n \cdot \left(\frac{\ell}{12}\right)^2 \sigma_s^2 + n$$

Obtaining the hardness estimates. We can now apply Lemma 6.1 with the following parameter settings:

$$\begin{array}{l} -\ \sigma_{h_s}^2 = \frac{\ell}{6} \cdot \left(\frac{m^r}{\varDelta^{r-1}}\right)^2 \\ -\ \sigma_{h_e}^2 = 0 \end{array}$$

 $-\sigma_{\epsilon}^2$  is set to the noise-flooding noise plus an additional  $\frac{5}{2}n^3\left(\frac{\ell}{12}\right)^2\sigma_s^4 + 2n^2\left(\frac{\ell}{12}\right)^2\sigma_s^2$ , the noise from the quadratic terms and the linear but non-Gaussian terms (which comes from the terms of the form  $(G_{L,0}\cdot G_{R,1})\cdot s$ ).

Note that the noise-flooding noise has variance at least  $(\frac{m^r}{\Delta^{r-1}})^2 \cdot \frac{n \cdot \ell}{6} \cdot \sigma_s^2$ , since the noise already in the ciphertext is larger than this quantity. Thus, for  $n \in \mathbb{N}$ , when

$$\left(\frac{m^r}{\Delta^{r-1}}\right)^2 \gg \frac{5}{2}n^2 \cdot \frac{\ell}{24} + 2 \cdot n \cdot \frac{\ell}{24} > \frac{9}{2}n^2 \cdot \frac{\ell}{24},\tag{16}$$

and m achieves the maximum allowed magnitude  $B_{msg}$  of each coordinate in the *encoded* plaintext (in which the message is viewed as a vector in the canonical embedding and is scaled up by  $\Delta$ ), we have that the noise-flooding noise dominates the additional  $\frac{5}{2}n^3\left(\frac{\ell}{12}\right)^2\sigma_s^4+2n^2\left(\frac{\ell}{12}\right)^2\sigma_s^2$ . Typically, after encoding, the maximum allowed magnitude of m in the canonical embedding is  $\approx \Delta$ . Thus, (16) is satisfied when  $\Delta \geq \frac{3n}{4} \cdot \sqrt{\frac{\ell}{3}}$ , which is typically satisfied for most parameter settings (in fact,  $\Delta$  is typically far larger).

Thus, we can plug the above parameter settings into Lemma 6.1 to obtain the hardness estimates for these circuits under a lattice reduction attack.

## 9.3 Guessing Attack for Class 1 and 2 Circuits

Now that we have determined  $\sigma_{h_s}^2$ ,  $\sigma_{h_e}^2$ , and  $\sigma_{\epsilon}^2$  for Class 1 and Class 2 circuits, we can use those values to derive formulas for the concrete security for guessing and hybrid attacks as well.

Recall that for Class 1 and Class 2 circuits, the hints are only on the **s** coordinates. So  $\Sigma'$  is a block matrix where the lower right hand  $n \times n$  submatrix is a diagonal matrix with diagonal  $(\sigma_e^2, \ldots, \sigma_e^2)$  and the upper left hand  $n \times n$  submatrix has n eigenvalues of the form  $[(\alpha_{2i+1}, \alpha_{2i+2})]_{i \in [n/2]}$  and for all  $i \in [n/2]$ ,  $\alpha_{2i+1} = \alpha_{2i+2}$ . Further, for each  $i \in [n/2]$ ,

$$\alpha_{2i+1} = \frac{\sigma_s^2}{1 + \frac{\sigma_s^2 \cdot R_{1,i}}{2\sigma_s^2}}.$$

Since with all but  $e^{-11}$  probability  $^{14}$ ,  $R_{1,i} \ge (2t - 6.63\sqrt{2t}) \cdot n\sigma_{h_s}^2$ , we have that with probability  $1 - n/2 \cdot e^{-11}$  all eigenvalues are less than

$$\sigma_{max}^{2} \le \frac{\sigma_{s}^{2}}{1 + \frac{\sigma_{s}^{2} \cdot (2t - 6.63\sqrt{2t}) \cdot n\sigma_{h_{s}}^{2}}{2\sigma^{2}}},\tag{17}$$

and so for every standard basis vector  $\mathbf{e}_i$ ,  $\mathbf{e}_i \mathbf{\Sigma}_S' \mathbf{e}_i^T \leq \sigma_{max}^2$ .

Finally, using the same techniques as above, this means that the guessing probability is at least

$$\operatorname{erf}\left(\frac{0.5}{\sqrt{2\sigma_{max}^2}}\right)^n. \tag{18}$$

Thus the total success probability of the attack is  $\text{erf}\left(\frac{0.5}{\sqrt{2\sigma_{max}^2}}\right)^n - n/2 \cdot e^{-11}$ . We note that for up to parameter  $n = 32768, \, n/2 \cdot e^{-11} \leq 0.28.^{15}$ 

<sup>&</sup>lt;sup>14</sup> For the parameter sets with n = 131072, we increase 6.63 below to 7.2 and decrease the probability to  $e^{-13}$ .

<sup>&</sup>lt;sup>15</sup> And for n = 131072,  $n/2 \cdot e^{-13} < 0.15$ .

#### 9.4 Hybrid Attack for Class 1 and 2 Circuits

Again, the attack for both Class 1 and Class 2 circuits is the same, with the only difference being the settings of  $\sigma_{h_s}^2$ ,  $\sigma_{h_s}^2$ , and  $\sigma_{\epsilon}^2$  in the two cases.

The guessing strategy for the hybrid attack is as follows: For each i, the adversary computes  $\mathbf{e}_i \mathbf{\Sigma}' \mathbf{e}_i^T$  and guesses  $\mu_i$  for the g number of indices i with the minimum values of  $\mathbf{e}_i \mathbf{\Sigma}' \mathbf{e}_i^T$ , where g is the maximum value such that

$$\operatorname{erf}\left(\frac{0.5}{\sqrt{2\sigma_{max}^2}}\right)^g \ge p,\tag{19}$$

for some probability threshold p. These guesses are made and incorporated as perfect hints. After this process, the covariance matrix is a principal submatrix of  $\Sigma'_S$  of dimension  $(n-g)\times(n-g)$ , which we denote by  $\Sigma''_S$ . Similarly, the lattice reduces dimension by g and its volume remains the same.

Let  $\alpha_1, \ldots, \alpha_g$  be the g minimum eigenvalues of  $\Sigma_S'$ . Using the Eigenvalue Interlacing Theorem [22], we have that  $\det(\Sigma_S'') \leq \frac{\det(\Sigma')}{\alpha_1 \cdots \alpha_g}$ . We therefore need a lower bound on  $\alpha_1 \cdots \alpha_g$ . Since with all but  $e^{-11}$  probability  $R_{1,i} \leq (2t + 6.63\sqrt{2t} + 22) \cdot n\sigma_{h_s}^2$ , we have that with probability  $1 - n/2 \cdot e^{-11}$  all eigenvalues are greater than

$$L = \frac{\sigma_s^2}{1 + \frac{\sigma_s^2 \cdot (2t + 6.63\sqrt{2t} + 22) \cdot n\sigma_{h_s}^2}{2\sigma^2}}.$$
 (20)

Combining the above, we have that with at least  $p - n \cdot e^{-11}$  probability, all g number of guesses are correct, and

$$\det(\mathbf{\Sigma}'') \le \frac{\det(\mathbf{\Sigma}')}{L^g}.$$
 (21)

We note that for the maximum setting of parameters n = 32768,  $n \cdot e^{-11} \le 0.55$ . Further,  $\det(\Sigma'')$  can be computed by plugging the parameter settings from Sections 9.1 and 9.2 into Lemma 6.1. Thus, we can use (21) to estimate the hardness of the residual instance (after guesses) under a lattice reduction attack.

## 10 Experiments

#### 10.1 Experimental Set-Up

Parameter sets. We consider the parameter sets proposed by the homomorphicencryption.org standards [2], which were proposed with target security levels of 128, 192 or 256 bits. We update the target estimates using the concrete hardness given by the tool of [16]. This is presented in the column "Original Security" in all the tables below. An entry of x/y represents the original target security level x, and y represents the concrete (updated) security level. The standards only consider a ring dimension of up to n = 32768, i.e.  $\log_2(n) = 15$ , but some FHE applications may require a larger ring dimension, up to  $\log_2(n) = 17$ . We additionally provide estimates for the concrete security of CKKS for values of  $\log_2(n) = 17$  by using the parameters given in [27].

Experimental validation. We first provide experimental validation of Lemma 6.1, in Section 10.2. We also provide concrete security estimation for provably secure (statistical) noise-flooding, as presented in [26]. We provide these as a baseline, and to validate our methods. Since there is no reduction in security when applying statistical noise-flooding, those results are presented in Appendix B.

Concrete security experiments set-up. Then, we consider the following experiments. We consider a lattice reduction attack, a guessing attack and a hybrid attack, as outlined in Sections 6, 7 and 8, respectively. We consider these on three types of circuit: the identity circuit, the class of circuits C1 and the class of circuits C2. Recall that these are described in Section 9.

<sup>&</sup>lt;sup>16</sup> For the parameter sets with n = 131072, we increase 6.63 below to 7.2 and decrease the probability to  $e^{-13}$ .

<sup>&</sup>lt;sup>17</sup> And for n = 131072,  $n \cdot e^{-13} \le 0.30$ .

<sup>&</sup>lt;sup>18</sup> Our analysis may give slightly different concrete hardness estimates than the LWE Estimator [4], since [16] takes into account the ellipsoidal distribution of the original secret/error.

Noise-flooding countermeasures. We use the results of [14] to estimate the output variance of the noise  $\rho_{\text{circ}}^2$ , where circ is one of Identity, C1 or C2. We then consider noise-flooding by  $\rho_{\text{circ}}^2$ ,  $100 \cdot \rho_{\text{circ}}^2$  and  $t \cdot \rho_{\text{circ}}^2$ , for t is the number of decryption queries. For guessing attacks, we do not include results for noise-flooding variance of  $t \cdot \rho_{\text{circ}}^2$ , since in this case, the guessing probability does not go above  $10^{-200}$  for any parameter set. Similarly, for hybrid attacks, we do not include results for noise-flooding variance of  $t \cdot \rho_{fresh}^2$ , since no coordinates can be guessed with high confidence for any parameter set, and so the attack is equivalent to a lattice reduction attack <sup>19</sup>.

## 10.2 Experimental Validation of Lemma 6.1

We first provide a verification of the theoretical results from Section 6, to demonstrate that the estimations hold in practice. In particular, Lemma 6.1 assumes that the distribution of the coefficients of  $\mathbf{e}_1^j$  and  $\mathbf{v}^j$  are independent Gaussians, while in practice this is not the case. The quantity of interest is  $\det(\Sigma')$ , as defined in Section 6. In the proof of Lemma 6.1, we use the following fact:

$$\det(\mathbf{\Sigma}^{\prime \sim}) = \frac{\det(\mathbf{H}\mathbf{\Sigma}\mathbf{H}^T + \mathbf{\Sigma}_{\varepsilon})}{\det(\mathbf{\Sigma}_{\varepsilon})\det(\mathbf{\Sigma})} = \frac{\det\left(\mathbf{I}_{2n} + \frac{1}{\sigma_{\varepsilon}^2}\mathbf{\Sigma}^{1/2}\mathbf{H}\mathbf{H}^T\mathbf{\Sigma}^{1/2}\right)}{\det(\mathbf{\Sigma})}.$$
 (22)

In order to validate the canonical embedding transformation used in the analysis of Lemma 6.1, we sample a random hint matrix  $\mathbf{H}$ , directly compute  $\mathbf{I}_{2n} + \mathbf{\Sigma}^{1/2}\mathbf{H}\mathbf{H}^T\mathbf{\Sigma}^{1/2}/\sigma_{\epsilon}^2$ , and calculate its determinant. In order to construct the hint matrix, we sample  $\mathbf{e}_1^j \leftarrow \chi$  and  $\mathbf{v}^j \leftarrow S$  as defined in Appendix 4. We perform this experiment for various settings for the dimension of the LWE secret and error, and for various numbers of hints applied. For each parameter set, we perform 256 trials and take the average of the results in order to compare to the expected value predicted by Lemma 6.1. Figure 2 reports the experimental results, which very closely match the predictions. Notably, we see that the predictions become more accurate as the number of applied hints increases.

We perform this experiment using the SageMath library and run the calculations on an Intel Ice Lake XCC server. Calculating the determinant for larger parameter sets proves computationally infeasible with our experimental setup due to the extreme scaling, as each trial requires multiplying matrices of size  $2n \times tn$  and  $tn \times 2n$ , as well as calculating the determinant of a matrix of size  $2n \times 2n$ , where n is the dimension and t is the number of hints. Additionally, in order to accurately calculate the final determinant, the numerical values within the matrix require increasingly high floating-point precision (e.g. hundreds or even thousands of bits), further slowing the computation. Our experiments take roughly a week to verify the largest parameter set in Figure 2 (n = 256, t = 16).

## 10.3 Concrete Security of Lattice Attacks on Identity Circuits

We begin by considering a lattice-reduction attack where the adversary may request any number of decryptions of fresh ciphertexts (i.e. evaluation of the identity circuit on a fresh ciphertext) with various noise-flooding levels. See Figure 4. To calculate the concrete hardness, we apply Lemma 6.1 to obtain the expected volume and dimension of the lattice after hints are integrated and homogenization/isotropization is completed. As in [16], after homogenization/isotropization are performed, the hardness estimates for BKZ require only the volume and dimension of the lattice. These are reported in the final column.

#### 10.4 Concrete Security of Guessing Attacks on Identity Circuits

Next we consider a guessing-only attack, where the adversary may request any number of decryptions of fresh ciphertexts (i.e. evaluation of the identity circuit on a fresh ciphertext) with various noise-flooding levels. See Figure 5. In this attack, the adversary requests enough decryptions so that n LWE secret/error coordinates can be guessed correctly with high probability. Once these coordinates are guessed correctly, the LWE system of equations has a unique solution which can be recovered efficiently using Gaussian elimination. To determine the number of decryptions required to recover the LWE secret/error with some threshold probability, we apply Lemma 7.1 and (10).

After ~ 200 million decryption queries, the estimated variance does not go below 3.6 for identity circuits, and after ~ 100 million decryption queries does not go below 0.33 and 0.36 for C1 and C2 circuits, respectively.

Dim	Num Hints	Predicted Determinant	Experimental Determinant
64	16	708.60	708.76
64	32	799.19	799.28
64	64	888.87	889.14
64	128	978.08	978.10
64	256	1067.04	1067.00
64	512	1155.89	1155.87
128	16	1594.55	1591.58
128	32	1175.78	1775.55
128	64	1955.17	1954.82
128	128	2133.59	2133.49
128	256	2311.52	2311.44
128	512	2489.22	2489.23
256	16	3543.88	3539.04

Fig. 2: Summary of results for experimental validation of Lemma 6.1. Each parameter set is specified by the dimension of the LWE secret/error (column 1) and the number of hints applied (column 2). The third column indicates the (ln of) the expected value of the determinant as predicted by Lemma 6.1. The final column reports the determinant calculated by performing the experiment, as averaged over 256 trials.

#### 10.5 Concrete Security of Hybrid Attacks on Identity Circuits

Here we consider a hybrid attack, where the adversary may request some number of decryptions of fresh ciphertexts (i.e. evaluation of the identity circuit on a fresh ciphertext) with various noise-flooding levels. See Figure 6. The adversary requests enough decryptions so that some number of LWE secret/error coordinates can be guessed correctly with high probability. The adversary then integrates these guesses into its DBDD instance as perfect hints (as in [16]). Finally, the adversary performs homogenization/isotropization to obtain an SVP instance, and uses a BKZ solver to recover the LWE secret/error. For a fixed number of decryptions, we use (11) to determine the number of guesses g that can be made such that all guesses are correct with high probability. The dimension of the lattice reduces by g, and we compute the volume of the resulting lattice by applying (15). As in [16], after homogenization/isotropization are performed, the hardness estimates for BKZ require only the volume and dimension of the lattice. These are reported in the final column.

## 10.6 Concrete Security of Lattice Attacks on Class 1 and 2 Circuits

This is the same attack as in Section 10.3, except the adversary requests decryptions of ciphertexts corresponding to the evaluation of a Class 1 or Class 2 circuit (see Sections 9.1 and 9.2) on fresh ciphertexts. To calculate the concrete hardness, we apply Lemma 6.1 to obtain the expected volume and dimension of the lattice after hints are integrated with the parameter settings for  $\sigma_{h_s}^2, \sigma_{h_e}^2, \sigma_{\epsilon}^2$  given in Section 9.1 or Section 9.2 The results are reported in Figures 7 and 8 in Appendix B.

#### 10.7 Concrete Security of Guessing Attacks on Class 1 and 2 Circuits

This is the same attack as in Section 10.4, except the adversary requests decryptions of ciphertexts corresponding to the evaluation of a Class 1 or Class 2 circuit (see Sections 9.1 and 9.2) on fresh ciphertexts. To determine the number of decryptions required to recover the LWE secret with high probability, we apply (18) with the settings of  $\sigma_{h_s}$ ,  $\sigma_{h_e}$ ,  $\sigma_{\epsilon}^2$  given in Section 9.1 or Section 9.2. The results for various noise-flooding levels are reported in Figures 9 and 10 in Appendix B.

## 10.8 Concrete Security of Hybrid Attacks on Class 1 and 2 Circuits

This is the same attack as in Section 10.5, except the adversary requests decryptions of ciphertexts corresponding to the evaluation of a Class 1 or Class 2 circuit (see Sections 9.1 and 9.2) on fresh ciphertexts.

For a fixed number of decryptions, we use (11), with the settings of  $\sigma_{h_2}^2$ ,  $\sigma_{h_e}^2$ , and  $\sigma_{\epsilon}^2$  given in Section 9.1 or Section 9.2, to determine the number of guesses g that can be made such that all guesses are correct with high probability. The dimension of the lattice reduces by g, and we compute the volume of the resulting lattice by applying (15), with the settings of  $\sigma_{h_2}^2$ ,  $\sigma_{h_e}^2$ , and  $\sigma_{\epsilon}^2$  given in Section 9.1 or Section 9.2. The results are reported in Figures 11 and 12 in Appendix B.

#### 11 Discussion of the Results

Trends for noise-flooding level of  $\rho_{\rm circ}^2$ . Our experimental data is summarized via the graphs in Figure 3. Figure 3(a) shows the reduction in bit security for a lattice reduction attack when the adversary obtains 1000 decryptions of identity, Class 1, and Class 2 circuits with noise-flooding level  $\rho_{\rm circ}^2$  equal to the noise already present in the ciphertext. We note that the graph exhibits a greater reduction in bit-security for identity circuits vs. Class 1 and 2 circuits. We believe the reason is that hints for identity circuits involve all 2n coordinates in the LWE secret/error, so the variance of all 2n coordinates is reduced after each hint, whereas hints for Class 1 and Class 2 circuits involve only the n coordinates from the LWE secret, so only the variance of these n coordinates is reduced. We also note that there is a greater security reduction for higher target security level vs. lower target security level. E.g., for the lattice reduction attack, we see that for  $\log_2(n) = 10$ , identity circuits, and for a security level target of 192, the value of the bit security is reduced by slightly over 70 bits. On the other hand, for the same circuit and target security level and the same attack, for  $\log_2(n) = 15$ , the reduction in the bit security level is less than 5 bits. In fact, the reduction in security seems highly correlated with decrease in modulus. When fixing the dimension n, target security level of 192 have smaller modulus  $q_L$ , compared to target security level of 128 and as the modulus  $q_L$  becomes smaller, "hints" obtained from decryption have more of an impact on the bit-security for lattice reduction attacks. The same trends can be seen in the Hybrid attack.

Figure 3(b) shows the number of queries required for guessing n coordinates with high probability for identity, Class 1 and Class 2 circuits. We note that guessing attacks perform significantly better for Class 1 and 2 circuits versus identity circuits. For identity circuits, there are a total of 2n eigenvalues that are reduced by obtaining hints, but n of these eigenvalues have relatively larger expectation, while n have smaller expectation (this is because for identity circuits, hints correspond to linear combinations of both the s and e variables, in which the s variables have variance 2/3, while the e variables have variance  $3.2^2$ ). The eigenvectors corresponding to these eigenvalues do not align with the standard basis. Therefore, for purposes of fast estimates, we only take into account the trace (i.e. sum of the eigenvalues) and, given trace T, we argue that the variance of the n secret or error coordinates with smallest variance is at most T/(2n). However, in practice, the n coordinates with the smallest variance may have variance significantly smaller than T/(2n). For Class 1 and 2 circuits, hints correspond to linear combinations of only the s variables from the LWE instance. Thus, we restrict our attention to a subspace with only s eigenvalues. All of these eigenvalues have the same distribution, and our proof shows that s the eigenvalues are less than maximum value s and s are less than large s and s are l

Figure 3(c) shows the reduction in bit-security for a hybrid attack when considering an adversary who obtains decryptions of identity, Class 1, and Class 2 circuits. Figure 3(d) shows the number of queries obtained in each of these attacks. We chose the number of queries for the identity, Class 1, and Class 2 circuits so that a significant number of guesses can be made for each parameter set (otherwise the attack will be very similar to a lattice reduction attack). Based on the discussion above, this means that the number of queries required is far higher for identity circuits than Class 1 and Class 2 circuits. Thus, after guesses are made, the residual instance has lower variance in the case of identity circuits (since more hints have been incorporated, with each hint slightly reducing the variance). This explains why for approximately the same number of guesses, the reduction in bit-security is greater for identity circuits versus Class 1 and Class 2 circuits, as can be observed from the graph.

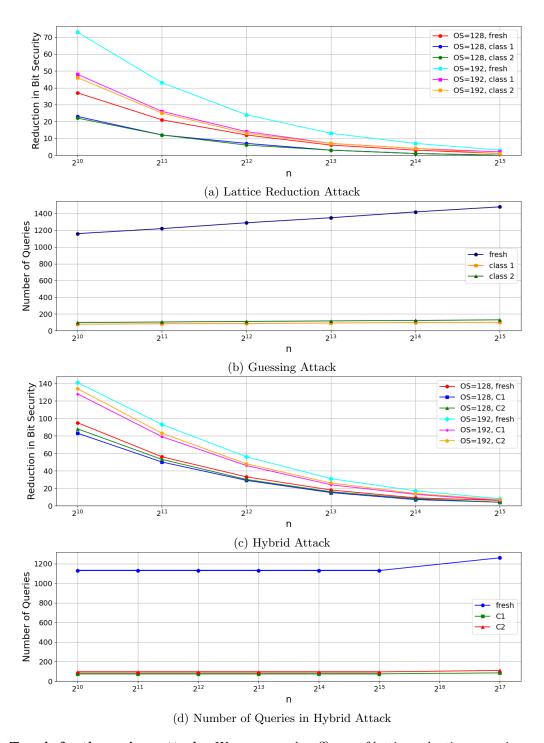


Fig. 3: Trends for the various attacks. We compare the efficacy of lattice reduction, guessing, and hybrid attacks for various parameter sets, and for identity, Class 1, and Class 2 circuits with noise-flooding level equal to  $\rho_{\text{fresh}}^2$ ,  $\rho_{\text{C1}}^2$ , and  $\rho_{\text{C2}}^2$ , respectively. (a) Shows the reduction in bit security for a lattice reduction attack against an adversary who obtains 1000 decryptions; (b) Shows the number of queries required for guessing n coordinates with probability at least 0.80. (c) Shows the reduction in bit security for a hybrid attack against an adversary who obtains a variable number of decryptions. The number of decryption queries for each parameter set is displayed in (d).

Trends across various noise-flooding levels. We first validate that there is no security drop in our experiments when using the statistically-secure noise-flooding levels proposed in [26]. Our results are presented in Figures 13, 14 and 15 in Appendix B. Indeed, we see in these tables that there is no reduction in either the security level or in the bikz for any parameter setting.

Recall that we investigate the effectiveness of noise-flooding levels  $\rho_{\rm circ}^2$ ,  $100 \cdot \rho_{\rm circ}^2$ , and  $t \cdot \rho_{\rm circ}^2$ , where t is the number of decryption queries, circ is one of Identity, C1 or C2, and  $\rho_{\rm circ}^2$  is the noise variance present in the ciphertext. As expected, we see that the biggest drop in bit security is observed when noise-flooding by  $\rho_{\rm circ}^2$ , across all parameter sets and across all circuits.

In contrast, we observe that noise-flooding by  $t \cdot \rho_{\text{circ}}^2$  leads to a very low reduction in the security level, if at all. As opposed to a 70-bit security drop seen for lattice attacks with  $\log_2(n) = 10$  and 192-bit security for identity circuits with noise level  $\rho_{\text{fresh}}^2$ , we see in Figure 4 that when noise-flooding by  $t \cdot \rho_{\text{fresh}}^2$ , the security level drops by only a few bits. Further, as the value of  $\log_2(n)$  (and thus also  $q_L$  increases), the security level drop decreases. We see for example in Figure 4 that for  $\log_2(n) = 17$ , there is no change in the security level.

Conclusions: We observe that, in practice, there is essentially no reduction in concrete security when noise flooding with variance  $t \cdot \rho_{\rm circ}^2$ , where t is the number of decryption queries available to the adversary, and  $\rho_{\rm circ}^2$  is the variance of the noise, as predicted by an average-case noise analysis. One may also consider noise flooding by  $\alpha \cdot t \cdot \rho$ , where  $0 < \alpha < 1$ , if it is acceptable to have the security level drop by a few bits. There is no definitive setting of  $\alpha$  which is "best," and one can rather think of  $\alpha$  as a parameter to be fine-tuned depending on the application. In particular, one can think of increasing  $\alpha$  as a way to allow for more decryption queries, or to reduce the message precision loss. We obtain the latter as: noise-flooding by  $x \cdot \rho_{\rm circ}^2$  incurs an additional loss of  $\frac{1}{2}\log_2(x+1)$  bits in the message precision.

Finally, we note that the techniques developed in this paper, as well as the experimental results presented, can be used as a way to establish key refreshing policies in a concrete application. E.g., if the noise level is set to  $\alpha \cdot t \cdot \rho$ , the keys should be refreshed after releasing t number of decryptions. Thus, there can be a tradeoff among frequency of key refresh, an acceptable precision loss, and an acceptable drop in bit-security.

## Acknowledgements

We thank Tikaram Sanyashi and Alexander Viand for helpful discussions and comments.

## References

- 1. Ahmad Al Badawi, Jack Bates, Flavio Bergamaschi, David Bruce Cousins, Saroja Erabelli, Nicholas Genise, Shai Halevi, Hamish Hunt, Andrey Kim, Yongwoo Lee, et al. Openfhe: Open-source fully homomorphic encryption library. In *Proceedings of the 10th Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, pages 53–63, 2022.
- 2. Martin Albrecht, Melissa Chase, Hao Chen, Jintai Ding, Shafi Goldwasser, Sergey Gorbunov, Shai Halevi, Jeffrey Hoffstein, Kim Laine, Kristin Lauter, Satya Lokam, Daniele Micciancio, Dustin Moody, Travis Morrison, Amit Sahai, and Vinod Vaikuntanathan. Homomorphic encryption standard. homomorphicencryption.org, 2018.
- 3. Martin R. Albrecht, Florian Göpfert, Fernando Virdia, and Thomas Wunderer. Revisiting the expected cost of solving uSVP and applications to LWE. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology ASIACRYPT 2017, Part I*, volume 10624 of *Lecture Notes in Computer Science*, pages 297–322, Hong Kong, China, December 3–7, 2017. Springer, Heidelberg, Germany.
- 4. Martin R. Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. Cryptology ePrint Archive, Report 2015/046, 2015. https://eprint.iacr.org/2015/046.
- Andreea Alexandru, Ahmad Al Badawi, Daniele Micciancio, and Yuriy Polyakov. Application-aware approximate homomorphic encryption: Configuring fhe for practical use. Cryptology ePrint Archive, Paper 2024/203, 2024. https://eprint.iacr.org/2024/203.
- 6. Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange A new hope. In Thorsten Holz and Stefan Savage, editors, *USENIX Security 2016: 25th USENIX Security Symposium*, pages 327–343, Austin, TX, USA, August 10–12, 2016. USENIX Association.
- 7. Private Communication, anonymized for submission.
- 8. Diego F. Aranha, Anamaria Costache, Antonio Guimarães, and Eduardo Soria-Vazquez. Heliopolis: Verifiable computation over homomorphically encrypted data from interactive oracle proofs is practical. Cryptology ePrint Archive, Paper 2023/1949, 2023. https://eprint.iacr.org/2023/1949.

- 9. Jonathan Bootle, Claire Delaplace, Thomas Espitau, Pierre-Alain Fouque, and Mehdi Tibouchi. LWE without modular reduction and improved side-channel attacks against BLISS. Cryptology ePrint Archive, Report 2018/822, 2018. https://eprint.iacr.org/2018/822.
- Jung Hee Cheon, Hyeongmin Choe, Alain Passelègue, Damien Stehlé, and Elias Suvanto. Attacks against the indepa-d security of exact flue schemes. Cryptology ePrint Archive, Paper 2024/127, 2024. https://eprint. iacr.org/2024/127.
- 11. Jung Hee Cheon, Seungwan Hong, and Duhyeong Kim. Remark on the security of CKKS scheme in practice. Cryptology ePrint Archive, Report 2020/1581, 2020. https://eprint.iacr.org/2020/1581.
- 12. Jung Hee Cheon, Andrey Kim, Miran Kim, and Yong Soo Song. Homomorphic encryption for arithmetic of approximate numbers. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology ASI-ACRYPT 2017, Part I*, volume 10624 of *Lecture Notes in Computer Science*, pages 409–437, Hong Kong, China, December 3–7, 2017. Springer, Heidelberg, Germany.
- 13. Anamaria Costache, Benjamin R. Curtis, Erin Hales, Sean Murphy, Tabitha Ogilvie, and Rachel Player. On the precision loss in approximate homomorphic encryption. Cryptology ePrint Archive, Report 2022/162, 2022. https://eprint.iacr.org/2022/162.
- Anamaria Costache, Benjamin R Curtis, Erin Hales, Sean Murphy, Tabitha Ogilvie, and Rachel Player. On the precision loss in approximate homomorphic encryption. In *International Conference on Selected Areas in Cryptography*, pages 325–345. Springer, 2023.
- Anamaria Costache, Lea Nürnberger, and Rachel Player. Optimisations and tradeoffs for HElib. In Mike Rosulek, editor, Topics in Cryptology – CT-RSA 2023, volume 13871 of Lecture Notes in Computer Science, pages 29–53, San Francisco, CA, USA, April 24–27, 2023. Springer, Heidelberg, Germany.
- 16. Dana Dachman-Soled, Léo Ducas, Huijing Gong, and Mélissa Rossi. LWE with side information: Attacks and concrete security estimation. In Daniele Micciancio and Thomas Ristenpart, editors, Advances in Cryptology CRYPTO 2020, Part II, volume 12171 of Lecture Notes in Computer Science, pages 329–358, Santa Barbara, CA, USA, August 17–21, 2020. Springer, Heidelberg, Germany.
- 17. Dana Dachman-Soled, Huijing Gong, Tom Hanson, and Hunter Kippen. Revisiting security estimation for LWE with hints from a geometric perspective. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology CRYPTO 2023*, *Part V*, volume 14085 of *Lecture Notes in Computer Science*, pages 748–781, Santa Barbara, CA, USA, August 20–24, 2023. Springer, Heidelberg, Germany.
- 18. Sanjam Garg, Aarushi Goel, and Mingyuan Wang. How to prove statements obliviously? Cryptology ePrint Archive, Paper 2023/1609, 2023. https://eprint.iacr.org/2023/1609.
- Rosario Gennaro, Craig Gentry, and Bryan Parno. Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In Tal Rabin, editor, Advances in Cryptology CRYPTO 2010, volume 6223 of Lecture Notes in Computer Science, pages 465–482, Santa Barbara, CA, USA, August 15–19, 2010. Springer, Heidelberg, Germany.
- 20. Timo Glaser, Alexander May, and Julian Nowakowski. Entropy suffices for guessing most keys. Cryptology ePrint Archive, Paper 2023/797, 2023. https://eprint.iacr.org/2023/797.
- 21. Qian Guo, Denis Nabokov, Elias Suvanto, and Thomas Johansson. Key recovery attacks on approximate homomorphic encryption with non-worst-case noise flooding countermeasures. In 33rd USENIX Security Symposium (USENIX Security 24). Philadelphia, PA: USENIX Association, 2024.
- 22. Suk-Geun Hwang. Cauchy's interlace theorem for eigenvalues of hermitian matrices. *American Mathematical Monthly*, pages 157–159, 2004.
- 23. Duhyeong Kim, Dongwon Lee, Jinyeong Seo, and Yongsoo Song. Toward practical lattice-based proof of knowledge from hint-MLWE. In Helena Handschuh and Anna Lysyanskaya, editors, Advances in Cryptology CRYPTO 2023, Part V, volume 14085 of Lecture Notes in Computer Science, pages 549–580, Santa Barbara, CA, USA, August 20–24, 2023. Springer, Heidelberg, Germany.
- 24. Rafał Latała and Dariusz Matlak. Royen's proof of the gaussian correlation inequality. In Geometric Aspects of Functional Analysis: Israel Seminar (GAFA) 2014–2016, pages 265–275. Springer, 2017.
- 25. Baiyu Li and Daniele Micciancio. On the security of homomorphic encryption on approximate numbers. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology EUROCRYPT 2021, Part I*, volume 12696 of *Lecture Notes in Computer Science*, pages 648–677, Zagreb, Croatia, October 17–21, 2021. Springer, Heidelberg, Germany.
- 26. Baiyu Li, Daniele Micciancio, Mark Schultz, and Jessica Sorrell. Securing approximate homomorphic encryption using differential privacy. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology CRYPTO 2022, Part I*, volume 13507 of *Lecture Notes in Computer Science*, pages 560–589, Santa Barbara, CA, USA, August 15–18, 2022. Springer, Heidelberg, Germany.

- 27. Oliver Masters, Hamish Hunt, Enrico Steffinlongo, Jack Crawford, Flavio Bergamaschi, Maria E. Dela Rosa, Caio C. Quini, Camila T. Alves, Feranda de Souza, and Deise G. Ferreira. Towards a homomorphic machine learning big data pipeline for the financial services sector. Cryptology ePrint Archive, Report 2019/1113, 2019. https://eprint.iacr.org/2019/1113.
- 28. Alexander May and Julian Nowakowski. Too many hints when lll breaks lwe. In Jian Guo and Ron Steinfeld, editors, *Advances in Cryptology ASIACRYPT 2023*, pages 106–137, Singapore, 2023. Springer Nature Singapore.

## A Analysis for Section 7

We let  $\hat{\Sigma}$ ,  $\hat{\Sigma}'$ ,  $\hat{H}$ ,  $\hat{\Sigma}_{\epsilon}$  correspond to the original covariance matrix, updated covariance matrix, hint, and error matrices in the canonical embedding to the Reals, scaled by  $\frac{1}{\sqrt{n}}$ . Note that  $\Sigma' = \tilde{R}\hat{\Sigma}'\tilde{R}^T$   $\Sigma = \hat{\Sigma}$ ,  $\Sigma_{\epsilon} = \hat{\Sigma}_{\epsilon}$ , and  $\mathbf{H} = \mathbf{R}_L \hat{\mathbf{H}} \mathbf{R}_R$ . Then we have

$$\begin{split} \hat{\boldsymbol{\Sigma}}' &= \boldsymbol{\Sigma} - \boldsymbol{\Sigma} \hat{\mathbf{H}} \left( \hat{\mathbf{H}}^T \boldsymbol{\Sigma} \hat{\mathbf{H}} + \boldsymbol{\Sigma}_{\epsilon} \right)^{-1} \hat{\mathbf{H}}^T \boldsymbol{\Sigma} \\ &= \boldsymbol{\Sigma}^{1/2} \Bigg( \mathbf{I} - \boldsymbol{\Sigma}^{1/2} \hat{\mathbf{H}} \boldsymbol{\Sigma}_{\epsilon}^{-1} \hat{\mathbf{H}}^T \boldsymbol{\Sigma}^{1/2} - \\ & \boldsymbol{\Sigma}^{1/2} \hat{\mathbf{H}} \boldsymbol{\Sigma}_{\epsilon}^{-1} \hat{\mathbf{H}}^T \boldsymbol{\Sigma}^{1/2} \Big( \mathbf{I} + \boldsymbol{\Sigma}^{1/2} \hat{\mathbf{H}} \boldsymbol{\Sigma}_{\epsilon}^{-1} \hat{\mathbf{H}}^T \boldsymbol{\Sigma}^{1/2} \Big)^{-1} \boldsymbol{\Sigma}^{1/2} \hat{\mathbf{H}} \boldsymbol{\Sigma}_{\epsilon}^{-1} \hat{\mathbf{H}}^T \boldsymbol{\Sigma}^{1/2} \Bigg) \boldsymbol{\Sigma}^{1/2}. \end{split}$$

We note that  $\mathbf{\Sigma}^{1/2}\hat{\mathbf{H}}\mathbf{\Sigma}_{\epsilon}^{-1}\hat{\mathbf{H}}^T\mathbf{\Sigma}^{1/2}$  is a block diagonal matrix where the *i*-th block for  $i \in [n/2]$  is of the form

$$\begin{bmatrix} a_i & 0 & c_i - d_i \\ 0 & a_i & d_i & c_i \\ c_i & d_i & b_i & 0 \\ -d_i & c_i & 0 & b_i \end{bmatrix}$$

(this is the same matrix that was computed in the proof of Lemma 6.1), and the *i*-th block of  $\left(\mathbf{I} + \mathbf{\Sigma}^{1/2} \hat{\mathbf{H}} \mathbf{\Sigma}_{\epsilon}^{-1} \hat{\mathbf{H}}^T \mathbf{\Sigma}^{1/2}\right)^{-1}$  is of the form

$$\begin{bmatrix} \tilde{a}_i & 0 & m_i \cdot c_i - m_i \cdot d_i \\ 0 & \tilde{a}_i & m_i \cdot d_i & m_i \cdot c_i \\ m_i \cdot c_i & m_i \cdot d_i & \tilde{b}_i & 0 \\ -m_i \cdot d_i & m_i \cdot c_i & 0 & \tilde{b}_i \end{bmatrix}$$

Thus,  $\hat{\Sigma}'$  is a block-diagonal matrix, where the *i*-th block is of the form

$$\begin{bmatrix} \hat{a}_i & 0 & -- & -- \\ 0 & \hat{a}_i & -- & -- \\ -- & -- & \hat{b}_i & 0 \\ -- & -- & 0 & \hat{b}_i \end{bmatrix}$$

Note that for indeces i corresponding to s (resp. e) variables,  $\mathbf{v}=\mathbf{e}_i\mathbf{R}$  is such that  $\mathbf{v}=\mathbf{v}_0,\ldots,\mathbf{v}_{2n}$  has 0's in positions j such that  $j=2 \mod 4$  or  $j=3 \mod 4$  (resp.  $j=0 \mod 4$  or  $j=1 \mod 4$ ). Furthermore, for  $\ell \in [n/2], \ v_{4(\ell-1)+1}^2 + v_{4(\ell-1)+2}^2 = \frac{2}{n}$  (resp.  $v_{4(\ell-1)+2}^2 + v_{4(\ell-1)+3}^2 = \frac{2}{n}$ ).

Assuming the first n indeces correspond to the s variables, and the last n indeces correspond to the e

Assuming the first n indeces correspond to the s variables, and the last n indeces correspond to the e variables, we have that for  $i \in [n]$ ,  $\mathbf{e}_i \boldsymbol{\Sigma}' \mathbf{e}_i^T + \mathbf{e}_{i+n} \boldsymbol{\Sigma}' \mathbf{e}_{i+n}^T = \frac{\mathbf{Tr}}{n}$ , where  $\mathbf{Tr}$  is the trace of  $\hat{\boldsymbol{\Sigma}}'$ , which is the same as the trace of  $\boldsymbol{\Sigma}'$ . Thus, for  $i \in [n]$ ,  $\min(\mathbf{e}_i \boldsymbol{\Sigma}' \mathbf{e}_i^T, \mathbf{e}_{i+n} \boldsymbol{\Sigma}' \mathbf{e}_{i+n}^T) \leq \frac{\mathbf{Tr}(\boldsymbol{\Sigma}')}{2n}$ .

## **B** Additional Experimental Results

Parameter	Original	Noise	Num	Final
Set	Security	Variance	Queries $(t)$	Security
		$ ho_{fresh}^2$	1000	$247 \text{ bikz} \approx 65 \text{ bits}$
$\log_2 n = 10, \log_2 q_L = 25$	128/102	$100 \cdot \rho_{fresh}^2$	1000	336 bikz $\approx 89$ bits
-2	,	$-\frac{t \cdot \rho_{\text{fresh}}^2}{2}$	1000	$374 \text{ bikz} \approx 99 \text{ bits}$
		$\rho_{fresh}^2$	$-\frac{1}{1000}$	$\overline{366}$ bikz $\approx 97$ bits
$\log_2 n = 10, \log_2 q_L = 17$	192/170	$100 \cdot \rho_{fresh}^2$	1000	536 bikz $\approx$ 142 bits
		$t \cdot  ho_{fresh}^2$	1000	615 bikz $\approx$ 163 bits
		$  ho_{fresh}^2$ $  -$	1000	$4\overline{61}$ bikz $\approx 12\overline{2}$ bits
$\log_2 n = 10, \log_2 q_L = 13$	256/234	$100 \cdot \rho_{fresh}^2$	1000	714 bikz $\approx$ 189 bits
		$t \cdot  ho_{fresh}^2$	1000	840 bikz $\approx 223$ bits
		$ ho_{fresh}^2$	1000	$288 \text{ bikz} \approx 76 \text{ bits}$
$\log_2 n = 11, \log_2 q_L = 51$	128/97	$100 \cdot \rho_{fresh}^2$	1000	$340 \text{ bikz} \approx 90 \text{ bits}$
		$-\frac{t}{\rho_{fresh}^2}^2$	1000	$359 \text{ bikz} \approx 95 \text{ bits}$
		$ ho_{fresh}^2$	1000	$450 \text{ bikz} \approx 119 \text{ bits}$
$\log_2 n = 11, \log_2 q_L = 35$	192/162	$100 \cdot \rho_{fresh}^2$	1000	557 bikz $\approx$ 148 bits
		$-rac{t}{ ho_{fresh}^2}$ -	1000	$599 \text{ bikz} \approx 159 \text{ bits}$
		$ ho_{fresh}^2$	1000	$590 \text{ bikz} \approx 157 \text{ bits}$
$\log_2 n = 11, \log_2 q_L = 27$	256/226	$100 \cdot \rho_{fresh}^2$	1000	761 bikz $\approx 201$ bits
		$t \cdot  ho_{fresh}^2$	1000	831 bikz $\approx 220$ bits
		$ ho_{fresh}^2$	1000	$322 \text{ bikz} \approx 85 \text{ bits}$
$\log_2 n = 12, \log_2 q_L = 101$	128/97	$100 \cdot \rho_{fresh}^2$	1000	$352 \text{ bikz} \approx 93 \text{ bits}$
		$t \cdot  ho_{fresh}^2$	1000	$362 \text{ bikz} \approx 96 \text{ bits}$
		$  ho_{fresh}^2$ $  -$	1000	$517 \text{ bikz} \approx 137 \text{ bits}$
$\log_2 n = 12, \log_2 q_L = 70$	192/161	$100 \cdot \rho_{fresh}^2$	1000	580 bikz $\approx$ 154 bits
		$t \cdot \rho_{\text{fresh}}^2$	1000	602 bikz $\approx$ 160 bits
		$ ho_{fresh}^2$	1000	$701 \text{ bikz} \approx 186 \text{ bits}$
$\log_2 n = 12, \log_2 q_L = 54$	256/227	$100 \cdot \rho_{fresh}^2$	1000	807 bikz $\approx 214$ bits
		$t \cdot  ho_{fresh}^2$	1000	845 bikz $\approx 224$ bits

Fig. 4: Concrete security of lattice reduction attacks after observing decryptions of fresh ciphertexts. For each parameter set, the second column provides the target security as well as the number of bits of security computed by the tool of [16]. The third column indicates the noise-flooding noise added before returning the decryption.  $\rho_{\text{fresh}}^2$  is the variance of the noise that is already present in a fresh ciphertext (see Section 4.5.1). The fourth column indicates the number of decryptions observed by the adversary. The final column provides the reduced security level after the attack in terms of bitz (see [16]) and bit-security.

Parameter	Original	Noise	Num	Final
Set	Security	Variance	Queries $(t)$	Security
$\log_2 n = 13, \log_2 q_L = 202$	128/96	$\begin{array}{c} \rho_{fresh}^2 \\ 100 \cdot \rho_{fresh}^2 \\ t \cdot \rho_{fresh}^2 \end{array}$	1000 1000 1000	$340 \text{ bikz} \approx 90 \text{ bits}$ $356 \text{ bikz} \approx 94 \text{ bits}$ $361 \text{ bikz} \approx 96 \text{ bits}$
$\log_2 n = 13, \log_2 q_L = 141$	192/159	$-rac{t\cdot ho_{ ext{fresh}}^2}{ ho_{ ext{fresh}}^2} - rac{t\cdot ho_{ ext{fresh}}^2}{ ho_{ ext{fresh}}^2} - rac{t\cdot ho_{ ext{fresh}}^2}{ ho_{ ext{fresh}}^2}$	1000 1000 1000	$553 \text{ bikz} \approx 146 \text{ bits}$ $587 \text{ bikz} \approx 156 \text{ bits}$ $598 \text{ bikz} \approx 159 \text{ bits}$
$\log_2 n = 13, \log_2 q_L = 109$	256/225	$-rac{t\cdot ho_{ ext{fresh}}^2}{ ho_{ ext{fresh}}^2} - rac{1}{ ho_{ ext{fresh}}^2} - rac{1}{$	1000 1000 1000	765 bikz $\approx 203$ bits 823 bikz $\approx 218$ bits 843 bikz $\approx 223$ bits
$\log_2 n = 14, \log_2 q_L = 411$	128/93	$\begin{array}{c} \rho_{fresh}^2 \\ 100 \cdot \rho_{fresh}^2 \\ t \cdot \rho_{fresh}^2 \end{array}$	1000 1000 1000	$341 \text{ bikz} \approx 90 \text{ bits}$ $349 \text{ bikz} \approx 93 \text{ bits}$ $352 \text{ bikz} \approx 93 \text{ bits}$
$\log_2 n = 14, \log_2 q_L = 284$	192/158	$ ho_{ extsf{fresh}}^2 \ 100 \cdot  ho_{ extsf{fresh}}^2 \ t \cdot  ho_{ extsf{fresh}}^2$	1000 1000 1000	$570 \text{ bikz} \approx 151 \text{ bits}$ $587 \text{ bikz} \approx 156 \text{ bits}$ $593 \text{ bikz} \approx 157 \text{ bits}$
$\log_2 n = 14, \log_2 q_L = 220$	256/222	$ ho_{ ext{fresh}}^2 \ t \cdot  ho_{ ext{fresh}}^2 \ t \cdot  ho_{ ext{fresh}}^2$	1000 1000 1000	796 bikz $\approx 211$ bits 826 bikz $\approx 219$ bits 836 bikz $\approx 222$ bits
$\log_2 n = 15, \log_2 q_L = 827$	128/92	$\begin{array}{c} \rho_{fresh}^2 \\ 100 \cdot \rho_{fresh}^2 \\ t \cdot \rho_{fresh}^2 \end{array}$	1000 1000 1000	$343 \text{ bikz} \approx 91 \text{ bits}$ $347 \text{ bikz} \approx 92 \text{ bits}$ $348 \text{ bikz} \approx 92 \text{ bits}$
$\log_2 n = 15, \log_2 q_L = 571$	192/156	$\begin{array}{c} \rho_{fresh}^2 \\ 100 \cdot \rho_{fresh}^2 \\ \underline{t \cdot \rho_{fresh}^2} \end{array}$	1000 1000 1000	$577 \text{ bikz} \approx 153 \text{ bits}$ $586 \text{ bikz} \approx 155 \text{ bits}$ $589 \text{ bikz} \approx 156 \text{ bits}$
$\log_2 n = 15, \log_2 q_L = 443$	256/220	$\begin{array}{c} \rho_{fresh}^2 \\ 100 \cdot \rho_{fresh}^2 \\ t \cdot \rho_{fresh}^2 \end{array}$	1000 1000 1000	810 bikz $\approx 215$ bits 826 bikz $\approx 219$ bits 831 bikz $\approx 220$ bits
$\log_2 n = 17, \log_2 q_L = 2400$	140/146	$\begin{array}{c} \rho_{fresh}^2 \\ 100 \cdot \rho_{fresh}^2 \\ t \cdot \rho_{fresh}^2 \end{array}$	1000 1000 1000	$548 \text{ bikz} \approx 145 \text{ bits}$ $550 \text{ bikz} \approx 146 \text{ bits}$ $551 \text{ bikz} \approx 146 \text{ bits}$
$\log_2 n = 17, \log_2 q_L = 2000$	193/187	$ ho_{ ext{fresh}}^2 \ 100 \cdot  ho_{ ext{fresh}}^2 \ t \cdot  ho_{ ext{fresh}}^2$	1000 1000 1000	703 bikz $\approx 186$ bits 706 bikz $\approx 187$ bits 707 bikz $\approx 187$ bits

Fig. 4: Concrete security of lattice reduction attacks after observing decryptions of fresh ciphertexts, cont'd. For each parameter set, the second column provides the target security as well as the number of bits of security computed by the tool of [16]. The third column indicates the noise-flooding noise added before returning the decryption.  $\rho_{\text{fresh}}^2$  is the variance of the noise that is already present in a fresh ciphertext (see Section 4.5.1). The fourth column indicates the number of decryptions observed by the adversary. The final column provides the reduced security level after the attack in terms of bikz (see [16]) and bit-security.

Parameter Set	Orig Security	Noise Var	Num Queries	Succ Prob
$\log_2 n = 10, \log_2 q_L = 25$	128/102	$\rho_{fresh}^2$	1160	0.81
$\log_2 n = 10, \log_2 q_L = 17$	192/170	$\frac{100 \cdot \rho_{fresh}^2}{\rho_{fresh}^2}$	1160	$\frac{0.80}{0.81}$
$\log_2 n = 10, \log_2 q_L = 13$	256/234	$100 \cdot \rho_{\text{fresh}}^2$	$-\frac{62,180}{1160}$	$-\frac{0.80}{0.81}$
$\frac{\log_2 n - 10, \log_2 q_L - 15}{n}$	200/204	$100 \cdot \rho_{fresh}^2$	62,180	0.80
$\log_2 n = 11, \log_2 q_L = 51$	128/97	$\begin{array}{c} \rho_{fresh}^2 \\ 100 \cdot \rho_{fresh}^2 \end{array}$		0.80 $0.80$
$\log_2 n = 11, \log_2 q_L = 35$	192/162	$ ho_{fresh}^2$ $100 \cdot \rho_{fresh}^2$	1220 $67,950$	$0.80 \\ 0.80$
$\log_2 n = 11, \log_2 q_L = 27$	256/226	$ ho_{fresh}^2$ $100 \cdot  ho_{fresh}^2$	1220 $67,950$	0.80 $0.80$
$\log_2 n = 12, \log_2 q_L = 101$	128/97	$ ho_{fresh}^2$	1290	0.81
$\log_2 n = 12, \log_2 q_L = 70$	192/161	$100 \cdot \rho_{\text{fresh}}^2$	$-\frac{73,760}{1290}$	$\frac{0.80}{0.81}$
		$100 \cdot \rho_{fresh}^2$	$-\frac{73,760}{1290}$	$\frac{0.80}{0.81}$
$\log_2 n = 12, \log_2 q_L = 54$	256/227	$100 \cdot \rho_{fresh}^2$	73,760	0.80

Fig. 5: Concrete security of guessing attacks after observing decryptions of fresh ciphertexts. For each parameter set, the second column provides the target security and the number of bits of security computed by the tool of [16]. The third column indicates the noise-flooding variance added before returning the decryption.  $\rho_{\text{fresh}}^2$  is the noise variance already present in a fresh ciphertext. The fourth column indicates the number of decryptions observed by the adversary. The final column indicates the success probability of the attack, which corresponds to the probability that all guesses are correct, conditioned on the event in Lemma 7.1 occurring.

			~
0		Num	Succ
Security	Var	Queries	Prob
100/00	$\rho_{fresh}^2$	1350	0.80
128/90		79,600	0.80
102/150	$ \rho_{fresh}^2$	$1\bar{3}5\bar{0}$	0.80
	$100 \cdot \rho_{\text{fresh}}^2$	79,600	0.80
256/225	$  ho_{fresh}^2$	$1\bar{3}\bar{5}\bar{0}$	0.80
200/220	$100 \cdot \rho_{fresh}^2$	79,600	0.80
199/02	$\rho_{fresh}^2$	1420	0.81
128/93		85,450	0.80
109/159	$   ho_{fresh}^2$ $  -$	1420	0.81
192/136	$100 \cdot \rho_{fresh}^2$	85,450	0.80
256/222	$ \rho_{fresh}^2$	1420	0.81
230/222	$100 \cdot \rho_{fresh}^2$	85,450	0.80
199/09	$ ho_{fresh}^2$	1480	0.80
126/92	$100 \cdot \rho_{fresh}^2$	$91,\!320$	0.80
109/156	$ \rho_{fresh}^2$	1480	0.80
192/130	$100 \cdot \rho_{fresh}^2$	$91,\!320$	0.80
256/220	$ ho_{fresh}^2$	1480	0.80
230/220	$100 \cdot \rho_{fresh}^2$	$91,\!320$	0.80
	$\rho_{fresh}^2$	1690	0.81
140/146		103,360	0.80
	$  ho_{fresh}^2$	1690	0.81
193/187	$100 \cdot \rho_{fresh}^2$	103,360	0.82
	Orig Security  128/96 192/159 256/225  128/93 192/158 192/156 256/222  140/146 193/187	$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	$\begin{array}{c ccccccccccccccccccccccccccccccccccc$

Fig. 5: Concrete security of guessing attacks after observing decryptions of fresh ciphertexts, cont'd. For each parameter set, the second column provides the target security and the number of bits of security computed by the tool of [16]. The third column indicates the noise-flooding variance added before returning the decryption.  $\rho_{\text{fresh}}^2$  is the noise variance already present in a fresh ciphertext. The fourth column indicates the number of decryptions observed by the adversary. The final column indicates the success probability of the attack, which corresponds to the probability that all guesses are correct, conditioned on the event in Lemma 7.1 occurring.

Parameter	Orig	Noise	Num	Num	Succ	Final
Set	Security	Var	Queries	Guess	Prob	Security
$\log_2 n = 10, \log_2 q_L = 25$	128/102	$ ho_{fresh}^2$	1130	793	0.80	$62 \text{ bikz} \approx 17 \text{ bits}$
$\log_2 n = 10, \log_2 q_L = 20$	120/102	$100 \cdot \rho_{fresh}^2$	58,000	621	0.80	48 bikz $\approx 13$ bits
$\log_2 n = 10, \log_2 q_L = 17$	192/170	$-\rho_{fresh}^{\overline{2}}$	1130	793	0.80	$109 \text{ bikz} \approx 29 \text{ bits}$
$\log_2 n = 10, \log_2 q_L = 11$	192/110	$100 \cdot \rho_{fresh}^2$	58,000	621	0.80	$160 \text{ bikz} \approx 42 \text{ bits}$
$\log_2 n = 10, \log_2 q_L = 13$	256/234	$-\rho_{fresh}^2$	1130	793	0.80	$160 \text{ bikz} \approx 42 \text{ bits}$
$\log_2 n = 10, \log_2 q_L = 13$	200/204	$100 \cdot \rho_{fresh}^2$	58,000	621	0.80	222 bikz $\approx 59$ bits
l	100/07	$ ho_{fresh}^2$	1130	801	0.80	$154 \text{ bikz} \approx 41 \text{ bits}$
$\log_2 n = 11, \log_2 q_L = 51$	128/97	$100 \cdot \rho_{fresh}^2$	58,000	623	0.80	$184 \text{ bikz} \approx 49 \text{ bits}$
$\log_2 n = 11, \log_2 q_L = 35$	192/162	$-\rho_{fresh}^2$	1130	801	0.80	$\overline{261}$ bikz $\approx 69$ bits
$\log_2 n = 11, \log_2 q_L = 33$	192/102	$100 \cdot \rho_{fresh}^2$	58,000	623	0.80	$306 \text{ bikz} \approx 81 \text{ bits}$
$\log_2 n = 11, \log_2 q_L = 27$	256/226	$-\rho_{fresh}^2$	1130	801	0.80	$\overline{358}$ bikz $\approx 95$ bits
$\log_2 n = 11, \log_2 qL = 21$	230/220	$100 \cdot  ho_{fresh}^2$	58,000	623	0.80	415 bikz $\approx 110$ bits
l 10 l 101	100/07	$ ho_{fresh}^2$	1130	807	0.80	$240 \text{ bikz} \approx 63 \text{ bits}$
$\log_2 n = 12, \log_2 q_L = 101$	128/97	$100 \cdot \rho_{fresh}^2$	58,000	624	0.80	260 bikz $\approx$ 69 bits
lam m 19 lam m 70	109/161	$-\rho_{fresh}^2$	1130	807	0.80	$395 \text{ bikz} \approx 105 \text{ bits}$
$\log_2 n = 12, \log_2 q_L = 70$	192/161	$100 \cdot \rho_{fresh}^2$	58,000	624	0.80	427 bikz $\approx$ 113 bits
log m = 12 log g = 54	256/227	$-\rho_{fresh}^2$	1130	807	0.80	$544 \text{ bikz} \approx 144 \text{ bits}$
$\log_2 n = 12, \log_2 q_L = 54$	256/227	$100 \cdot  ho_{\mathrm{fresh}}^2$	58,000	624	0.80	587 bikz $\approx 156$ bits

Fig. 6: Concrete security of hybrid attacks after observing decryptions of fresh ciphertexts. For each parameter set, the second column provides the target security and the number of bits of security computed by the tool of [16]. The third column indicates the noise-flooding variance added before returning the decryption.  $\rho_{\text{fresh}}^2$  is the noise variance that is already present in a fresh ciphertext. The fourth column indicates the number of decryptions observed by the adversary. The fifth column indicates the number of coordinates of the LWE secret/error that are guessed by the adversary. The sixth column indicates the success probability of the attack, which corresponds to the probability that all guesses are correct, conditioned on the events in Lemma 7.1 and (15) occurring. The final column provides the reduced security level after the attack in terms of bikz (see [16]) and bit-security.

Parameter Set	Orig Security	Noise Var	Num Queries	Num Guess		Final Security
$\log_2 n = 13, \log_2 q_L = 202$	128/96	$\rho_{fresh}^2 \\ 100 \cdot \rho_{fresh}^2$	1130 58,000	810 625	0.80 0.80	$294 \text{ bikz} \approx 78 \text{ bits}$ $306 \text{ bikz} \approx 81 \text{ bits}$
$\log_2 n = 13, \log_2 q_L = 141$	192/159	$\begin{array}{c} -\frac{1}{\rho_{fresh}^2} \\ 100 \cdot \rho_{fresh}^2 \end{array}$	1130 58,000	$\frac{1}{810}$ $\frac{1}{625}$		$4\overline{82}$ bikz $\approx 12\overline{8}$ bits $502$ bikz $\approx 133$ bits
$\log_2 n = 13, \log_2 q_L = 109$	256/225	$\begin{array}{c} -\frac{1}{\rho_{fresh}^2} \\ 100 \cdot \rho_{fresh}^2 \end{array}$	1130 58,000	$     \begin{array}{r}                                     $		$671 \text{ bikz} \approx 178 \text{ bits}$ $699 \text{ bikz} \approx 185 \text{ bits}$
$\log_2 n = 14, \log_2 q_L = 411$	128/93	$\rho_{fresh}^2$ $100 \cdot \rho_{fresh}^2$	1130 58,000	813 625	0.80	$317 \text{ bikz} \approx 84 \text{ bits}$ $324 \text{ bikz} \approx 86 \text{ bits}$
$\log_2 n = 14, \log_2 q_L = 184$	192/158	$-\frac{7}{\rho_{fresh}^2}$ $100 \cdot \rho_{fresh}^2$	1130 58,000	$\frac{1}{813}$		$5\overline{32}$ bikz $\approx 1\overline{41}$ bits $543$ bikz $\approx 144$ bits
$\log_2 n = 14, \log_2 q_L = 220$	256/222	$\rho_{\text{fresh}}^2 = 100 \cdot \rho_{\text{fresh}}^2$	1130 58,000	$\overline{813}$ $\overline{625}$		$745 \text{ bikz} \approx 197 \text{ bits}$ $760 \text{ bikz} \approx 202 \text{ bits}$
$\log_2 n = 15, \log_2 q_L = 827$	128/92	$\rho_{fresh}^2 \\ 100 \cdot \rho_{fresh}^2$	1130 58,000	814 626	0.80 0.80	$331 \text{ bikz} \approx 88 \text{ bits}$ $334 \text{ bikz} \approx 89 \text{ bits}$
$\log_2 n = 15, \log_2 q_L = 571$	192/156	$\frac{-\frac{7}{\rho_{fresh}^2}}{100 \cdot \rho_{fresh}^2}$	1130	814 626		$558 \text{ bikz} \approx 148 \text{ bits}$ $564 \text{ bikz} \approx 149 \text{ bits}$
$\log_2 n = 15, \log_2 q_L = 443$	256/220	$\begin{array}{c} - \rho_{\text{fresh}}^2 \\ 100 \cdot \rho_{\text{fresh}}^2 \end{array}$	1130 58,000	814 626		$783 \text{ bikz} \approx 208 \text{ bits}$ $792 \text{ bikz} \approx 210 \text{ bits}$
$\log_2 n = 17, \log_2 q_L = 2400$	140/146	$\begin{array}{c} \rho_{fresh}^2 \\ 100 \cdot \rho_{fresh}^2 \end{array}$		1557 1411	0.80	$539 \text{ bikz} \approx 143 \text{ bits}$ $540 \text{ bikz} \approx 143 \text{ bits}$
$\log_2 n = 17, \log_2 q_L = 2000$	193/187	$\frac{\rho_{fresh}^2}{100 \cdot \rho_{fresh}^2}$	1260 65,000	1557 1411		$692 \text{ bikz} \approx 183 \text{ bits}$ $691 \text{ bikz} \approx 183 \text{ bits}$

Fig. 6: Concrete security of hybrid attacks after observing decryptions of fresh ciphertexts, cont'd. For each parameter set, the second column provides the target security and the number of bits of security computed by the tool of [16]. The third column indicates the noise-flooding variance added before returning the decryption.  $\rho_{\text{fresh}}^2$  is the noise variance that is already present in a fresh ciphertext. The fourth column indicates the number of decryptions observed by the adversary. The fifth column indicates the number of coordinates of the LWE secret/error that are guessed by the adversary. The sixth column indicates the success probability of the attack, which corresponds to the probability that all guesses are correct, conditioned on the events in Lemma 7.1 and (15) occurring. The final column provides the reduced security level after the attack in terms of bikz (see [16]) and bit-security.

Parameter	Original	Noise	Num	Final
Set	Security	Variance	Queries $(t)$	Security
		$ ho_{C1}^2$	1000	$298 \text{ bikz} \approx 79 \text{ bits}$
$\log_2 n = 10, \log_2 q_L = 25$	128/102	$100 \cdot \rho_{C1}^2$	1000	$352 \text{ bikz} \approx 93 \text{ bits}$
		$t \cdot  ho_{C1}^2$	1000	376 bikz $\approx$ 100 bits
		$-\frac{1}{\rho_{C1}^2}$	1000	$460 \text{ bikz} \approx 122 \text{ bits}$
$\log_2 n = 10, \log_2 q_L = 17$	192/170	$100 \cdot \rho_{C1}^2$	1000	568 bikz $\approx$ 150 bits
		$t \cdot \rho_{\text{C1}}^2$	1000	619 bikz $\approx$ 164 bits
		$-\rho_{C1}^2$	1000	$598 \text{ bikz} \approx 158 \text{ bits}$
$\log_2 n = 10, \log_2 q_L = 13$	256/234	$100 \cdot \rho_{C1}^2$	1000	764 bikz $\approx$ 202 bits
		$t \cdot  ho_{C1}^2$	1000	847 bikz $\approx$ 224 bits
		$\rho_{C1}^2$	1000	$319 \text{ bikz} \approx 85 \text{ bits}$
$\log_2 n = 11, \log_2 q_L = 51$	128/97	$100 \cdot \rho_{C1}^2$	1000	$348 \text{ bikz} \approx 92 \text{ bits}$
		$t \cdot  ho_{C1}^2$	1000	$360 \text{ bikz} \approx 95 \text{ bits}$
		$-\rho_{C1}^{2}$	1000	$5\overline{13}$ bikz $\approx 1\overline{36}$ bits
$\log_2 n = 11, \log_2 q_L = 35$	192/162	$100 \cdot \rho_{C1}^2$	1000	575 bikz $\approx 152$ bits
		$-\frac{t\cdot\rho_{C\underline{1}}^2}{2}$	1000	601 bikz $\approx$ 159 bits
		$-\rho_{\text{C1}}^2$	1000	$689 \text{ bikz} \approx 183 \text{ bits}$
$\log_2 n = 11, \log_2 q_L = 27$	256/226	$100 \cdot \rho_{C1}^2$	1000	790 bikz $\approx$ 209 bits
		$t \cdot  ho_{C1}^2$	1000	834 bikz $\approx 221$ bits
		$ ho_{C1}^2$	1000	$341 \text{ bikz} \approx 90 \text{ bits}$
$\log_2 n = 12, \log_2 q_L = 101$	128/97	$100 \cdot \rho_{C1}^2$	1000	$356 \text{ bikz} \approx 94 \text{ bits}$
		$t \cdot  ho_{C1}^2$	1000	$363 \text{ bikz} \approx 96 \text{ bits}$
		$ ho_{C1}^2$	1000	$555 \text{ bikz} \approx 147 \text{ bits}$
$\log_2 n = 12, \log_2 q_L = 70$	192/161	$100 \cdot \rho_{C1}^2$	1000	589 bikz $\approx$ 156 bits
		$t \cdot  ho_{C1}^2$	1000	603 bikz $\approx$ 160 bits
		$-\rho_{C1}^2$	1000	$764 \text{ bikz} \approx 203 \text{ bits}$
$\log_2 n = 12, \log_2 q_L = 54$	256/227	$100 \cdot \rho_{C1}^2$	1000	823 bikz $\approx$ 218 bits
		$t \cdot  ho_{C1}^2$	1000	$847 \text{ bikz} \approx 224 \text{ bits}$

Fig. 7: Concrete security of lattice reduction attacks after observing decryptions of Class 1 ciphertexts. For each parameter set, the second column provides the target security as well as the number of bits of security computed by the tool of [16]. The third column indicates the noise-flooding noise added before returning the decryption to the adversary.  $\rho_{C1}^2$  is the variance of the noise that is already present in a ciphertext obtained from evaluating a Class 1 circuit on fresh encryptions. The final column provides the reduced security level after the attack in terms of bikz (see [16]) and bit-security.

Parameter	Original	Noise	Num	Final
Set	Security	Variance	Queries $(t)$	Security
$\log_2 n = 13, \log_2 q_L = 202$	128/96	$\begin{array}{c} \rho_{\text{C1}}^2 \\ 100 \cdot \rho_{\text{C1}}^2 \\ t \cdot \rho_{\text{C1}}^2 \end{array}$	1000 1000 1000	$350 \text{ bikz} \approx 93 \text{ bits}$ $358 \text{ bikz} \approx 95 \text{ bits}$ $361 \text{ bikz} \approx 96 \text{ bits}$
$\log_2 n = 13, \log_2 q_L = 141$	192/159	$ \begin{array}{c} -\rho_{\text{C1}}^2 \\ 100 \cdot \rho_{\text{C1}}^2 \\ \underline{t \cdot \rho_{\text{C1}}^2} \end{array} $	1000 1000 1000	$574 \text{ bikz} \approx 152 \text{ bits}$ $592 \text{ bikz} \approx 157 \text{ bits}$ $599 \text{ bikz} \approx 159 \text{ bits}$
$\log_2 n = 13, \log_2 q_L = 109$	256/225	$\rho_{C1}^2$ $100 \cdot \rho_{C1}^2$ $t \cdot \rho_{C1}^2$	1000 1000 1000	800 bikz $\approx 212$ bits 831 bikz $\approx 220$ bits 844 bikz $\approx 224$ bits
$\log_2 n = 14, \log_2 q_L = 411$	128/93	$\rho_{C1}^2 \\ 100 \cdot \rho_{C1}^2 \\ t \cdot \rho_{C1}^2$	1000 1000 1000	$346 \text{ bikz} \approx 92 \text{ bits}$ $350 \text{ bikz} \approx 93 \text{ bits}$ $352 \text{ bikz} \approx 93 \text{ bits}$
$\log_2 n = 14, \log_2 q_L = 284$	192/158	$ \begin{array}{c} \rho_{\text{C1}}^{2} \\ 100 \cdot \rho_{\text{C1}}^{2} \\ \underline{t \cdot \rho_{\text{C1}}^{2}} \end{array} $	1000 1000 1000	$581 \text{ bikz} \approx 154 \text{ bits}$ $590 \text{ bikz} \approx 156 \text{ bits}$ $593 \text{ bikz} \approx 157 \text{ bits}$
$\log_2 n = 14, \log_2 q_L = 220$	256/222	$ \begin{array}{c} \rho_{C1}^2 \\ 100 \cdot \rho_{C1}^2 \\ t \cdot \rho_{C1}^2 \end{array} $	1000 1000 1000	815 bikz $\approx 216$ bits 831 bikz $\approx 220$ bits 837 bikz $\approx 222$ bits
$\log_2 n = 15, \log_2 q_L = 827$	128/92	$ \begin{array}{c} \rho_{\text{C1}}^{2} \\ 100 \cdot \rho_{\text{C1}}^{2} \\ \underline{t \cdot \rho_{\text{C1}}^{2}} \end{array} $	1000 1000 1000	$345 \text{ bikz} \approx 92 \text{ bits}$ $347 \text{ bikz} \approx 92 \text{ bits}$ $348 \text{ bikz} \approx 92 \text{ bits}$
$\log_2 n = 15, \log_2 q_L = 571$	192/156	$ \begin{array}{c} \rho_{C1}^2 \\ 100 \cdot \rho_{C1}^2 \\ \underline{t \cdot \rho_{C1}^2} \end{array} $	1000 1000 1000	$583 \text{ bikz} \approx 154 \text{ bits}$ $588 \text{ bikz} \approx 156 \text{ bits}$ $589 \text{ bikz} \approx 156 \text{ bits}$
$\log_2 n = 15, \log_2 q_L = 443$	256/220	$ \begin{array}{c} \rho_{C1}^2 \\ 100 \cdot \rho_{C1}^2 \\ t \cdot \rho_{C1}^2 \end{array} $	1000 1000 1000	820 bikz $\approx 217$ bits 828 bikz $\approx 219$ bits 831 bikz $\approx 220$ bits
$\log_2 n = 17, \log_2 q_L = 2400$	140/146	$\begin{array}{c} \rho_{\text{C1}}^2 \\ 100 \cdot \rho_{\text{C1}}^2 \\ \underline{t \cdot \rho_{\text{C1}}^2} \end{array}$	1000 1000 1000	$549 \text{ bikz} \approx 145 \text{ bits}$ $550 \text{ bikz} \approx 146 \text{ bits}$ $551 \text{ bikz} \approx 146 \text{ bits}$
$\log_2 n = 17, \log_2 q_L = 2000$	193/187	$\begin{array}{c} \rho_{fresh}^2 \\ 100 \cdot \rho_{C1}^2 \\ t \cdot \rho_{C1}^2 \end{array}$	1000 1000 1000	705 bikz $\approx 187$ bits 706 bikz $\approx 187$ bits 707 bikz $\approx 187$ bits

Fig. 7: Concrete security of lattice reduction attacks after observing decryptions of Class 1 ciphertexts, continued. For each parameter set, the second column provides the target security as well as the number of bits of security computed by the tool of [16]. The third column indicates the noise-flooding noise added before returning the decryption to the adversary.  $\rho_{C1}^2$  is the variance of the noise that is already present in a ciphertext obtained from evaluating a Class 1 circuit on fresh encryptions. The final column provides the reduced security level after the attack in terms of bikz (see [16]) and bit-security.

	Original	Noise	$_{ m Num}$	Final
Set	Security	Variance	Queries $(t)$	Security
		$ ho_{C2}^2$	1000	$302 \text{ bikz} \approx 80 \text{ bits}$
$\log_2 n = 10, \log_2 q_L = 25$	128/102	$100 \cdot \rho_{C2}^2$	1000	$354 \text{ bikz} \approx 94 \text{ bits}$
		$t \cdot \rho_{C2}^2$	1000	377 bikz $\approx$ 100 bits
		$-\rho_{C2}^{2}$	1000	$4\overline{67}$ bikz $\approx 12\overline{4}$ bits
$\log_2 n = 10, \log_2 q_L = 17$	192/170	$100 \cdot \rho_{C2}^2$	1000	568 bikz $\approx$ 150 bits
		$t \cdot  ho_{C2}^2$	1000	622 bikz $\approx$ 165 bits
		$-\frac{t\cdot\rho_{C2}^2}{\rho_{C2}^2}-$	1000	$609 \text{ bikz} \approx 161 \text{ bits}$
$\log_2 n = 10, \log_2 q_L = 13$	256/234	$100 \cdot \rho_{C2}^2$	1000	772 bikz $\approx$ 205 bits
		$t \cdot  ho_{C2}^2$	1000	851 bikz $\approx$ 226 bits
		$ ho_{C2}^2$	1000	$321 \text{ bikz} \approx 85 \text{ bits}$
$\log_2 n = 11, \log_2 q_L = 51$	128/97	$100 \cdot \rho_{C2}^2$	1000	$349 \text{ bikz} \approx 93 \text{ bits}$
		$t \cdot \rho_{C2}^2$	1000	$361 \text{ bikz} \approx 96 \text{ bits}$
		$-\rho_{C2}^2$	1000	$517 \text{ bikz} \approx 137 \text{ bits}$
$\log_2 n = 11, \log_2 q_L = 35$	192/162	$100 \cdot \rho_{C2}^2$	1000	577 bikz $\approx$ 153 bits
		$t \cdot \rho_{C2}^2$	1000	603 bikz $\approx$ 160 bits
		$-\rho_{C2}^2$	1000	$695 \text{ bikz} \approx 184 \text{ bits}$
$\log_2 n = 11, \log_2 q_L = 27$	256/226	$100 \cdot \rho_{C2}^2$	1000	794 bikz $\approx$ 210 bits
		$t \cdot  ho_{C2}^2$	1000	836 bikz $\approx$ 222 bits
		$ ho_{C2}^2$	1000	$342 \text{ bikz} \approx 91 \text{ bits}$
$\log_2 n = 12, \log_2 q_L = 101$	128/97	$100 \cdot \rho_{C2}^2$	1000	$357 \text{ bikz} \approx 95 \text{ bits}$
		$-\frac{t}{\rho_{C2}^2} - \frac{p_{C2}^2}{\rho_{C2}^2} -$	1000	$363 \text{ bikz} \approx 96 \text{ bits}$
		$-\rho_{C2}^2$	1000	$557 \text{ bikz} \approx 148 \text{ bits}$
$\log_2 n = 12, \log_2 q_L = 70$	192/161	$100 \cdot \rho_{C2}^2$	1000	591 bikz $\approx$ 157 bits
		$-\frac{t \cdot \rho_{C2}^2}{\rho_{C2}^2} -$	1000	$604 \text{ bikz} \approx 160 \text{ bits}$
		$-\frac{1}{\rho_{C2}^2}$	1000	$769 \text{ bikz} \approx 204 \text{ bits}$
$\log_2 n = 12, \log_2 q_L = 54$	256/227	$100 \cdot \rho_{C2}^2$	1000	848 bikz $\approx$ 225 bits
		$t \cdot  ho_{C2}^2$	1000	825 bikz $\approx$ 219 bits

Fig. 8: Concrete security of lattice reduction attacks after observing decryptions of Class 2 ciphertexts. For each parameter set, the second column provides the target security as well as the number of bits of security computed by the tool of [16]. The third column indicates the noise-flooding noise added before returning the decryption to the adversary.  $\rho_{\text{C2}}^2$  is the variance of the noise that is already present in a ciphertext obtained from evaluating a Class 2 circuit on fresh encryptions. The fourth column indicates the number of decryptions observed by the adversary. The final column provides the reduced security level after the attack in terms of bikz (see [16]) and bit-security. The (encoded) message magnitude is equal to  $n \cdot \sqrt{\ell/3}$  in all rows, where  $\ell$  is set to 20.

Parameter	Original		Num	Final
Set	Security	variance	Queries $(t)$	Security
		$ ho_{C2}^2$	1000	$351 \text{ bikz} \approx 93 \text{ bits}$
$\log_2 n = 13, \log_2 q_L = 202$	128/96	$100 \cdot \rho_{C2}^2$	1000	$359 \text{ bikz} \approx 95 \text{ bits}$
		$t \cdot  ho_{C2}^2$	1000	$362 \text{ bikz} \approx 96 \text{ bits}$
		$-\rho_{C2}^2$	1000	$575 \text{ bikz} \approx 152 \text{ bits}$
$\log_2 n = 13, \log_2 q_L = 141$	192/159	$100 \cdot \rho_{C2}^2$	1000	$592 \text{ bikz} \approx 157 \text{ bits}$
		$t \cdot \rho_{C2}^2$	1000	$599 \text{ bikz} \approx 159 \text{ bits}$
		$-\rho_{C2}^2$	1000	$802 \text{ bikz} \approx 213 \text{ bits}$
$\log_2 n = 13, \log_2 q_L = 109$	256/225	$100 \cdot \rho_{C2}^2$	1000	832 bikz $\approx$ 221 bits
		$t \cdot  ho_{C2}^2$	1000	844 bikz $\approx 224$ bits
		$\rho_{C2}^2$	1000	$346 \text{ bikz} \approx 92 \text{ bits}$
$\log_2 n = 14, \log_2 q_L = 411$	128/93	$100 \cdot \rho_{C2}^2$	1000	$350 \text{ bikz} \approx 93 \text{ bits}$
, 021	,	$t \cdot \rho_{C2}^2$	1000	$352 \text{ bikz} \approx 93 \text{ bits}$
		$-\frac{1}{\rho_{C2}^2}$	1000	$581 \text{ bikz} \approx 154 \text{ bits}$
$\log_2 n = 14, \log_2 q_L = 284$	192/158	$100 \cdot \rho_{C2}^2$	1000	590 bikz $\approx 156$ bits
-2	,	$t \cdot \rho_{C2}^2$	1000	$594 \text{ bikz} \approx 157 \text{ bits}$
		$-\rho_{C2}^2$	1000	$816 \text{ bikz} \approx 216 \text{ bits}$
$\log_2 n = 14, \log_2 q_L = 220$	256/222	$100 \cdot \rho_{C2}^2$	1000	831 bikz $\approx$ 220 bits
		$t \cdot  ho_{C2}^2$	1000	837 bikz $\approx$ 222 bits
		$\rho_{C2}^2$	1000	$346 \text{ bikz} \approx 92 \text{ bits}$
$\log_2 n = 15, \log_2 q_L = 827$	128/92	$100 \cdot \rho_{C2}^2$	1000	$348 \text{ bikz} \approx 92 \text{ bits}$
, 621	,	$t \cdot \rho_{C2}^2$	1000	$348 \text{ bikz} \approx 92 \text{ bits}$
		$-\frac{1}{\rho_{C2}^2}$	1000	$583 \text{ bikz} \approx 155 \text{ bits}$
$\log_2 n = 15, \log_2 q_L = 571$	192/156	$100 \cdot \rho_{C2}^2$	1000	588 bikz $\approx 156$ bits
-2	•	$t \cdot \rho_{C2}^2$	1000	589 bikz $\approx 156$ bits
		$-\frac{r}{\rho_{C2}^2}$	1000	$8\overline{20}$ bikz $\approx 2\overline{17}$ bits
$\log_2 n = 15, \log_2 q_L = 443$	256/220	$100 \cdot \rho_{C2}^2$	1000	828 bikz $\approx$ 219 bits
	•	$t \cdot \rho_{C2}^2$	1000	831 bikz $\approx$ 220 bits
		$\rho_{C2}^2$	1000	$549 \text{ bikz} \approx 145 \text{ bits}$
$\log_2 n = 17, \log_2 q_L = 2400$	140/146	$100 \cdot \rho_{C2}^2$	1000	$550 \text{ bikz} \approx 146 \text{ bits}$
/ 02 1-	,	$t \cdot \rho_{C2}^2$	1000	551 bikz $\approx 146$ bits
		$-\frac{1}{\rho_{C2}^2}$	1000	$705 \text{ bikz} \approx 187 \text{ bits}$
$\log_2 n = 17, \log_2 q_L = 2000$	193/187	$100 \cdot \rho_{C2}^2$	1000	706 bikz $\approx$ 187 bits
/ 02 1-	,	$t \cdot \rho_{C2}^2$	1000	707 bikz $\approx 187$ bits
		. 02		

Fig. 8: Concrete security of lattice reduction attacks after observing decryptions of Class 2 ciphertexts, continued. For each parameter set, the second column provides the target security as well as the number of bits of security computed by the tool of [16]. The third column indicates the noise-flooding noise added before returning the decryption to the adversary.  $\rho_{\text{C2}}^2$  is the variance of the noise that is already present in a ciphertext obtained from evaluating a Class 2 circuit on fresh encryptions. The fourth column indicates the number of decryptions observed by the adversary. The final column provides the reduced security level after the attack in terms of bikz (see [16]) and bit-security. The (encoded) message magnitude is equal to  $n \cdot \sqrt{\ell/3}$  in all rows, where  $\ell$  is set to 20.

Parameter Set	Orig Security	Noise Var	Num Queries	Succ Prob
$\log_2 n = 10, \log_2 q_L = 25$	128/102	$\rho_{C1}^2$ $100 \cdot \rho_{C1}^2$	77 3850	0.81 0.80
$\log_2 n = 10, \log_2 q_L = 17$	192/170	$\begin{array}{c} -\rho_{\text{C1}}^2 \\ 100 \cdot \rho_{\text{C1}}^2 \end{array}$	$\frac{77}{3850}$	0.81 $0.80$
$\log_2 n = 10, \log_2 q_L = 13$	256/234	$\begin{array}{c} -\rho_{C1}^2 \\ 100 \cdot \rho_{C1}^2 \end{array}$	$\frac{77}{3850}$	0.81 $0.80$
$\log_2 n = 11, \log_2 q_L = 51$	128/97	$\begin{array}{c} \rho_{\text{C1}}^2 \\ 100 \cdot \rho_{\text{C1}}^2 \end{array}$	82 4200	0.82 0.80
$\log_2 n = 11, \log_2 q_L = 35$	192/162	$\begin{array}{c} \rho_{\text{C1}}^2 \\ 100 \cdot \rho_{\text{C1}}^2 \end{array}$	82	$\begin{array}{c} \overline{0.82} \\ 0.80 \end{array}$
$\log_2 n = 11, \log_2 q_L = 27$	256/226	$\begin{array}{c} -\frac{7}{\rho_{C1}^2} \\ 100 \cdot \rho_{C1}^2 \end{array}$	82 4200	0.82 $0.80$
$\log_2 n = 12, \log_2 q_L = 101$	128/97	$\rho_{C1}^2$ $100 \cdot \rho_{C1}^2$	86 4570	0.80 0.80
$\log_2 n = 12, \log_2 q_L = 70$	192/161	$\begin{array}{c} -\rho_{\text{C1}}^2 \\ 100 \cdot \rho_{\text{C1}}^2 \end{array}$	86 4570	0.80 $0.80$
$\log_2 n = 12, \log_2 q_L = 54$	256/227	$\rho_{C1}^2 \\ 100 \cdot \rho_{C1}^2$	86 4570	0.80 0.80

Fig. 9: Concrete security of guessing attacks after observing decryptions of Class 1 ciphertexts. For each parameter set, the second column provides the target security as well as the number of bits of security computed by the tool of [16]. The third column indicates the noise-flooding noise added before returning the decryption to the adversary.  $\rho_{\text{Cl}}^2$  is the variance of the noise that is already present in a ciphertext obtained by evaluating a Class 1 circuit on fresh encryptions. The fourth column indicates the number of decryptions observed by the adversary. The final column indicates the success probability of the attack, which corresponds to the probability that all guesses are correct, conditioned on the event in (17) occurring.

Donomotor	Onia	Noise	Num	Succ
Parameter	Orig			
Set	Security	Var	Queries	Prob
1 12 1 200	100/00	$ ho_{C1}^2$	91	0.81
$\log_2 n = 13, \log_2 q_L = 202$	128/96	$100 \cdot \rho_{C1}^2$	4930	0.80
log m = 12 log g = 141	109/150	$-\frac{1}{\rho_{C1}^2}$	91	0.81
$\log_2 n = 13, \log_2 q_L = 141$	192/159	$100 \cdot \rho_{C1}^2$	4930	0.80
$\log m = 12 \log n = 100$	256/225	$-\rho_{C1}^2$	91	0.81
$\log_2 n = 13, \log_2 q_L = 109$	200/220	$100 \cdot \rho_{C1}^2$	4930	0.80
	100/00	$\rho_{C1}^2$	96	0.82
$\log_2 n = 14, \log_2 q_L = 411$	128/93	$100 \cdot \rho_{C1}^2$	5290	0.80
14 1	100/150	$-\frac{2}{\rho_{C1}^2}$	96	0.82
$\log_2 n = 14, \log_2 q_L = 284$	192/158	$100 \cdot \rho_{C1}^2$	5290	0.80
14 1	056/000	$-\frac{1}{\rho_{C1}^2}$	96	0.82
$\log_2 n = 14, \log_2 q_L = 220$	256/222	$100 \cdot \rho_{C1}^2$	5290	0.80
	100/00	$\rho_{C1}^2$	100	0.80
$\log_2 n = 15, \log_2 q_L = 827$	128/92	$100 \cdot \rho_{C1}^2$	5660	0.80
1 15 1 571	100/150	$-\frac{1}{\rho_{C1}^2}$	100	$\bar{0}.\bar{8}0$
$\log_2 n = 15, \log_2 q_L = 571$	192/156	$100 \cdot \rho_{C1}^2$	5660	0.80
lam m 15 lam m 442	256/220	$-\frac{1}{\rho_{C1}^2}$	100	0.80
$\log_2 n = 15, \log_2 q_L = 443$	256/220	$100 \cdot \rho_{C1}^2$	5660	0.80
		$\rho_{C1}^2$	115	0.80
$\log_2 n = 17, \log_2 q_L = 2400$	140/146	$100 \cdot \rho_{\text{C1}}^2$	6420	0.80
		$-\frac{7}{\rho_{C1}^2}$	115	$\bar{0}.\bar{8}0$
$\log_2 n = 17, \log_2 q_L = 2000$	193/187	$100 \cdot \rho_{C1}^2$	6420	0.80

Fig. 9: Concrete security of guessing attacks after observing decryptions of Class 1 ciphertexts, continued. For each parameter set, the second column provides the target security as well as the number of bits of security computed by the tool of [16]. The third column indicates the noise-flooding noise added before returning the decryption to the adversary.  $\rho_{C1}^2$  is the variance of the noise that is already present in a ciphertext obtained by evaluating a Class 1 circuit on fresh encryptions. The fourth column indicates the number of decryptions observed by the adversary. The final column indicates the success probability of the attack, which corresponds to the probability that all guesses are correct, conditioned on the event in (17) occurring.

Parameter	Orig	Noise	Num	Succ
Set	Security	Var	Queries	Prob
$\log m = 10 \log a_2 = 25$	198/109	$ ho_{C2}^2$	97	0.81
$\log_2 n = 10, \log_2 q_L = 25$	128/102	$100 \cdot \rho_{\text{C2}}^2$	4620	0.80
$\log_2 n = 10, \log_2 q_L = 17$	192/170	$ ho_{C2}^2$	97	0.81
$\log_2 n = 10, \log_2 q_L = 11$		$100 \cdot \rho_{\text{C2}}^2$	4620	0.80
$\log_2 n = 10, \log_2 q_L = 13$	256/234	$-\rho_{C2}^2$	97	0.81
$\log_2 n = 10, \log_2 q_L = 10$	200/201	$100 \cdot \rho_{C2}^2$	4620	0.80
lam m 11 lam m E1	100/07	$ ho_{C2}^2$	103	0.81
$\log_2 n = 11, \log_2 q_L = 51$	128/97	$100 \cdot \rho_{C2}^2$	5050	0.80
$\log_2 n = 11, \log_2 q_L = 35$	192/162	$-\rho_{C2}^2$	103	0.81
$\log_2 n = 11, \log_2 q_L = 50$		$100 \cdot \rho_{C2}^2$	5050	0.80
$\log_2 n = 11, \log_2 q_L = 27$	256/226	$- ho_{C2}^2$	103	0.81
$\log_2 n = 11, \log_2 q_L = 21$	200/220	$100 \cdot \rho_{C2}^2$	5050	0.80
1	100/07	$\rho_{C2}^2$	110	0.82
$\log_2 n = 12, \log_2 q_L = 101$	128/97	$100 \cdot \rho_{C2}^2$	5490	0.80
$\log_2 n = 12, \log_2 q_L = 70$	192/161	$-\rho_{C2}^2$	110	0.82
$\log_2 n = 12, \log_2 q_L = 10$	192/101	$100 \cdot \rho_{C2}^2$	5490	0.80
$\log n = 12 \log a_{\rm r} = 54$	256/227	$-\rho_{C2}^2$	110	0.82
$\log_2 n = 12, \log_2 q_L = 54$	200/221	$100 \cdot \rho_{C2}^2$	5490	0.80

Fig. 10: Concrete security of guessing attacks after observing decryptions of Class 2 ciphertexts. For each parameter set, the second column provides the target security as well as the number of bits of security computed by the tool of [16]. The third column indicates the noise-flooding noise added before returning the decryption to the adversary.  $\rho_{\text{C2}}^2$  is the variance of the noise that is already present in a ciphertext obtained from evaluating a Class 2 circuit on fresh encryptions. The fourth column indicates the number of decryptions observed by the adversary. The final column indicates the success probability of the attack, which corresponds to the probability that all guesses are correct, conditioned on the event in (17). The (encoded) message magnitude is equal to  $n \cdot \sqrt{\ell/3}$  in all rows, where  $\ell$  is set to 20.

Parameter Set	Orig Security		Num Queries	Succ Prob
$\log_2 n = 13, \log_2 q_L = 202$	128/96	$\begin{array}{c} \rho_{\text{C2}}^2 \\ 100 \cdot \rho_{\text{C2}}^2 \end{array}$	116 5930	0.81 0.80
$\log_2 n = 13, \log_2 q_L = 141$	192/159	$\begin{array}{c} -\rho_{C2}^2 \\ 100 \cdot \rho_{C2}^2 \end{array}$	116 5930	$0.81 \\ 0.80$
$\log_2 n = 13, \log_2 q_L = 109$	256/225	$\begin{array}{c} -\rho_{\text{C2}}^2 \\ 100 \cdot \rho_{\text{C2}}^2 \end{array}$	116 5930	$0.81 \\ 0.80$
$\log_2 n = 14, \log_2 q_L = 411$	128/93	$\rho_{C2}^2$ $100 \cdot \rho_{C2}^2$	122 6370	0.80
$\log_2 n = 14, \log_2 q_L = 284$	192/158	$\frac{-\frac{7}{\rho_{C2}^2}}{100 \cdot \rho_{C2}^2}$	122	0.80 $0.80$
$\log_2 n = 14, \log_2 q_L = 220$	256/222	$ \begin{array}{c} \rho_{\text{C2}}^2 \\ 100 \cdot \rho_{\text{C2}}^2 \end{array} $	122 6370	0.80
$\log_2 n = 15, \log_2 q_L = 827$	128/92	$\begin{array}{c} \rho_{\text{C2}}^2 \\ 100 \cdot \rho_{\text{C2}}^2 \end{array}$	129 6810	0.82 0.80
$\log_2 n = 15, \log_2 q_L = 571$	192/156	$\begin{array}{c} -\rho_{C2}^{2} \\ 100 \cdot \rho_{C2}^{2} \end{array}$	129 6810	0.82 $0.80$
$\log_2 n = 15, \log_2 q_L = 443$	256/220	$\begin{array}{c} -\rho_{C2}^2 \\ 100 \cdot \rho_{C2}^2 \end{array}$	129 6810	0.82 $0.80$
$\log_2 n = 17, \log_2 q_L = 2400$	140/146	$ \begin{array}{c} \rho_{C2}^2 \\ 100 \cdot \rho_{C2}^2 \end{array} $		0.80 0.80
$\log_2 n = 17, \log_2 q_L = 2000$	193/187	$\rho_{C2}^2 \\ 100 \cdot \rho_{C2}^2$	147 7720	0.80 0.80

Fig. 10: Concrete security of guessing attacks after observing decryptions of Class 2 ciphertexts, continued. For each parameter set, the second column provides the target security as well as the number of bits of security computed by the tool of [16]. The third column indicates the noise-flooding noise added before returning the decryption to the adversary.  $\rho_{\text{C2}}^2$  is the variance of the noise that is already present in a ciphertext obtained from evaluating a Class 2 circuit on fresh encryptions. The fourth column indicates the number of decryptions observed by the adversary. The final column indicates the success probability of the attack, which corresponds to the probability that all guesses are correct, conditioned on the event in (17). The (encoded) message magnitude is equal to  $n \cdot \sqrt{\ell/3}$  in all rows, where  $\ell$  is set to 20.

Parameter Set	Orig Security	Noise Var	Num Queries	Num Guess		Final Security
$\log_2 n = 10, \log_2 q_L = 25$	128/102	$\rho_{\text{C1}}^2$ $100 \cdot \rho_{\text{C1}}^2$	75 3500	821 531	0.80	$72 \text{ bikz} \approx 19 \text{ bits}$ $150 \text{ bikz} \approx 40 \text{ bits}$
$\log_2 n = 10, \log_2 q_L = 17$	192/170	$-\frac{1}{\rho_{\text{C1}}^2}$ $100 \cdot \rho_{\text{C1}}^2$	$^{-}75$	$-\frac{1}{821}$ 531	0.80 $0.80$	$ \begin{array}{cccccccccccccccccccccccccccccccccccc$
$\log_2 n = 10, \log_2 q_L = 13$	256/234	$\begin{array}{c} -\frac{1}{\rho_{C1}^2} \\ 100 \cdot \rho_{C1}^2 \end{array}$	$-\frac{75}{3500}$	$-\frac{1}{821}$ 531	0.80 $0.80$	$\overline{248}$ bikz $\approx \overline{66}$ bits $394$ bikz $\approx 104$ bits
$\log_2 n = 11, \log_2 q_L = 51$	128/97	$\frac{\rho_{C1}^2}{100 \cdot \rho_{C1}^2}$	75 3500	825 534	0.80 0.80	$178 \text{ bikz} \approx 47 \text{ bits}$ $231 \text{ bikz} \approx 61 \text{ bits}$
$\log_2 n = 11, \log_2 q_L = 35$	192/162	$ \begin{array}{c} -\frac{1}{\rho_{C1}^2} \\ 100 \cdot \rho_{C1}^2 \end{array} $	-75 $3500$	$-\frac{1}{825}$ $-\frac{1}{534}$	0.80 $0.80$	$\bar{3}1\bar{4}$ bikz $\approx 8\bar{3}$ bits $395$ bikz $\approx 105$ bits
$\log_2 n = 11, \log_2 q_L = 27$	256/226	$\begin{array}{c} -\vec{\rho}_{\text{C1}}^2 \\ 100 \cdot \rho_{\text{C1}}^2 \end{array}$	-75 $-3500$	825 534		$447 \text{ bikz} \approx 118 \text{ bits}$ $553 \text{ bikz} \approx 147 \text{ bits}$
$\log_2 n = 12, \log_2 q_L = 101$	128/97	$\rho_{C1}^2$ $100 \cdot \rho_{C1}^2$	75 3500	827 535	0.80 0.80	$258 \text{ bikz} \approx 68 \text{ bits}$ $291 \text{ bikz} \approx 77 \text{ bits}$
$\log_2 n = 12, \log_2 q_L = 70$	192/161	$-\frac{1}{\rho_{\text{C1}}^2}$ $100 \cdot \rho_{\text{C1}}^2$	$\begin{array}{r} -75 \\ 3500 \end{array}$	$-\frac{1}{827}$		$436 \text{ bikz} \approx 115 \text{ bits}$ $487 \text{ bikz} \approx 129 \text{ bits}$
$\log_2 n = 12, \log_2 q_L = 54$	256/227	$ \begin{array}{ccc}  & \overline{\rho_{\text{C1}}^2} \\ 100 \cdot \rho_{\text{C1}}^2 \end{array} $	$75 \\ 3500$	-827 $-535$		$615 \text{ bikz} \approx 163 \text{ bits}$ $683 \text{ bikz} \approx 181 \text{ bits}$

Fig. 11: Concrete security of hybrid attacks after observing decryptions of Class 1 ciphertexts. For each parameter set, the second column provides the target security as well as the number of bits of security computed by the tool of [16]. The third column indicates the noise-flooding noise added before returning the decryption to the adversary.  $\rho_{C1}^2$  is the variance of the noise that is already present in a ciphertext obtained from evaluating a Class 1 circuit on fresh encryptions. The fourth column indicates the number of decryptions observed by the adversary. The fifth column indicates the number of coordinates of the LWE secret that are guessed by the adversary. The sixth column indicates the success probability of the attack, which corresponds to the probability that all guesses are correct, conditioned on the events in (17) and (20) occurring. The final column provides the reduced security level after the attack in terms of bikz (see [16]) and bit-security.

Parameter Set	Orig Security	Noise Var	Num Queries	Num Guess		Final Security
$\log_2 n = 13, \log_2 q_L = 202$	128/96	$\begin{array}{c} \rho_{C1}^2 \\ 100 \cdot \rho_{C1}^2 \end{array}$	75 3500	828 535	0.80 0.80	$305 \text{ bikz} \approx 81 \text{ bits}$ $323 \text{ bikz} \approx 86 \text{ bits}$
$\log_2 n = 13, \log_2 q_L = 141$	192/159	$\begin{array}{c} -\frac{1}{\rho_{C1}^2} \\ 100 \cdot \rho_{C1}^2 \end{array}$	$75 \\ 3500$	828 535		$508 \text{ bikz} \approx 135 \text{ bits}$ $537 \text{ bikz} \approx 142 \text{ bits}$
$\log_2 n = 13, \log_2 q_L = 109$	256/225	$\begin{array}{c} -\vec{\rho}_{\text{C1}}^2 \\ 100 \cdot \rho_{\text{C1}}^2 \end{array}$	$\begin{array}{c} -75 \\ 3500 \end{array}$	828 535		$7\overline{16}$ bikz $\approx \overline{190}$ bits $756$ bikz $\approx 200$ bits
$\log_2 n = 14, \log_2 q_L = 411$	128/93	$\rho_{C1}^2$ $100 \cdot \rho_{C1}^2$	75 3500	828 536	0.80 0.80	$323 \text{ bikz} \approx 86 \text{ bits}$ $333 \text{ bikz} \approx 88 \text{ bits}$
$\log_2 n = 14, \log_2 q_L = 284$	192/158	$ \begin{array}{c} -\frac{1}{\rho_{C1}^2} \\ 100 \cdot \rho_{C1}^2 \end{array} $	$-\frac{75}{75}$ $-\frac{1}{3500}$	$-\frac{1}{828}$		$547 \text{ bikz} \approx 145 \text{ bits}$ $562 \text{ bikz} \approx 149 \text{ bits}$
$\log_2 n = 14, \log_2 q_L = 220$	256/222	$\begin{array}{c} -\overline{\rho}_{\text{C1}}^2 \\ 100 \cdot \rho_{\text{C1}}^2 \end{array}$	$\begin{array}{c} -75 \\ 3500 \end{array}$	828 536		770 bikz $\approx 204$ bits 791 bikz $\approx 210$ bits
$\log_2 n = 15, \log_2 q_L = 827$	128/92	$\rho_{C1}^2$ $100 \cdot \rho_{C1}^2$	75 3500	829 536	0.80	$334 \text{ bikz} \approx 88 \text{ bits}$ $339 \text{ bikz} \approx 90 \text{ bits}$
$\log_2 n = 15, \log_2 q_L = 571$	192/156	$-\frac{1}{\rho_{\text{C1}}^2}$ $100 \cdot \rho_{\text{C1}}^2$	$-\frac{75}{75}$ 3500	$-\frac{1}{829}$		$565 \text{ bikz} \approx 150 \text{ bits}$ $573 \text{ bikz} \approx 152 \text{ bits}$
$\log_2 n = 15, \log_2 q_L = 443$	256/220	$ \begin{array}{ccc}  & \overline{\rho_{\text{C1}}^2} \\ 100 & \overline{\rho_{\text{C1}}^2} \end{array} $	$75 \\ 3500$	829 536		$797 \text{ bikz} \approx 211 \text{ bits}$ $808 \text{ bikz} \approx 214 \text{ bits}$
$\log_2 n = 17, \log_2 q_L = 2400$	140/146	$\begin{array}{c} \rho_{\text{C1}}^2 \\ 100 \cdot \rho_{\text{C1}}^2 \end{array}$	85 4000	1722 1334		$540 \text{ bikz} \approx 143 \text{ bits}$ $542 \text{ bikz} \approx 144 \text{ bits}$
$\log_2 n = 17, \log_2 q_L = 2000$		$ \begin{array}{c} -\overline{\rho_{\text{C1}}^2} \\ 100 \cdot \rho_{\text{C1}}^2 \end{array} $	85 4000	1722 1334		$693 \text{ bikz} \approx 184 \text{ bits}$ $696 \text{ bikz} \approx 184 \text{ bits}$

Fig. 11: Concrete security of hybrid attacks after observing decryptions of Class 1 ciphertexts, continued. For each parameter set, the second column provides the target security as well as the number of bits of security computed by the tool of [16]. The third column indicates the noise-flooding noise added before returning the decryption to the adversary.  $\rho_{C1}^2$  is the variance of the noise that is already present in a ciphertext obtained from evaluating a Class 1 circuit on fresh encryptions. The fourth column indicates the number of decryptions observed by the adversary. The fifth column indicates the number of coordinates of the LWE secret that are guessed by the adversary. The sixth column indicates the success probability of the attack, which corresponds to the probability that all guesses are correct, conditioned on the events in (17) and (20) occurring. The final column provides the reduced security level after the attack in terms of bikz (see [16]) and bit-security.

Parameter	Orig	Noise	Num	Num	Succ	Final
Set	Security	Var	Queries			
	Security		Queries	Guess	1 100	
$\log_2 n = 10, \log_2 q_L = 25$	128/102	$ ho_{C2}^2$	95	892	0.80	$53 \text{ bikz} \approx 14 \text{ bits}$
$\log_2 n = 10, \log_2 q_L = 20$	120/102	$100 \cdot \rho_{C2}^2$	4300	620	0.80	$125 \text{ bikz} \approx 33 \text{ bits}$
$\log_2 n = 10, \log_2 q_L = 17$	192/170	$ ho_{C2}^2$	95	892	0.80	$134 \text{ bikz} \approx 36 \text{ bits}$
$\log_2 n = 10, \log_2 q_L = 11$		$100 \cdot \rho_{\text{C2}}^2$	$_{-}4300$	620	0.80	$235 \text{ bikz} \approx 62 \text{ bits}$
$\log_2 n = 10, \log_2 q_L = 13$	256/234	$ ho_{C2}^2$	95	892	0.80	$216 \text{ bikz} \approx 57 \text{ bits}$
$\log_2 n = 10, \log_2 q_L = 13$	200/204	$100 \cdot \rho_{C2}^2$	4300	620	0.80	$345 \text{ bikz} \approx 91 \text{ bits}$
1 11 1 21	100 /05	$ ho_{C2}^2$	95	896	0.80	$166 \text{ bikz} \approx 44 \text{ bits}$
$\log_2 n = 11, \log_2 q_L = 51$	128/97	$100 \cdot \rho_{C2}^2$	4300	622	0.80	214 bikz $\approx 57$ bits
11 1 25	100/100	$  \overline{\rho}_{\text{C2}}^{2}$	<sub>95</sub> -	$-89\bar{6}$	0.80	$297 \text{ bikz} \approx 79 \text{ bits}$
$\log_2 n = 11, \log_2 q_L = 35$	192/162	$100 \cdot \rho_{C2}^2$	4300	622	0.80	$370 \text{ bikz} \approx 98 \text{ bits}$
$\log_2 n = 11, \log_2 q_L = 27$	256/226	$ \overline{ ho}_{C2}^2$ $-$	95	896	0.80	$424 \text{ bikz} \approx 112 \text{ bits}$
$\log_2 n = 11, \log_2 q_L = 21$	250/220	$100 \cdot \rho_{C2}^2$	4300	622	0.80	$521 \text{ bikz} \approx 138 \text{ bits}$
1 101	100/05	$\rho_{C2}^2$	95	897	0.80	$251 \text{ bikz} \approx 67 \text{ bits}$
$\log_2 n = 12, \log_2 q_L = 101$	128/97	$100 \cdot \rho_{C2}^2$	4300	623	0.80	$281 \text{ bikz} \approx 74 \text{ bits}$
10.1		$-\frac{1}{\rho_{C2}^2}$	<sub>95</sub> -	$-\bar{897}$	0.80	$4\overline{25}$ bikz $\approx \overline{113}$ bits
$\log_2 n = 12, \log_2 q_L = 70$	192/161	$100 \cdot \rho_{C2}^2$	4300	623	0.80	$471 \text{ bikz} \approx 125 \text{ bits}$
lam m 19 lam m 54	101 74 076/007	$ \rho_{C2}^2$ $-$	95	-897	0.80	$601 \text{ bikz} \approx 159 \text{ bits}$
$\log_2 n = 12, \log_2 q_L = 54$	200/221	$100 \cdot \rho_{C2}^2$	4300	623	0.80	664 bikz $\approx 176$ bits
$\log_2 n = 12, \log_2 q_L = 54$	256/227					

Fig. 12: Concrete security of hybrid attacks after observing decryptions of Class 2 ciphertexts. For each parameter set, the second column provides the target security as well as the number of bits of security computed by the tool of [16]. The third column indicates the noise-flooding noise added before returning the decryption to the adversary.  $\rho_{\text{C2}}^2$  is the variance of the noise that is already present in a ciphertext obtained from evaluating a Class 2 circuit on fresh encryptions. The fourth column indicates the number of decryptions observed by the adversary. The fifth column indicates the number of coordinates of the LWE secret that are guessed by the adversary. The sixth column indicates the success probability of the attack, which corresponds to the probability that all guesses are correct, conditioned on the events in (17) and (20) occurring. The final column provides the reduced security level after the attack in terms of bikz (see [16]) and bit-security. The (encoded) message magnitude is equal to  $n \cdot \sqrt{\ell/3}$  in all rows, where  $\ell$  is set to 20.

Parameter	Orig	Noise	Num	Num	Succ	Final
Set	Security		Queries			
	Security		Queries	Guess	1 100	
$\log_2 n = 13, \log_2 q_L = 202$	199/06	$ ho_{C2}^2$	95	898	0.80	$301 \text{ bikz} \approx 80 \text{ bits}$
$\log_2 n = 13, \log_2 q_L = 202$	128/96	$100 \cdot \rho_{C2}^2$	4300	624	0.80	$318 \text{ bikz} \approx 84 \text{ bits}$
log m = 12 log g = 141	192/159	$ \overline{ ho}_{C2}^2$ $-$	95	-898	0.80	$502 \text{ bikz} \approx 133 \text{ bits}$
$\log_2 n = 13, \log_2 q_L = 141$	192/139	$100 \cdot \rho_{C2}^2$	4300	624	0.80	529 bikz $\approx$ 140 bits
$\log_2 n = 13, \log_2 q_L = 109$	256/225	$ ho_{C2}^{2}$	95	898	0.80	$709 \text{ bikz} \approx 188 \text{ bits}$
$\log_2 n = 15, \log_2 q_L = 109$	230/223	$100 \cdot \rho_{C2}^2$	4300	624	0.80	745 bikz $\approx$ 197 bits
1	100/02	$ ho_{C2}^2$	95	898	0.80	$321 \text{ bikz} \approx 85 \text{ bits}$
$\log_2 n = 14, \log_2 q_L = 411$	128/93	$100 \cdot \rho_{C2}^2$	4300	624	0.80	330 bikz $\approx 87$ bits
lam m 14 lam m 294	100/150	$ \rho_{C2}^2$ $-$	95	898	0.80	$543 \text{ bikz} \approx 144 \text{ bits}$
$\log_2 n = 14, \log_2 q_L = 284$	192/158	$100 \cdot \rho_{C2}^2$	4300	624	0.80	557 bikz $\approx$ 148 bits
1 14.1 200	050/000	$  \overline{ ho}_{C2}^2$ $-$	<sub>95</sub> -	$-\bar{898}$	0.80	$7\overline{66}$ bikz $\approx 20\overline{3}$ bits
$\log_2 n = 14, \log_2 q_L = 220$	256/222	$100 \cdot \rho_{C2}^2$	4300	624	0.80	786 bikz $\approx$ 208 bits
1 15 1 207	100/00	$\rho_{C2}^2$	95	899	0.80	$333 \text{ bikz} \approx 88 \text{ bits}$
$\log_2 n = 15, \log_2 q_L = 827$	128/92	$100 \cdot \rho_{C2}^2$	4300	624	0.80	337 bikz $\approx 89$ bits
lam m 15 lam m 571	109/156	$-\frac{1}{\rho_{C2}^2}$	95	899	0.80	$564 \text{ bikz} \approx 149 \text{ bits}$
$\log_2 n = 15, \log_2 q_L = 571$	192/156	$100 \cdot \rho_{C2}^2$	4300	624	0.80	571 bikz $\approx$ 151 bits
15 1		$ \overline{\rho}_{C2}^2$ $-$	95	-899	0.80	$795 \text{ bikz} \approx 211 \text{ bits}$
$\log_2 n = 15, \log_2 q_L = 443$	256/220	$100 \cdot \rho_{C2}^2$	4300	624	0.80	805 bikz $\approx$ 213 bits
		$ ho_{C2}^2$	110	2508	0.80	$535 \text{ bikz} \approx 142 \text{ bits}$
$\log_2 n = 17, \log_2 q_L = 2400$	140/146	$100 \cdot \rho_{C2}^2$	4900	1551		$541 \text{ bikz} \approx 143 \text{ bits}$
		$-\frac{1}{\rho_{C2}^2}$	$\bar{1}\bar{1}\bar{0}$	2508	0.80	$688 \text{ bikz} \approx 182 \text{ bits}$
$\log_2 n = 17, \log_2 q_L = 2000$	193/187	$100 \cdot \rho_{\text{C2}}^2$	4900	1551	0.80	695 bikz $\approx$ 184 bits

Fig. 12: Concrete security of hybrid attacks after observing decryptions of Class 2 ciphertexts, continued. For each parameter set, the second column provides the target security as well as the number of bits of security computed by the tool of [16]. The third column indicates the noise-flooding noise added before returning the decryption to the adversary.  $\rho_{C2}^2$  is the variance of the noise that is already present in a ciphertext obtained from evaluating a Class 2 circuit on fresh encryptions. The fourth column indicates the number of decryptions observed by the adversary. The fifth column indicates the number of coordinates of the LWE secret that are guessed by the adversary. The sixth column indicates the success probability of the attack, which corresponds to the probability that all guesses are correct, conditioned on the events in (17) and (20) occurring. The final column provides the reduced security level after the attack in terms of bikz (see [16]) and bit-security. The (encoded) message magnitude is equal to  $n \cdot \sqrt{\ell/3}$  in all rows, where  $\ell$  is set to 20.

Parameter Set	Original Security		Final Security
$\log_2 n = 10, \log_2 q_L = 25$	128/102.34 (386.21)		$386.21 \text{ bikz} \approx 102.34 \text{ bits}$
$\frac{\log_2 n = 10, \log_2 q_L = 17}{\log_2 n = 10, \log_2 q_L = 13}$	$\frac{192/170.04 (641.65)}{256/234.29 (884.13)}$		$641.65 \text{ bikz} \approx 170.04 \text{ bits}$ $884.13 \text{ bikz} \approx 234.29 \text{ bits}$
$\frac{\log_2 n = 11, \log_2 q_L = 51}{\log_2 n = 11, \log_2 q_L = 51}$	128/96.84 (365.43)	$ ho_{stat}^2$	$365.43 \text{ bikz} \approx 96.84 \text{ bits}$
$\frac{\log_2 n = 11, \log_2 q_L = 35}{\log_2 n = 11, \log_2 q_L = 27}$	$\begin{array}{c ccccccccccccccccccccccccccccccccccc$		$612.49 \text{ bikz} \approx 162.31 \text{ bits}$ $853.25 \text{ bikz} \approx 226.11 \text{ bits}$
$\frac{\log_2 n = 12, \log_2 q_L = 101}{\log_2 n = 12, \log_2 q_L = 101}$	128/96.81 (365.34)		$365.34 \text{ bikz} \approx 96.81 \text{ bits}$
$\frac{\log_2 n}{\log_2 n} = \frac{12}{12}, \log_2 q_L = \frac{70}{54}$	$\frac{192/161.41}{256/227.10} \underbrace{(609.11)}_{(856.98)}$		$609.11 \text{ bikz} \approx 161.41 \text{ bits}$ $856.98 \text{ bikz} \approx 227.10 \text{ bits}$
$\log_2 n = 13, \log_2 q_L = 202$	128/96.11 (362.66)	$ ho_{stat}^2$	$362.66 \text{ bikz} \approx 96.11 \text{ bits}$
$\frac{\log_2 n}{\log_2 n} = \frac{13}{13}, \frac{\log_2 q_L}{\log_2 q_L} = \frac{141}{109}$	$\frac{192/159.40}{256/224.89} \frac{(601.49)}{(848.63)}$		$601.49 \text{ bikz} \approx 159.40 \text{ bits} $ $848.63 \text{ bikz} \approx 224.89 \text{ bits}$
$\log_2 n = 14, \log_2 q_L = 411$	128/93.37 (352.34)		$352.34 \text{ bikz} \approx 93.37 \text{ bits}$
$\frac{\log_2 n = 14, \log_2 q_L = 284}{\log_2 n = 14, \log_2 q_L = 220}$	$\frac{192/157.62}{256/222.42} \frac{(594.78)}{(839.32)}$		$594.78 \text{ bikz} \approx 157.62 \text{ bits}  839.32 \text{ bikz} \approx 222.42 \text{ bits}$
$\log_2 n = 15, \log_2 q_L = 827$	128/92.37 (348.55)		$348.55 \text{ bikz} \approx 92.37 \text{ bits}$
$\frac{\log_2 n = 15, \log_2 q_L = 571}{\log_2 n = 15, \log_2 q_L = 443}$	$\begin{array}{c} 192/156.35 \ (590.00) \\ \hline 256/220.52 \ (832.15) \end{array}$		$590.00 \text{ bikz} \approx 156.35 \text{ bits}$ $832.15 \text{ bikz} \approx 220.52 \text{ bits}$
$\frac{\log_2 n = 17, \log_2 q_L = 2400}{\log_2 n = 17, \log_2 q_L = 2000}$	140/145.88 (550.51)	$ ho_{stat}^2$	$550.51 \text{ bikz} \approx 145.88 \text{ bits}$ $707.17 \text{ bikz} \approx 187.40 \text{ bits}$
$10g_2 n = 17, 10g_2 q_L = 2000$	199/101.40 (101.11)	$ ho_{stat}^2$	101.11 DIKZ ≈ 101.40 DITS

Fig. 13: Concrete security of lattice reduction attacks after observing 1000 decryptions of fresh ciphertexts. For each parameter set, the second column provides the target security as well as the number of bits of security computed by the tool of [16] (bikz are provided in parenthesis). The third column indicates the noise-flooding noise added before returning the decryption to the adversary.  $\rho_{\text{stat}}^2 = 12 \cdot t \cdot 2^{\kappa} \cdot \rho_{\text{fresh}}^2$ , where  $\rho_{\text{fresh}}^2$  is the variance of the noise that is already present in a fresh ciphertext (see Section 4.5.1), and  $\kappa = 30$  is the statistical security parameter. The fourth column indicates the number of decryptions observed by the adversary. The final column provides the reduced security level after the attack in terms of bikz (see [16]) and bit-security.

Parameter	Original	Noise	Final
Set	Security	Variance	Security
$\log_2 n = 10, \log_2 q_L = 25$	128/102.34 (386.21)	$ ho_{stat}^2$	$386.21 \text{ bikz} \approx 102.34 \text{ bits}$
$\log_2 n = 10, \log_2 q_L = 17$	192/170.04 (641.65)	$  ho_{stat}^2$ $-$	$641.65 \text{ bikz} \approx 170.04 \text{ bits}$
$\log_2 n = 10, \log_2 q_L = 13$	256/234.29 (884.13)	$ \rho_{stat}^2$ $-$	$884.13 \text{ bikz} \approx 234.29 \text{ bits}$
$\log_2 n = 11, \log_2 q_L = 51$	128/96.84 (365.43)	$ ho_{stat}^2$	$365.43 \text{ bikz} \approx 96.84 \text{ bits}$
$\log_2 n = 11, \log_2 q_L = 35$	$19\overline{2}/162.\overline{31}$ $\overline{(612.49)}$	$  ho_{stat}^2$ $ -$	$6\overline{12.49}$ bikz $\approx \overline{162.31}$ bits
$\log_2 n = 11, \log_2 q_L = 27$	256/226.11 (853.25)	$ ho_{stat}^2$	$853.25 \text{ bikz} \approx 226.11 \text{ bits}$
$\log_2 n = 12, \log_2 q_L = 101$	128/96.81 (365.34)	$ ho_{stat}^2$	$365.34 \text{ bikz} \approx 96.81 \text{ bits}$
$\log_2 n = 12, \log_2 q_L = 70$	$19\overline{2}/161.4\overline{1} (6\overline{0}9.1\overline{1})$	$  ho_{stat}^2$ $-$	$609.11 \text{ bikz} \approx 161.41 \text{ bits}$
$\log_2 n = 12, \log_2 q_L = 54$	256/227.10 (856.98)	$ ho_{stat}^2$	$856.98 \text{ bikz} \approx 227.10 \text{ bits}$
$\log_2 n = 13, \log_2 q_L = 202$	128/96.11 (362.66)	$ ho_{stat}^2$	$362.66 \text{ bikz} \approx 96.11 \text{ bits}$
$\log_2 n = 13, \log_2 q_L = 141$	192/159.40 (601.49)	$  ho_{stat}^2$ $-$	$601.49 \text{ bikz} \approx 159.40 \text{ bits}$
$\log_2 n = 13, \log_2 q_L = 109$	256/224.89 (848.63)	$ ho_{fresh}^2$	$848.63 \text{ bikz} \approx 224.89 \text{ bits}$
$\log_2 n = 14, \log_2 q_L = 411$	128/93.37 (352.34)	$ ho_{stat}^2$	$352.34 \text{ bikz} \approx 93.37 \text{ bits}$
$\log_2 n = 14, \log_2 q_L = 284$	192/157.62 (594.78)	$ ho_{stat}^2$	$594.78 \text{ bikz} \approx 157.62 \text{ bits}$
$\log_2 n = 14, \log_2 q_L = 220$	256/222.42(839.32)	$ ho_{stat}^2$	$839.32 \text{ bikz} \approx 222.42 \text{ bits}$
$\log_2 n = 15, \log_2 q_L = 827$	128/92.37 (348.55)	$ ho_{stat}^2$	$348.55 \text{ bikz} \approx 92.37 \text{ bits}$
$\log_2 n = 15, \log_2 q_L = 571$	192/156.35 (590.00)	$  ho_{stat}^2$ $-$	$590.00 \text{ bikz} \approx 156.35 \text{ bits}$
$\log_2 n = 15, \log_2 q_L = 443$	256/220.52 (832.15)	$-\overline{ ho}_{stat}^2$	$832.15 \text{ bikz} \approx 220.52 \text{ bits}$
$\log_2 n = 17, \log_2 q_L = 2400$	140/145.88 (550.51)	$ ho_{stat}^2$	$550.51 \text{ bikz} \approx 145.88 \text{ bits}$
$\log_2 n = 17, \log_2 q_L = 2000$		$-\bar{ ho}_{\sf stat}^2$	$707.17 \text{ bikz} \approx 187.40 \text{ bits}$

Fig. 14: Concrete security of lattice reduction attacks after observing 1000 decryptions of Class 1 ciphertexts. For each parameter set, the second column provides the target security as well as the number of bits of security computed by the tool of [16] (bikz are provided in parenthesis). The third column indicates the noise-flooding noise added before returning the decryption to the adversary.  $\rho_{\text{stat}}^2 = 12 \cdot t \cdot 2^{\kappa} \cdot \rho_{\text{Cl}}^2$ , where  $\rho_{\text{Cl}}^2$  is the variance of the noise that is already present in a ciphertext obtained from evaluating a Class 1 circuit on fresh encryptions, and  $\kappa = 30$  is the statistical security parameter.  $\rho_{\text{fresh}}^2$  is the variance of the noise that is already present in a fresh ciphertext (see Section 4.5.1).. The fourth column indicates the number of decryptions observed by the adversary. The final column provides the reduced security level after the attack in terms of bikz (see [16]) and bit-security.

Parameter Set	Original Security	Noise Variance	Final Security
$\begin{array}{c} \log_2 n = 10, \log_2 q_L = 25 \\ \log_2 n = 10, \log_2 q_L = 17 \\ \log_2 n = 10, \log_2 q_L = 13 \end{array}$	$\begin{array}{c} \underline{128/102.34} \ (386.21) \\ \underline{192/170.04} \ (641.65) \\ \underline{256/234.29} \ (884.13) \end{array}$	$ ho_{stat}^2$	$\begin{array}{l} 386.21 \text{ bikz} \approx 102.34 \text{ bits} \\ \hline 641.65 \text{ bikz} \approx 170.04 \text{ bits} \\ 884.13 \text{ bikz} \approx 234.29 \text{ bits} \end{array}$
$\frac{\log_2 n = 11, \log_2 q_L = 51}{\log_2 n = 11, \log_2 q_L = 35}$ $\log_2 n = 11, \log_2 q_L = 27$	$\begin{array}{c} 128/96.84 \ (365.43) \\ \hline 192/162.31 \ (612.49) \\ \hline 256/226.11 \ (853.25) \end{array}$		$365.43 \text{ bikz} \approx 96.84 \text{ bits}  612.49 \text{ bikz} \approx 162.31 \text{ bits}  853.25 \text{ bikz} \approx 226.11 \text{ bits}$
$\begin{array}{ c c c c c }\hline \log_2 n = 12, \log_2 q_L = 101\\ \log_2 n = 12, \log_2 q_L = 70\\ \log_2 n = 12, \log_2 q_L = 54\\ \end{array}$	$\begin{array}{c} 128/96.81 \ (365.34) \\ \hline 192/161.41 \ (609.11) \\ \hline 256/227.10 \ (856.98) \end{array}$		$365.34 \text{ bikz} \approx 96.81 \text{ bits}  \hline 609.11 \text{ bikz} \approx 161.41 \text{ bits}  856.98 \text{ bikz} \approx 227.10 \text{ bits}$
$\begin{array}{ c c c c c }\hline \log_2 n = 13, \log_2 q_L = 202\\ \log_2 n = 13, \log_2 q_L = 141\\ \log_2 n = 13, \log_2 q_L = 109\\ \end{array}$	128/96.11 (362.66) 192/159.40 (601.49) 256/224.89 (848.63)		$362.66 \text{ bikz} \approx 96.11 \text{ bits}  \hline 601.49 \text{ bikz} \approx 159.40 \text{ bits}  848.63 \text{ bikz} \approx 224.89 \text{ bits}$
$\begin{array}{c} \log_2 n = 14, \log_2 q_L = 411 \\ \log_2 n = \overline{14}, \log_2 q_L = 2\overline{84} \\ \log_2 n = \overline{14}, \log_2 q_L = 2\overline{20} \end{array}$	$\begin{array}{c} 128/93.37 \ (352.34) \\ \hline 192/157.62 \ (594.78) \\ \hline 256/222.42 \ (839.32) \end{array}$	$\rho_{stat}^2$	$\begin{array}{c} 352.34 \text{ bikz} \approx 93.37 \text{ bits} \\ 594.78 \text{ bikz} \approx 157.62 \text{ bits} \\ 839.32 \text{ bikz} \approx 222.42 \text{ bits} \end{array}$
$\begin{array}{c} \log_2 n = 15, \log_2 q_L = 827 \\ \log_2 n = \overline{15}, \log_2 q_L = 571 \\ \log_2 n = \overline{15}, \log_2 q_L = 443 \end{array}$	$\begin{array}{c} 128/92.37 \ (348.55) \\ \hline 192/156.35 \ (590.00) \\ \hline 256/220.52 \ (832.15) \end{array}$	$ ho_{stat}^2$	$\begin{array}{c} 348.55 \text{ bikz} \approx 92.37 \text{ bits} \\ \hline 590.00 \text{ bikz} \approx 156.35 \text{ bits} \\ 832.15 \text{ bikz} \approx 220.52 \text{ bits} \end{array}$
$\frac{\log_2 n = 17, \log_2 q_L = 2400}{\log_2 n = 17, \log_2 q_L = 2000}$			$550.51 \text{ bikz} \approx 145.88 \text{ bits} \\ 707.17 \text{ bikz} \approx 187.40 \text{ bits}$

Fig. 15: Concrete security of lattice reduction attacks after observing 1000 decryptions of Class 2 ciphertexts. For each parameter set, the second column provides the target security as well as the number of bits of security computed by the tool of [16] (bikz are provided in parenthesis). The third column indicates the noise-flooding noise added before returning the decryption to the adversary.  $\rho_{\rm stat}^2 = 12 \cdot t \cdot 2^{\kappa} \cdot \rho_{\rm C2}^2$ , where  $\rho_{\rm C2}^2$  is the variance of the noise that is already present in a ciphertext obtained from evaluating a Class 2 circuit on fresh encryptions, and  $\kappa = 30$  is the statistical security parameter. The fourth column indicates the number of decryptions observed by the adversary. The final column provides the reduced security level after the attack in terms of bikz (see [16]) and bit-security. The (encoded) message magnitude is equal to  $n \cdot \sqrt{\ell/3}$  in all rows, where  $\ell$  is set to 20.