On the Concrete Security of Approximate FHE with Noise-Flooding Countermeasures

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

\textsuperscript{1} Intel Labs
flavio@intel.com
\textsuperscript{2} Norwegian University of Science and Technology
anamaria.costache@ntnu.no
\textsuperscript{3} University of Maryland
\{danadach@, hkippen@, llabuff@terpmail., ruitang@\}umd.edu

\textbf{Abstract.} Approximate fully homomorphic encryption (FHE) schemes such as the CKKS scheme (Asiacrypt '17) are popular in practice due to their efficiency and utility for machine learning applications. Unfortunately, Li and Micciancio (Eurocrypt, '21) showed that, while achieving standard semantic (or IND-CPA security), the CKKS scheme is broken under a variant security notion known as IND-CPA\textsuperscript{D}. Subsequently, Li, Micciancio, Schultz, and Sorrell (Crypto '22) proved the security of the CKKS scheme with a noise-flooding countermeasure, which adds Gaussian noise of sufficiently high variance before outputting the decrypted value. However, the variance required for provable security is very high, inducing a large loss in message precision.

In this work, we ask whether there is an intermediate noise-flooding level, which may not be provably secure, but allows to maintain the performance of the scheme, while resisting known attacks. We analyze the security with respect to different adversarial models and various types of attacks. We investigate the effectiveness of lattice reduction attacks, guessing attacks and hybrid attacks with noise-flooding with variance $\rho^2_{\text{arc}}$, the variance of the noise already present in the ciphertext as estimated by an average-case analysis, $100 \cdot \rho^2_{\text{arc}}$, and $t \cdot \rho^2_{\text{arc}}$, where $t$ is the number of decryption queries. For noise levels of $\rho^2_{\text{arc}}$ and $100 \cdot \rho^2_{\text{arc}}$, we find that a full guessing attack is feasible for all parameter sets and circuit types. We find that a lattice reduction attack is the most effective attack for noise-flooding level $t \cdot \rho^2_{\text{arc}}$, but it only induces at most a several bit reduction in the security level.

Due to the large dimension and modulus in typical FHE parameter sets, previous techniques even for estimating the concrete security of these 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.

\textsuperscript{\ast} Supported in part by Intel through the Intel Labs Soteria Research Collaboration

\textsuperscript{\ast\ast} Supported in part by NSF grant \#CNS-2154705 and by Intel through the Intel Labs Crypto Frontiers Research Center.

\textsuperscript{\ast\ast\ast} 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 put forth by Cheon, Kim, Kim, and Song [13]. Their proposed scheme, henceforth referred to as CKKS, is one of the leading schemes in terms of efficiency, in particular in terms of suitability for Machine Learning (ML) tasks, as well as its paralellisation capabilities. Unfortunately, in [26], it was pointed out that approximate FHE schemes come with added risks: While they indeed achieve the standard notion of CPA-security, they can fail against a variant, IND-CPA\(^D\) introduced by Li and Micciancio [26], in which the adversary is given limited access to the decryption oracle. In the same work [26], the authors show that for exact schemes (such as BGV, BFV and TFHE), the notions of IND-CPA\(^D\) and IND-CPA are equivalent.\(^4\)

Noise-flooding techniques have been suggested as a practical countermeasure against 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 [27], 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 security remains very high, and as a result, CKKS may lose some of the efficiency that originally made it attractive in comparison to exact FHE schemes. The main exact FHE schemes today are BGV [11], BFV [10, 19], and TFHE [14]. A precise comparison between the schemes is hard to provide, since these schemes all have different trade-offs in terms of latency, amortized latency and the type of circuits that they can support. CKKS typically performs quite well in terms of amortized latency – see for example [9]. Therefore, while quantifying exactly how CKKS relates to these schemes with a much smaller message precision is not easily done, it certainly loses at least part of its edge over the other schemes in terms of amortized latency when noise-flooding is introduced. To bridge this gap, we ask whether there is an intermediate noise-flooding level, which may fault short of provable security, but which withstands a rigorous security analysis and which affords the efficiency needed in practice.

Our goal in this work is to investigate the concrete security degradation of the CKKS scheme when an adversary observes some number of key-recovery after some number of decryptions have been observed. The notion of “approximate FHE” – fully homomorphic encryption schemes that guarantee only approximate correctness of decryption – was put forth by Cheon, Kim, Kim, and Song [13]. Their proposed scheme, henceforth referred to as CKKS, is one of the leading schemes in terms of efficiency, in particular in terms of suitability for Machine Learning (ML) tasks, as well as its paralellisation capabilities. Unfortunately, in [26], it was pointed out that approximate FHE schemes come with added risks: While they indeed achieve the standard notion of CPA-security, they can fail against a variant, IND-CPA\(^D\) introduced by Li and Micciancio [26], in which the adversary is given limited access to the decryption oracle. In the same work [26], the authors show that for exact schemes (such as BGV, BFV and TFHE), the notions of IND-CPA\(^D\) and IND-CPA are equivalent.\(^4\)

Noise-flooding techniques have been suggested as a practical countermeasure against 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 [27], 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 security remains very high, and as a result, CKKS may lose some of the efficiency that originally made it attractive in comparison to exact FHE schemes. The main exact FHE schemes today are BGV [11], BFV [10, 19], and TFHE [14]. A precise comparison between the schemes is hard to provide, since these schemes all have different trade-offs in terms of latency, amortized latency and the type of circuits that they can support. CKKS typically performs quite well in terms of amortized latency – see for example [9]. Therefore, while quantifying exactly how CKKS relates to these schemes with a much smaller message precision is not easily done, it certainly loses at least part of its edge over the other schemes in terms of amortized latency when noise-flooding is introduced. To bridge this gap, we ask whether there is an intermediate noise-flooding level, which may fault short of provable security, but which withstands a rigorous security analysis and which affords the efficiency needed in practice.

Our goal in this work is to investigate the concrete security degradation of the CKKS scheme when an adversary observes some number of key-recovery after some number of decryptions have been observed. The optimal setting of \(\rho^2\) in terms of message precision is to set \(\rho^2\) equal to the variance of the noise already present in an honestly generated ciphertext, since this means that only 1 additional 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 also 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).

In the next section, we discuss in more detail the adversarial model we consider and our methodology for determining the concrete hardness of key-recovery after some number of decryptions have been observed. We emphasize that prior methods for providing concrete hardness estimates such as [17], require performing 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 \(256K \times 256K\) and thus several hundred terrabytes are required to naively store the values (this is assuming 64-bit precision, whereas in our experimental results in Section 9.2, we find that up to 2,000 bit precision is required for meaningful results). Therefore, one of our main contributions is developing new tools to provide fast and accurate estimates that do not require these high-dimensional matrix operations.

\(^4\) In a recent work [12], this is called into question, as the authors point out that 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 [12] 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 TFHE.
1.1 Our Methodology

In our work, we consider several types of adversarial models and attacks. The choice of adversarial model governs how the adversary is allowed to interact with the encryption/eval/decryption functionalities, while the choice of attack corresponds to the way the information obtained from the decryption oracle is used to perform key recovery. Importantly, we note that all our adversarial models are semi-honest in the sense that the adversary obtains fresh ciphertexts sampled from the correct distribution, and only requests computations on circuits whose inputs are fresh, independent, ciphertexts. Thus, the estimates of the noise present in the ciphertext (which determine the variance for the noise-flooding) and the actual noise present in every ciphertext submitted for decryption are consistent. Such a constraint could be enforced in practice by requiring zero knowledge proofs of well-formedness of fresh ciphertexts and/or signatures of designated parties to be checked before decryption is performed, or Verifiable Computation (VC) techniques [8,20,21]. This is different from attacks such as those of Guo, Nabokov, Suvanto, and Johansson [22], which work by constructing adversarial ciphertexts with noise distribution that is far from the noise distribution estimated during the noise flooding step.

The first adversarial model we consider allows the adversary to run the encryption algorithm honestly and to request decryptions of “fresh” ciphertexts, i.e. the adversary queries the decryption oracle on the identity circuit. The second adversarial model we consider allows the adversary black-box access to the encryption algorithm, and to request decryptions of ciphertexts resulting from the evaluation of circuits from one of two circuit classes on the encryptions. The attacks we consider are lattice reduction attacks, guessing attacks, and hybrid attacks. We elaborate below on the adversarial models, attacks, and our methodology for analyzing the concrete hardness of each attack.

Decryption Queries on Identity Circuits. We start by considering an attacker who submits a number $t$ of fresh ciphertexts for decryption. This can also be viewed as an attacker who requests a decryption of a ciphertext corresponding to the evaluation of the identity circuit on a fresh encryption. In the original paper of Li and Micciancio [26], these simple IND-CPA$^D$ attacks were already shown to allow full key-recovery against CKKS. We, however, consider a strengthening of their adversarial model. They allowed the adversary only black-box access to the encryption oracle. We assume that the adversary obtains “white-box” access to the encryption oracle, namely the internal randomness of the encryption is returned to the adversary, along with the ciphertext. Once the ciphertexts have been created, our attacker observes decryptions with “noise-flooding” added before the message is returned, where the noise is a centered Gaussian of variance $\rho^2$. After observing this information, we consider the concrete security of a key recovery attack under three types of attacks: (1) Lattice Reduction attacks, (2) Guessing attacks, (3) Hybrid attacks.

Lattice Reduction Attacks. Here we assume that the adversary embeds the original LWE instance and the additional information that it obtains (which we refer to as “hints”) into a DBDD instance (introduced by [17]), which is then transformed into a u-SVP instance. We note that the “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 [17] 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. We provide concrete security guarantees in terms of bikz (i.e. BKZ-$\beta$) required to solve the final u-SVP instance, as well as the bit-security.

Importantly, although the attack template proceeds as the one outlined in [17], our analysis of the attack differs. To obtain concrete security estimates as in [17], 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 9.2). 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
coefficients, which are individually uniformly random between $[0, 1)$.

We use the Eigenvalue Interlacing Theorem (see Section 2) 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 5). 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 [25] (See Section 6 and Lemma 6.1).

**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 [25] (See Section 6 and Lemma 6.1).

**Hybrid Attacks.** Here the attacker guesses $g < n$ number of coordinates as above, but cannot guess $n$ of them w.h.p. 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. [25]) 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 7 and Lemma 7.1).

**Broader Classes of Circuits.** We extend our analysis to broader classes of circuits (see Section 8 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 a Gaussian distribution. 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.

For circuits in Class 1 and 2, our adversarial model is captured by the IND-CPA$_D$-definition. I.e. the adversary does not need to know the internal randomness used by the encryption process, and can launch the attack with only black-box access to the encryption algorithm. The analysis in this case is facilitated by the fact that it was shown in prior work [16,7] that after a rescale step, the rounding noise (which can be publicly computed) dominates the noise present in the ciphertext. 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 correspond to noisy linear equations on both the LWE secret and error. As before, we consider three types of attacks for each of the two classes of circuits: (1) Lattice Reduction attacks, (2) Guessing attacks, (3) Hybrid attacks.

**1.2 Summary of Experimental Results**

We performed extensive experimentation for a wide range of parameter sets proposed by the homomorphic encryption.org standards [2], as well as a larger parameter set with a ring dimension of $\log_2 n = 17$. In Section 9, we provide an experimental validation of Lemma 5.1, as well as tables detailing the effectiveness
of each of the three attack types on fresh ciphertexts (identity circuits) at various noise-flooding levels: $\rho_{\text{circ}}^2$—the noise variance already present in a ciphertext—100 $\cdot$ $\rho_{\text{circ}}^2$, and $t \cdot \rho_{\text{circ}}^2$, where $t$ is the number of decryptions the attacker may observe. For additional tabular data on Class 1 and 2 circuits, readers may consult with our supplementary material, section A.

In Section 10, we provide a graphical representation of our results and highlight our key findings. Most notably, we find that with noise-flooding levels of $\rho_{\text{circ}}^2$ and 100 $\cdot$ $\rho_{\text{circ}}^2$, full guessing attacks are feasible after observing a sufficient number of decryption queries (at most $\sim$ 100K needed), for all parameter sets and types of circuits considered. On the other hand, for noise level of $t \cdot \rho_{\text{circ}}^2$, lattice reduction attacks are the only effective attacks. Given the above, a noise-flooding magnitude of $\alpha \cdot t \cdot \rho_{\text{circ}}^2$, where $\rho_{\text{circ}}^2$ is the average-case noise present in a ciphertext output by a circuit $\text{circ}$, appears sufficient to preserve security when $t$ decryptions are made available to the adversary. Tuning $0 < \alpha \leq 1$ establishes a way to enforce security–precision tradeoffs in concrete applications. Finally, we note that all attacks become less effective as $\log_2(n)$ increases. Establishing the “appropriate” noise-flooding values will therefore depend on the application itself, on the number of decryption queries $t$ that may be available to an adversary, as well as the FHE parameters, in particular the ring dimension $\log_2(n)$.

1.3 Related Work

After the advent of the CKKS scheme, Li and Micciancio [26] demonstrated it is insecure under a variant of the IND-CPA security notion, which they called IND-CPA$^{\log}$. Their work left the door open as to whether noise-flooding countermeasures, in which additional noise is added to the decrypted message before it is returned, can patch the vulnerability. Li, Micciancio, Schultz, and Sorrell [27] proved the security of the CKKS scheme with the noise-flooding countermeasure for Gaussian noise with sufficiently high variance. However, the variance required for provable security is very high, inducing a large loss in message precision. Later, the inherent noise already present in a CKKS ciphertext was analyzed closely in [15]. This analysis allows for a better understanding of how much message precision is lost via the noise-flooding countermeasures.

The tools of incorporating side information on the LWE secret/error into a lattice reduction attack were developed in [17] 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 uSVP 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 [18], but those techniques similarly do not scale to FHE-size parameter sets.

The work of Kim, Lee, Seo, and Song [24] 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 [24] 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 [24] is for decisional LWE, whereas our attacks are for the search LWE problem, which makes the two results somewhat incomparable.

Two recent works [22,12] 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 [22,12] 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.

2 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 \( I_d \) the identity matrix of dimension \( d \). We denote by \( \{ e_i \}_{i \in [n]} \) the standard basis vectors in dimension \( n \).

We use the notation \( R_q \) to denote the ring \( \mathbb{Z}[x]/\langle \Phi_m(x), q \rangle \), 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 \(( \text{mod} \ q) \) (usually centered around 0).

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

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

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

\[
B = \left( \begin{array}{cc}
\frac{1}{\sqrt{2}} I & \frac{1}{\sqrt{2}} J \\
\frac{1}{\sqrt{2}} J & -\frac{1}{\sqrt{2}} I
\end{array} \right)
\]

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

The canonical embedding of \( a \in \mathbb{Q}[x]/\langle \Phi_m(x) \rangle \) into \( \mathbb{C}^n \) is the vector of evaluations of \( a \) at the roots of \( \Phi_m(x) \). Specifically \( \sigma(a) = [a(\zeta^i)]_{j \in \mathbb{Z}_m^*} \), 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 \( a \), we can express the canonical embedding transformation as a linear transformation \( aV \).

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 \( I_{\dim} \), the \( \dim \) variables are all independent.

DBDD and concrete hardness estimates. A DBDD instance (defined in [17]) 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 [17]). The prior work of [17] showed how to transform a DBDD instance into a uSVP instance with lattice \( \Lambda' \) using the homogenization and isotropization steps, and further showed that the secret vector of this uSVP instance has expected squared norm \( ||s||^2 = \dim(\Lambda') \). Thus, standard techniques can be used to estimate the hardness of the resulting uSVP instance, where hardness is measured in terms of the “bikz” or BKZ-\( \beta \) required to find the unique solution. In particular, following [3,6,17], \( \beta \) can be estimated as the minimum integer that satisfies

\[
\sqrt{\beta} \leq \delta^{2\beta - \dim(\Lambda')^{-1}} \text{Vol}(\Lambda')^{1/\dim(\Lambda')}
\]

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

The CKKS scheme. See Appendix 3 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 [15], we also present the noise variance in a fresh CKKS ciphertext, and in a ciphertext resulting from a multiplication and rescale operation (See Appendix 3.5).
3 The CKKS scheme [13]

Let $\chi$ be a discrete Gaussian of standard deviation $\sigma = 3.2$. We denote by $\mathcal{O}(\rho)$ 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 $S$. This is the uniform ternary distribution.

We assume a sequence of moduli $q_\ell, \ldots, q_0$. After $\ell$ levels of multiplication, we obtain level $\ell$ ciphertexts with moduli $q_\ell$, where $\ell = L - \ell$. We note that, although we present encryption as being performed at the “top” level $L$, it can be performed at any level $\ell$.

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

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

**EvaluationKeyGen($sk, w$):** Set $s = sk$. Sample $a' \leftarrow R_{Q_L}$, $(Q = P_{qL})$ uniformly at random and $e' \leftarrow \chi$.

Output $evk = ((-a's + e' + Ps^2)_{Q}, a')$.

**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 $ct = ([m + pv + e_0]_{q}, [p_1 v + e_1]_{q})$.

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

**Add($ct_0, ct_1$):** Given two level $\ell$ ciphertexts, output $ct = ((ct_0[0] + ct_1[0])_{q\ell}, (ct_0[1] + ct_1[1])_{q\ell})$.

**Pre-Multiply($ct_0, ct_1$):** Given two level $\ell$ ciphertexts, set

$$
d_0 = [ct_0[0]ct_1[0]]_{q\ell}
$$

$$
d_1 = [ct_0[0]ct_1[1] + ct_0[1]ct_1[0]]_{q\ell}
$$

$$
d_2 = [ct_0[1]ct_1[1]]_{q\ell}
$$

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


Let $evk[0] = -a's + e' + Ps^2$ and $evk[1] = a'$. Set

$$
e_0' = [d_0 + P^{-1} \cdot d_2 \cdot (-a's + e' + Ps^2)]_{q\ell}
$$

$$
e_1' = [d_1 + P^{-1} \cdot d_2 \cdot a']_{q\ell}
$$

Output $ct' = (e_0', e_1')$.

**Rescale($ct, A$):** Given level a level $\ell$ ciphertext as input, let $ct = (c_0, c_1)$. Set $c_0' = [[c_0 A \chi]]_{q\ell-1}$ and $c_1' = [[c_1 A \chi]]_{q\ell-1}$. Output $ct = (c_0', c_1')$.

3.1 Decrypting a fresh ciphertext

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

$$
\text{Decrypt}(ct, sk) = c_0 + sc_1 \pmod{q_L}
$$

$$
= m + pv + e_0 +svp_1 + se_1
$$

$$
= m + ve + e_0 + se_1.
$$

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 $\mathcal{O}(\rho)$.  

7
3.2 Decrypting a multiplication, no rescale

Let $\text{ct} = (c_0, c_1)$ and $\text{ct}' = (c'_0, c'_1)$ 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 $\text{Pre-Mult}$ is

$$(d_0, d_1, d_2) = (c_0c'_0, c_0c'_1 + c'_0c_1, c_1c'_1).$$

Note that this decrypts as

$$d_0 + sd_1 + s^2d_2 = (c_0 + c_1s)(c'_0 + sc'_1)$$
$$= \frac{m^{2^{\ell+1}}}{\Delta^{2\ell+1-2}} + \hat{E},$$

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

$$C_0 = d_0 + \left[P^{-1} \cdot d_2 \cdot (-a's + e' + Ps^2)\right]$$
$$= d_0 + P^{-1} \cdot d_2 \cdot (-a's + e' + Ps^2) + \epsilon_0$$
$$C_1 = d_1 + \left[P^{-1} \cdot d_2 \cdot a'\right]$$
$$= d_1 + P^{-1} \cdot d_2 \cdot a' + \epsilon_1,$$

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

$$C_0 + sC_1 = d_0 + P^{-1}d_2(-a's + e' + Ps^2) + \epsilon_0 + sd_1 + sP^{-1}d_2a' + se_1$$
$$= d_0 + sd_1 + s^2d_2 + (\epsilon_0 + \epsilon_2s) + P^{-1}d_2e'$$
$$= \frac{m^{2^{\ell+1}}}{\Delta^{2\ell+1-2}} + \hat{E} + (\epsilon_0 + \epsilon_1s) + P^{-1}d_2e'.$$

It has been shown that for all the FHE parameter sets we consider, the error above is dominated by $\hat{E} = E' \cdot \frac{m^{2^\ell}}{\Delta^{2\ell-1}} + E' \cdot \frac{m^{2^\ell}}{\Delta^{2\ell-1}}$ [16,7].

3.3 Decrypting a multiplication, with rescale

From the previous subsection, we have that the noise after a $\text{Pre-Mult}$ and a $\text{Relin}$ is

$$C_0 + sC_1 = \frac{m^{2^{\ell+1}}}{\Delta^{2\ell+1-2}} + E + (\epsilon_0 + \epsilon_1s) + P^{-1}d_2e'.$$

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

$$(C_0, C_1) = \text{Relin}(\text{Pre-Mult}(\text{ct, ct}')).$$

Let $(C'_0, C'_1) = \text{Rescale}(C_0, C_1) = \left([\left[ \frac{C_0}{2} \right]]_{q_{\ell-1}}, \left[ \frac{C_1}{2} \right]_{q_{\ell-1}} \right)$. Then
\[
\text{Dec}((C_0', C_1'), \sk) = C_0' + sC_1'
\]
\[
= \left\lfloor \frac{C_0}{\Delta} \right\rfloor + s \left\lfloor \frac{C_1}{\Delta} \right\rfloor
\]
\[
= \frac{C_0}{\Delta} + s \frac{C_1}{\Delta} + \delta_0 + s\delta_1
\]
\[
= \frac{1}{\Delta} (C_0 + sC_1) + \delta_0 + s\delta_1
\]
\[
= \frac{1}{\Delta} \left( \frac{m^{2\ell+1}}{\Delta^{2\ell+2} - 2} + E + (\epsilon_0 + \epsilon_1 s) + P^{-1}d_2 e' \right) + \delta_0 + s\delta_1
\]
\[
= \frac{m^{2\ell+1}}{\Delta^{2\ell+1} + 1} + \frac{1}{\Delta} (E + (\epsilon_0 + \epsilon_1 s) + P^{-1}d_2 e') + \delta_0 + s\delta_1,
\]

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

It was observed in [16,7] 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 \( \evk[0] = [-a's + e' + Ps^2]_Q \) but not \( \evk[1] \), 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 \( \evk[1] \) 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 \( \evk \).

### 3.4 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 \( q_j \). Then, for any level \( i \), the ciphertext modulus \( Q_i = \prod_{j=0}^{i} q_j \). We encrypt “at the top” level \( Q_L \), and go “down” one level after each multiplication.

Let \( \ct_0 \) and \( \ct_1 \) 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 \( \ct_0 \) and \( \ct_1 \), 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 \([-0.5, 0.5]\).

\[
\text{Dec}(\ct_0, \sk) = \frac{m^{2\ell}}{\Delta^{2\ell-1}} + E_{\ell,0} \pmod{q_\ell}
\]
\[
\text{Dec}(\ct_1, \sk) = \frac{m^{2\ell}}{\Delta^{2\ell-1}} + E_{\ell,1} \pmod{q_\ell}.
\]

Re-using the analysis from Section 3.2, we have that the error after multiplication without rescale is dominated by:

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

### 3.5 CKKS Error Estimation

The following formulas are taken from [15] and will be useful in our work.
3.5.1 Fresh ciphertext The variance of the error of a fresh ciphertext with error distribution \(N(0, \sigma_v^2 I_n)\) and ternary secret distribution (over domain \(\{-1, 0, 1\}\)) of variance \(2/3\) is approximated as

\[
\rho_{\text{fresh}}^2 = \left(\frac{4}{3}n + 1\right)\sigma_v^2.
\]

3.5.2 Multiplication with rescale Multiplication of two ciphertexts with rescale results in a ciphertext with error of the following form

\[
B_{\text{final error}} = \Delta^{-1}(B_{\text{mult}} + B_{\text{ks}}) + B_{\text{round}}. \quad (2)
\]

For the parameter sets we consider, \(B_{\text{round}}\) dominates the error, where \(B_{\text{round}}\) has variance

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

4 Adversarial Model

Let us first examine the IND-CPA\(^D\) adversarial model introduced by Li and Micciancio [26]. In their setting, the adversary was a passive observer—as in the IND-CPA security game—but with the additional (limited) power to request decryptions of evaluations of honestly generated ciphertexts. For reasons of space, we present the definition in Appendix ??.

**Definition 4.1 (IND-CPA\(^D\) Security [26])**. Let \(E = (\text{KeyGen}, \text{Encrypt}, \text{Decrypt}, \text{Eval})\) be a public-key homomorphic, approximate encryption scheme with plaintext space \(\mathcal{M}\) and ciphertext space \(\mathcal{C}\). We define an experiment \(\text{Expr}_{b}^{\text{indcpa}^D}[\mathcal{A}]\), 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 \(\text{Encrypt}(pk, m_0, m_1)\) that, given a pair of plaintext messages \(m_0, m_1\), computes \(ct \leftarrow \text{Encrypt}(pk, m_0)\), 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 \(\text{Eval}(g, J)\) that, given a function \(g : \mathcal{M}^k \rightarrow \mathcal{M}\) and a sequence of indices \(J = \{j_1, \ldots, j_k\} \subseteq \{1, \ldots, |S|\}^k\), computes the ciphertext \(ct \leftarrow \text{Eval}(evk, g, S[j_1], ct, \ldots, S[j_k], ct)\), extends the state

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

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

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

The experiment is defined as

\[
\text{Expr}_{b}^{\text{indcpa}^D}[\mathcal{A}](\kappa) : (sk, pk, evk) \leftarrow \text{KeyGen}(\kappa) \\
S := [] \\
b' \leftarrow \mathcal{A}_{\text{Encrypt}(pk, \cdot), \text{Eval}(evk, \cdot), \text{Decrypt}(sk, \cdot)}(1^\kappa, pk, evk) \\
\text{return}(b')
\]

The advantage of adversary \(\mathcal{A}\) against the IND-CPA\(^D\) security of the scheme is

\[
\text{Adv}_{\text{indcpa}^D}[\mathcal{A}](\kappa) = |\text{Pr}[\text{Expr}_{0}^{\text{indcpa}^D}[\mathcal{A}](1^\kappa) = 1] - \text{Pr}[\text{Expr}_{1}^{\text{indcpa}^D}[\mathcal{A}](1^\kappa) = 1]|.
\]

In this work, we consider two adversarial models that are variants and/or special cases of the IND-CPA\(^D\) model presented in Definition 4.1.
The first adversarial model. We introduce a “white-box” variant of the encryption oracle, denoted Encrypt*(pk,·,·). When queried with two messages $m_0 = m_1$, this oracle returns a ciphertext ct, as well as the internal randomness $\epsilon, e_0$, and $e_1$ generated during the encryption process (see the definition of the encryption function in Section 3). If $m_0 \neq m_1$, it returns the ciphertext only. Other than this change, the adversarial model can be viewed as a special case of Li and Micciancio’s IND-CPA$^D$ in which the Encrypt* oracle is only called with $m_0 = m_1$ and the Eval oracle is called on the identity function only. Thus, the set $S$ consists only of fresh ciphertexts ct, and only those for which $m_0 = m_1$ may be queried to the decryption oracle. The decryption oracle we consider returns $\text{Decrypt}(sk, ct) + \mathcal{N}(0, \sigma^2_n)$, for some noise-flooding variance $\sigma^2_n$. The goal of our adversary will be full key recovery, at which point it can trivially break the IND-CPA$^D$ security by performing an encryption query with $m_0 \neq m_1$, obtaining ct, and using the recovered key to decrypt and find the value of $b$. In Section 4.1, we formalize the information the adversary observes as “hints” for this adversarial model.

The second adversarial model. This model is a special case of the IND-CPA$^D$ model. There is no “white-box” encryption oracle and the adversary is a legal IND-CPA$^D$ adversary. The Encrypt(pk,·,·) oracle is only called with $m_0 = m_1$ and the H$\text{er}_{xk}(\cdot,\cdot)$ oracle is only called with functions $g : \mathcal{M}^k \to \mathcal{M}$ in Class 1 or Class 2 and with input indices $J = (j_1, \ldots, j_k)$ that correspond to $k$ distinct, fresh ciphertexts outputted by calls to Encrypt(pk,m$_0$,m$_1$) with $m_0 = m_1$ and have not been included in a set $J$ in a previous call to H$\text{er}_{xk}(\cdot,\cdot)$. Decryption queries are only made with ciphertexts ct corresponding to the output of calls to H$\text{er}_{xk}(\cdot,\cdot)$ as described above. The decryption oracle we consider returns $\text{Decrypt}(sk, ct) + \mathcal{N}(0, \sigma^2_n)$, for some noise-flooding variance $\sigma^2_n$. As before, the goal of our adversarial model is full key recovery, at which point it can trivially break the IND-CPA$^D$ security by performing one more encryption query with $m_0 \neq m_1$. In Section 8, we extend our analysis from Section 4.1 to capture the “hints” obtained in this adversarial model.

4.1 Modeling Noisy Decryptions of Identity Circuit as Hints

We concretely consider an adversary who obtains $t$ independently sampled encryptions and then asks for $t$ decryptions of the constructed ciphertexts. Instantiating this attack with the CKKS + noise-flooding scheme, for each $j \in [t]$, the adversary obtains the (noisy) polynomial $e'_j \cdot s + v^j \cdot e \approx \gamma^j$, where multiplication is over the ring $R_q$. The adversary knows $e'_j$ and $v^j$ whose coordinates are modeled as independent Gaussians with 0 mean and variance $\sigma^2_{n,h_j}$ and $\sigma^2_{h_j}$, 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 $\mathbb{Z}_q[x]/\Phi_m(x)$, where $\Phi_m(x)$ is the $m$-th cyclotomic polynomial of degree $n = \phi(m)$, and $n$ is a power of two.

5 Security Loss under a Lattice Reduction Attack

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 $\sigma^2_n$, the second $n$ diagonal entries set to $\sigma^2$. The matrix $\Sigma_c$ corresponds to the covariance of the noise in the set of linear equations obtained on the LWE secret $s$ from decrypting a ciphertext. Formally, $\Sigma_c = \sigma^2_c \cdot I_n$. $\gamma = \gamma^1||\cdots||\gamma^t$ corresponds to the obtained outputs.

First, note that for $j \in [t]$, $e'_j \cdot s = s \text{VBP} \left(M(e'_j)\right) P^{-1} B^{-1} V^{-1},$ 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 $A^j_i := M(e'_j)$ is a block diagonal matrix with $n/2$ blocks, each of dimension $2 \times 2$, where the $i$-th block is

$$A^j_{i,i} := \begin{bmatrix} 1/\sqrt{2} w^j_{i,h_i} & 1/\sqrt{2} w^j_{n-i,h_i} \\ -1/\sqrt{2} w^j_{n-i,h_i} & 1/\sqrt{2} w^j_{i,h_i} \end{bmatrix},$$

and $w^j_{h_i} = (w^j_{1,h_i}, \ldots, w^j_{n,h_i})$ is equal to $w^j_{h_i} = e'_j \text{VB}$. Since $\text{VB}$ is an isometry (an orthogonal matrix scaled by $\sqrt{n}$), we have that $\sigma^2_{h_i}(\text{VB})(\text{VB})^T = n \sigma^2_{h_i} I_n$. So the random variables $[w^j_{1,h_i}, w^j_{n-i,h_i}]_{j \in [t], i \in [n/2]}$
are distributed as independent Gaussians with variance $n\sigma_{h_v}^2$. Note that $R = (VBP)$ is a real matrix, even though $V$ and $B$ themselves are complex.

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

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

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

$$A_{2,i}^j := \begin{bmatrix} 1/\sqrt{2}w_{i,h_v}^j & 1/\sqrt{2}w_{n-i,h_v}^j \\ -1/\sqrt{2}w_{n-i,h_v}^j & 1/\sqrt{2}w_{i,h_v}^j \end{bmatrix}$$ 

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

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

$$H = \begin{bmatrix} R & 0 \\ 0 & R \end{bmatrix} \begin{bmatrix} A_1^1 & A_2^1 & \cdots & A_1^t \\ A_1^1 & A_2^2 & \cdots & A_2^t \\ \vdots & \vdots & \ddots & \vdots \\ A_1^1 & A_2^t & \cdots & A_2^t \end{bmatrix} \begin{bmatrix} R^{-1} \\ & \ddots \rule{0pt}{2.5ex} \\ & & R^{-1} \end{bmatrix},$$

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

Applying the approximate hints of [17], the transformed covariance matrix $\Sigma'$ and mean $\mu'$ are as follows (the dimension and lattice of the DBDD instance remain unchanged):

$$\Sigma' = \Sigma - \Sigma H (H^T \Sigma H + \Sigma_e)^{-1} H^T \Sigma \quad (3)$$

$$\mu' = \gamma (H^T \Sigma H + \Sigma_e)^{-1} H^T \Sigma. \quad (4)$$

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 $H$.

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

$$\mathbb{E}[\det(\Sigma')] = \mathbb{E} \left[ \frac{\det(H^T \Sigma H + \Sigma_e)}{\det(\Sigma_e) \det(\Sigma)} \right], \quad (5)$$

Since $\Sigma_e$ 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 $\mathbb{E}[\det(H^T \Sigma H + \Sigma_e)]$, which can then be plugged into (5).

**Lemma 5.1.** Let $H, R, [A_1^j = M(e_1^j), A_2^j = M(v^j)]_{j \in [t]}$ be as described above. Then

$$\mathbb{E}[\det(H^T \Sigma H + \Sigma_e)] =$$

$$\left( \sigma^4 \sigma^4 \sigma^4 \frac{t^2}{4} \frac{1}{t-1} n^4 \sigma_{h_v}^4 \sigma_{h_v}^4 + t n^2 \sigma^4 \left( \frac{\sigma_{h_v}^2}{\sigma_{h_v}^2} + \frac{\sigma_{h_v}^2}{\sigma_{h_v}^2} \right) + \left( t(t-1)n^2 \sigma_{h_v}^4 \sigma_{h_v}^4 + t n^2 \sigma^4 \left( \frac{\sigma_{h_v}^2}{\sigma_{h_v}^2} + \frac{\sigma_{h_v}^2}{\sigma_{h_v}^2} \right) + \frac{\sigma^2}{\sigma^2} + \frac{\sigma^2}{\sigma^2} \right)^2 \right)^{\frac{1}{2}},$$

where the expectation is taken over choice of $e_1^t \sim N(0, \sigma_{h_v}^2 n)$ and $v^j \sim N(0, \sigma_{h_v}^2 n)$ for all $j \in [t]$.

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

$$\det \left( A + U V^T \right) = \det \left( I_m + V A^{-1} U \right) \det(A),$$

where $I_m$ is the $m$-by-$m$ identity matrix.
and the definition of $H$ and $\Sigma$ to rewrite $\det(H^T\Sigma H + \Sigma_\epsilon)$ as

$$
\det(H^T\Sigma H + \Sigma_\epsilon) = \det \left( I_{2n} + \frac{1}{\sigma^2} \Sigma^{1/2} H H^T \Sigma^{1/2} \right) \det(\Sigma_\epsilon)
$$

$$
= \det \left( I_{2n} + \frac{1}{\sigma^2} \left[ \begin{array}{cc} \sigma^2 \Sigma_{B,1,1} & \sigma \sigma_\epsilon \Sigma_{B,1,2} \\ \sigma \sigma_\epsilon \Sigma_{B,2,1} & \sigma^2 \Sigma_{B,2,2} \end{array} \right] \right) \det(\Sigma_\epsilon)
$$

$$
= \det \left( \begin{array}{cc}
\frac{\sigma^2}{\sigma^2} \sum_{j=1}^n \mathbf{A}_j^T (\mathbf{A}_j^T)^T + I_n & \frac{\sigma \sigma_\epsilon}{\sigma^2} \sum_{j=1}^n \mathbf{A}_j^T (\mathbf{A}_j^T)^T + I_n \\
\frac{\sigma \sigma_\epsilon}{\sigma^2} \sum_{j=1}^n \mathbf{A}_j^T (\mathbf{A}_j^T)^T & \frac{\sigma^2}{\sigma^2} \sum_{j=1}^n \mathbf{A}_j^T (\mathbf{A}_j^T)^T + I_n
\end{array} \right) \det(\Sigma_\epsilon)
$$

$$
= \det(\ast) \det(\Sigma_\epsilon),
$$

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

$$
\det(\ast) \det(\Sigma_\epsilon) = \det \left[ \begin{array}{ccc}
\mathbf{S}_1 & 0 & \ldots & 0 & 0 \\
0 & \mathbf{S}_2 & \ldots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \ldots & \mathbf{S}_{2-1} & 0 \\
0 & 0 & \ldots & 0 & \mathbf{S}_2
\end{array} \right] = \det(\Sigma_\epsilon) = \det(\circ)
$$

where

$$
\det(\circ) \det(\Sigma_\epsilon) = \det \left[ \begin{array}{cc}
a_i & 0 \\
0 & \mathbf{S}_i
\end{array} \right] = \det \left[ \begin{array}{cc}
A & B \\
C & D
\end{array} \right] = \det(AD - BC),
$$

(6)

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

$$
a_i = \frac{\sigma^2}{2\sigma^2} \left( \sum_{j=1}^n \left( w_{j,h_s}^j + w_{j,n-i,h_s}^j \right)^2 + \frac{2\sigma^2}{\sigma^2} \right),
$$

$$
b_i = \frac{\sigma^2}{2\sigma^2} \left( \sum_{j=1}^n \left( w_{j,h_s}^j + w_{j,n-i,h_s}^j \right)^2 + \frac{2\sigma^2}{\sigma^2} \right),
$$

$$
c_i = \frac{\sigma \sigma_\epsilon}{2\sigma^2} \sum_{j=1}^n \left( w_{i,h_s}^j w_{i,h_s}^j + w_{j,n-i,h_s}^j w_{j,n-i,h_s}^j \right),
$$

$$
d_i = \frac{\sigma \sigma_\epsilon}{2\sigma^2} \sum_{j=1}^n \left( w_{i,h_s}^j w_{j,n-i,h_s}^j - w_{j,i,h_s}^j w_{j,n-i,h_s}^j \right).
$$

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

$$
\det \left( \begin{array}{cc}
A & B \\
C & D
\end{array} \right) = \det(AD - BC).
$$
We therefore have that

$$\det(\odot) = \det(\Sigma) \prod_{i=1}^{n/2} \det(S_i)$$

$$= \sigma_e^{2n} \prod_{i=1}^{n/2} \frac{\sigma_h^2 \sigma_e^2}{16 \sigma_e^4} \left( \left( \sum_{j=1}^{t} (w_{i,h_i}^j)^2 + (w_{n-i,h_i}^j)^2 \right)^2 \frac{2 \sigma_e^2}{\sigma_h^2} \left( \sum_{j=1}^{t} (w_{i,h_i}^j)^2 + (w_{n-i,h_i}^j)^2 \right)^2 \frac{2 \sigma_e^2}{\sigma_h^2} \right)$$

$$- \left( \sum_{j=1}^{t} (w_{i,h_i}^j w_{i,h_i}^j + w_{n-i,h_i}^j w_{n-i,h_i}^j) \right)^2 - \left( \sum_{j=1}^{t} (w_{i,h_i}^j w_{n-i,h_i}^j - w_{i,h_i}^j w_{n-i,h_i}^j) \right)^2$$

$$= \left( \frac{\sigma_h^2 \sigma_e^2}{16 \sigma_e^2} \right)^2 \prod_{i=1}^{t} \left( \sum_{1 \leq j < k \leq t} \left( w_{i,h_i}^j w_{i,h_i}^k + w_{n-i,h_i}^j w_{n-i,h_i}^k \right)^2 + \left( w_{n-i,h_i}^j w_{i,h_i}^k + w_{n-i,h_i}^j w_{n-i,h_i}^k \right)^2 \right)$$

$$+ \left( \frac{2 \sigma_e^2}{\sigma_h^2} \right)^2 \sum_{j=1}^{t} \left( (w_{i,h_i}^j)^2 + (w_{n-i,h_i}^j)^2 \right) + \left( \frac{2 \sigma_e^2}{\sigma_h^2} \right)^2 \sum_{j=1}^{t} \left( (w_{i,h_i}^j)^2 + (w_{n-i,h_i}^j)^2 \right) + \frac{4 \sigma_e^4}{\sigma_h^4 \sigma_e^2}.$$

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

$$Y_i := \sum_{1 \leq j < k \leq t} \left( \left( w_{i,h_i}^j w_{i,h_i}^k \right)^2 + \left( w_{i,h_i}^j w_{n-i,h_i}^k \right)^2 + \left( w_{n-i,h_i}^j w_{i,h_i}^k \right)^2 + \left( w_{n-i,h_i}^j w_{n-i,h_i}^k \right)^2 \right)$$

$$- \left( w_{i,h_i}^j w_{i,h_i}^j w_{i,h_i}^k w_{i,h_i}^k + w_{n-i,h_i}^j w_{n-i,h_i}^j w_{n-i,h_i}^k w_{n-i,h_i}^k \right)$$

$$+ \left( w_{i,h_i}^j w_{n-i,h_i}^j w_{i,h_i}^k w_{n-i,h_i}^k + w_{n-i,h_i}^j w_{n-i,h_i}^j w_{n-i,h_i}^k w_{n-i,h_i}^k \right)$$

$$+ \frac{2 \sigma_e^2}{\sigma_h^2} \sum_{j=1}^{t} \left( (w_{i,h_i}^j)^2 + (w_{n-i,h_i}^j)^2 \right) + \frac{2 \sigma_e^2}{\sigma_h^2} \sum_{j=1}^{t} \left( (w_{i,h_i}^j)^2 + (w_{n-i,h_i}^j)^2 \right) + \frac{4 \sigma_e^4}{\sigma_h^4 \sigma_e^2}.$$

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

$$\mathbb{E}[\det \left( H^T \Sigma H + \Sigma_e \right)] = \left( \frac{\sigma_h^4 \sigma_e^4}{16 \sigma_e^4} \right)^2 \prod_{i=1}^{t} \mathbb{E} = \left( \frac{\sigma_h^4 \sigma_e^4}{16 \sigma_e^4} \right)^2 \prod_{i=1}^{t} \left( \text{Var}Y + \mathbb{E}^2Y \right),$$

where

$$\text{Var}Y = 28t(t-1)n^4 \sigma_h^4 \sigma_e^4 + 16t n^2 \sigma_e^4 \left( \frac{\sigma_h^4}{\sigma_e^4} \right),$$

$$\mathbb{E}^2Y = \left( 4t(t-1)n^2 \sigma_h^2 \sigma_e^2 + 4tn^2 \left( \frac{\sigma_h^2}{\sigma_e^2} \right) + 4 \frac{\sigma_e^4}{\sigma_h^4} \right)^2.$$
Finally, we obtain that
\[
\mathbb{E}[\det(H^T \Sigma H + \Sigma_v)] = \\
\left(\sigma_h^4 \sigma_e^2 \sum_{t=1}^{7} t(t-1)n^4 \sigma_h^4 \sigma_e^4 + t^2 \sigma_h^4 \left(\frac{\sigma_h^4}{\sigma_e^2} + \frac{\sigma_h^4}{\sigma_e^2} + \frac{\sigma_h^4}{\sigma_e^2} \right) \right) \cdot \left(\frac{7}{4} \frac{t(t-1)n^4 \sigma_h^4 \sigma_e^4 + t^2 \sigma_h^4}{\sigma_h^4 \sigma_e^2 + \frac{\sigma_h^4}{\sigma_e^2} + \frac{\sigma_h^4}{\sigma_e^2} \sigma_e^2} \right) + \left(t^2 \sigma_h^4 \left(\frac{\sigma_h^4}{\sigma_e^2} + \frac{\sigma_h^4}{\sigma_e^2} \right) \right) \cdot \left(\frac{7}{4} \frac{t(t-1)n^4 \sigma_h^4 \sigma_e^4 + t^2 \sigma_h^4}{\sigma_h^4 \sigma_e^2 + \frac{\sigma_h^4}{\sigma_e^2} + \frac{\sigma_h^4}{\sigma_e^2} \sigma_e^2} \right). \\
\]
Proof. Recall that

\[ \Sigma' = \Sigma - \Sigma \Sigma H^T \Sigma H + \Sigma \epsilon \Sigma = \Sigma H^T \Sigma H \epsilon \Sigma \epsilon H^T \Sigma. \]

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

Using the generalization of the matrix determinant lemma, this is the same as finding \( \alpha \) such that \( \det(\Sigma \epsilon + \Sigma H^T \Sigma \epsilon H) = 0 \).

Let \( \Sigma = \Sigma - (\Sigma' - \alpha \cdot I)^{-1} \). Then we must find \( \alpha \) such that \( \det(\Sigma \epsilon + \Sigma H^T \Sigma \epsilon H) = 0 \). Then \( \Sigma \) is a diagonal matrix with entries \( \frac{-\sigma^2_s \cdot \epsilon}{\sigma^2_s - \epsilon} \) in the first \( n \) positions and entries \( \frac{-\sigma^2_s \cdot \epsilon}{\sigma^2_s - \epsilon} \) in the last \( n \) positions. Using the analysis from the proof of Lemma 5.1, we have that

\[ \det(\Sigma \epsilon + \Sigma H^T \Sigma \epsilon H) = \Pi_{i \in [n/2]}(a_i b_i - c_i^2 - d_i^2), \quad (7) \]

where

\[
\begin{align*}
a_i &= \frac{-\sigma^2_s \cdot \alpha}{2(\sigma^2_s - \alpha)\sigma_e} R_{1,i} + 1, \\
b_i &= \frac{-\sigma^2_s \cdot \alpha}{2(\sigma^2_e - \alpha)\sigma_e} R_{2,i} + 1, \\
c_i &= \frac{-\sigma_s \cdot \sigma_3 \cdot \alpha}{2\sigma^2_s - \alpha} \sqrt{\sigma^2_e - \alpha} R_{3,i}, \\
d_i &= \frac{-\sigma_s \cdot \sigma_3 \cdot \alpha}{2\sigma^2_e - \alpha} \sqrt{\sigma^2_e - \alpha} R_{4,i}
\end{align*}
\]

and

\[
\begin{align*}
R_{1,i} &= \sum_{j=1}^{t} (W_{j,i,h_s}^j)^2 + (W_{n-i,h_s}^j)^2, \\
R_{2,i} &= \sum_{j=1}^{t} (W_{j,i,h_s}^j)^2 + (W_{n-j,i,h_s}^j)^2, \\
R_{3,i} &= \sum_{j=1}^{t} W_{j,i,h_s}^j W_{n-j,i,h_s}^j W_{n-j,i,h_s}^j, \\
R_{4,i} &= \sum_{j=1}^{t} W_{j,i,h_s}^j W_{n-j,i,h_s}^j - W_{j,i,h_s}^j W_{n-j,i,h_s}^j.
\end{align*}
\]

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 \( a_i b_i - c_i^2 - d_i^2 = 0 \). \( \sum_{j \in [4]} \alpha_{4i+j} \) is then equal to the sum of those roots, \( -\frac{q_{a,i}}{q_{a,i}} = -\frac{2q_{b,i}}{q_{a,i}} \),

where

\[
\begin{align*}
q_{a,i} &= \frac{\sigma^2_s \cdot \sigma_e \cdot R_{1,i} + R_{2,i}^2}{4 \cdot \sigma^4_s} + \frac{\sigma^2_s \cdot R_{1,i} + R_{2,i}^2}{4 \cdot \sigma^4_e} + 1 - \frac{\sigma^2_s \cdot \sigma^2_e \cdot R_{3,i}^2}{4 \cdot \sigma^4_e} - \frac{\sigma^2_s \cdot \sigma^2_e \cdot R_{4,i}^2}{4 \cdot \sigma^4_e}, \\
q_{b,i} &= -\left( \frac{\sigma^2_s \cdot R_{1,i}}{2 \cdot \sigma^4_s} + \frac{\sigma^2_e \cdot R_{2,i}}{2 \cdot \sigma^4_e} + \sigma^2_s + \sigma^2_e \right).
\end{align*}
\]

Towards bounding \( E[-\frac{q_{b,i}}{q_{a,i}}] \), 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

\[
R_{3,i} = \sum_{j=1}^{2t} 1/4(X_j^t)^2 + \sum_{j=1}^{2t} 1/4(X_j^t)^2
\]

16
where \(X'_j\) and \(X''_j\) are a Gaussian random variable with variance \(n(\sigma^2_{\Delta^2} + \sigma^2_{\Delta^2})\), \(X'_1, \ldots, X'_{2t}\) are independent and \(X''_1, \ldots, X''_{2t}\) are independent. The probability that either \(1/4 \sum_{j=1}^{2t} (X'_j)^2 \neq 2t\cdot n(\sigma^2_{\Delta^2} + \sigma^2_{\Delta^2}) \pm 2n(\sigma^2_{\Delta^2} + \sigma^2_{\Delta^2})(3.5\sqrt{2\sigma^2} + 12.25)\) or \(1/4 \sum_{j=1}^{2t} (X''_j)^2 \neq 2t\cdot n(\sigma^2_{\Delta^2} + \sigma^2_{\Delta^2}) \pm 2n(\sigma^2_{\Delta^2} + \sigma^2_{\Delta^2})(3.5\sqrt{2\sigma^2} + 12.25)\) is at most \(2e^{-12.25}\).

Thus, \(R_{3,i}\) and \(R_{4,i}\) are both in \([-n(\sigma^2_{\Delta^2} + \sigma^2_{\Delta^2})(5\sqrt{2\sigma^2} + 25), n(\sigma^2_{\Delta^2} + \sigma^2_{\Delta^2})(3.5\sqrt{2\sigma^2} + 12.25)]\) with probability at least \(4e^{-12.25}\).

Further, \(R_{3,i}^2\) (and similarly \(R_{4,i}^2\)) is at most \(n^2(\sigma^2_{\Delta^2} + \sigma^2_{\Delta^2})(3.5\sqrt{2\sigma^2} + 12.25)^2\).

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

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

\[
q_{a,i} = \frac{\sigma^2 \cdot \sigma^2 \cdot R_{1,i} \cdot R_{2,i}}{4 \cdot \sigma^2} + \frac{\sigma^2 \cdot R_{1,i}}{2 \cdot \sigma^2} + \frac{\sigma^2 \cdot R_{2,i}}{2 \cdot \sigma^2} + 1
\]

\[
\geq \frac{\sigma^2 \cdot \sigma^2 \cdot (2t - 7\sqrt{2t})^2 \cdot n^2 \sigma^2_{\Delta^2} \cdot \sigma^2_{\Delta^2}}{4 \cdot \sigma^2} - \frac{\sigma^2 \cdot \sigma^2 \cdot R_{3,i}}{4 \cdot \sigma^2} - \frac{\sigma^2 \cdot \sigma^2 \cdot R_{4,i}}{4 \cdot \sigma^2} + \frac{\sigma^2 \cdot \sigma^2 \cdot (2t - 7\sqrt{2t}) \cdot n\sigma^2_{\Delta^2}}{2 \cdot \sigma^2} + \frac{\sigma^2 \cdot \sigma^2 \cdot \sigma^2 \cdot n\sigma^2_{\Delta^2} + \sigma^2_{\Delta^2}}{2 \cdot \sigma^2} - 1
\]

\[
= B.
\]

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

\[
\mathbb{E}\left[\frac{-2q_{b,i}}{q_{a,i}}\right] \leq \frac{\mathbb{E}[-2q_{b,i}]}{B} = \frac{2\left(\frac{\sigma^2 \cdot \sigma^2 \cdot \mathbb{E}[R_{1,i}]}{2 \cdot \sigma^2} + \frac{\sigma^2 \cdot \sigma^2 \cdot \mathbb{E}[R_{2,i}]}{2 \cdot \sigma^2} + \sigma^2 + \sigma^2\right)}{B} = 2\left(\frac{\sigma^2 \cdot \sigma^2 \cdot 2t \cdot n\sigma^2_{\Delta^2}}{2 \cdot \sigma^2} + \frac{\sigma^2 \cdot \sigma^2 \cdot 2t \cdot n\sigma^2_{\Delta^2}}{2 \cdot \sigma^2} + \sigma^2 + \sigma^2\right)\]

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. \(\mathbb{E}[R_{2,i}], \mathbb{E}[R_{1,i}^2], \mathbb{E}[R_{2,i}^2]\)) is the same for all \(i \in [n/2]\). Therefore, we denote \(\mathbb{E}[R_{1}] = \mathbb{E}[R_{1,i}]\) (resp. \(\mathbb{E}[R_{2}] = \mathbb{E}[R_{2,i}], \mathbb{E}[R_{2,i}^2] = \mathbb{E}[R_{2,i}^2], \mathbb{E}[R_{2,i}^2] = \mathbb{E}[R_{2,i}^2]\)). We have:

\[
\mathbb{E}[R_{1}] = 2t \cdot n\sigma^2_{\Delta^2},
\mathbb{E}[R_{2}] = 2t \cdot n\sigma^2_{\Delta^2},
\mathbb{E}[R_{1}^2] = 4t^2 \sigma^4_{\Delta^2} + 4t^2 n^2 \sigma^4_{\Delta^2},
\mathbb{E}[R_{2}^2] = 4t^2 \sigma^4_{\Delta^2} + 4t^2 \sigma^4_{\Delta^2}.
\]

Further, \(R_{1}\) and \(R_{2}\) are independent.

\[
V = \mathbb{E}[(q_{b})^2] - \mathbb{E}[q_{b}]^2
\]

\[
= \frac{\sigma^4 \cdot \sigma^4 \cdot \mathbb{E}[R_{1}^2] + \mathbb{E}[R_{2}^2]}{4 \cdot \sigma^4} + \frac{2 \sigma^4 \cdot \sigma^4 \cdot \mathbb{E}[R_{1}] \cdot \mathbb{E}[R_{2}]}{4 \cdot \sigma^4} + \frac{2 \sigma^4 \cdot \sigma^4 \cdot \mathbb{E}[R_{1}]}{2 \cdot \sigma^4} \cdot \frac{\mathbb{E}[R_{2}]}{2 \cdot \sigma^4} + \frac{\sigma^4 \cdot \sigma^4 \cdot \mathbb{E}[R_{1}]}{2 \cdot \sigma^4}\]

\[
- \frac{\sigma^4 \cdot \sigma^4}{2 \cdot \sigma^4} \cdot \mathbb{E}[R_{1}] + \frac{\sigma^4 \cdot \sigma^4}{2 \cdot \sigma^4} \cdot \mathbb{E}[R_{2}] + \sigma^2 + \sigma^2\]
Using Chebyshev, we therefore have that $\Pr[|E[\mathbf{Tr}(\Sigma')] - \mathbf{Tr}(\Sigma')| > 0.01]\leq 0.01.$ Thus, putting everything together, we have that with $0.99 - 3n \cdot e^{-12.25}$ probability,
\[\mathbf{Tr}(\Sigma') \leq n \cdot \left( \frac{\sigma_2^2 \cdot 2n \cdot (\sigma_1^2 + \sigma_2^2)}{2 \cdot \sigma_2^2} + \sigma_2^2 \right) + 10\sqrt{2n \cdot \mathbf{V}}.\]

Given the above, we consider the distribution of $e||s - \mu'$, where $\mu'$ is the mean from equation (4). The random variable $e||s - \mu'$ is distributed as the multivariate Gaussian distribution $\mathcal{N}(0, \Sigma')$. $\mu'$ is the correct guess for $e||s$ 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 $\mathcal{N}(0, \Sigma')$. We use the following theorem to lower bound this probability weight:

**Theorem 6.2 (Special case of the Gaussian Correlation Inequality [25]).** Let $X$ be an $n$-dimensional Gaussian random variable. Then for any $t_1, \ldots, t_n > 0$,
\[
\mathbb{P}(|X_1| \leq t_1, \ldots, X_n \leq t_n) \geq \mathbb{P}(|X_1| \leq t_1) \cdots \mathbb{P}(|X_n| \leq t_n).
\]

We instantiate the above theorem with $X$ consisting of a subset $S$ of size $n$ of the coordinates of the conditional Gaussian distribution ((s||e) - $\mu') \sim \mathcal{N}(0, \Sigma')$, with $t_j = 0.5, j \in S$ We thus have that
\[
\mathbb{P}(|X_j| \leq t_j, i \in S) \geq \prod_{j \in S} \mathbb{P}_{X_j \sim \mathcal{N}(0, e_j \Sigma' e_j^T)}(|X_j| \leq t_j),
\]
where the $e_j$ are the standard basis vectors.

To analyze $\Pr_{X_j \sim \mathcal{N}(0, e_j \Sigma' e_j^T)}(|X_j| \leq 0.5)$, we note that $\sum_{i \in [2n]} e_i \cdot e_i^T = \mathbf{Tr}(\Sigma')$. By Lemma 6.1, we have that $\mathbf{Tr}(\Sigma') \leq 1$ with $53\%$ probability. Thus, the indices $j$ corresponding to the $n$ smallest values among $\{e_i \cdot e_i^T : i \in [2n]\}$ have sum at most $\frac{T}{2}$, and average $\frac{T}{2n}$, which we use in our estimates. Let $S \subseteq [2n]$ of size $n$ be this set of minimum values. For each $j \in S$,
\[
\mathbb{P}_{X_j \sim \mathcal{N}(0, e_j \Sigma' e_j^T)}(|X_j| \leq 0.5) \geq -\text{erf} \left( \frac{-0.5}{\sqrt{2 \cdot \frac{T}{2n}}} \right).
\]

Finally, the attack is as follows: The adversary chooses to guess the values of $e_j$ or $s_j$ 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
\[
\prod_{j \in S} -\text{erf} \left( \frac{-0.5}{\sqrt{2 \cdot \frac{T}{2n}}} \right) \geq -\text{erf} \left( \frac{-0.5}{\sqrt{2 \cdot \frac{T}{2n}}} \right)^n = -\text{erf} \left( \frac{-0.5}{\sqrt{\frac{T}{n}}} \right)^n.
\]

The final success probability of the attack is:
\[
-\text{erf} \left( \frac{-0.5}{\sqrt{\frac{T}{n}}} \right)^n - 3n \cdot e^{-12.25} - 0.01.
\]

---

6 A more rigorous but looser analysis can be achieved by upperbounding the largest of the $n$ smallest values by $\frac{T}{n}$.  
7 And for $n = 131072$, we replace $e^{-12.25}$ with $e^{-14}$.  

18
7 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 $e_i \Sigma' e_i^T$ and guesses $\mu_i$ for the $g$ minimum values where $g$ is the maximum value such that

$$ \text{erf} \left( \frac{-0.5}{\sqrt{\frac{1}{n}}} \right) \geq p, $$

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 $\text{PSub}_{2n-g}(\Sigma')$ 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 7.1. Let $g \in \{0, 1, \ldots, n\}$. Let $\Sigma'$ be defined as in (3). Let $\Sigma'' = \arg\max_{\Sigma \in \text{PSub}_{2n-g}(\Sigma') \cap \Sigma'_{\text{det}}} \text{Tr}(\Sigma')$. With probability $0.99 - 4n \cdot e^{-12.25}$ over choice of hint vectors,$^8$

$$ \text{Tr}(\Sigma') \leq T \quad \text{and} \quad \det(\Sigma'') \leq \frac{\det(\Sigma')}{(\frac{T}{B})^g}, $$

where $T$ and $B$ are defined as in Lemma 6.1, and

$$ L = \frac{G + \sqrt{G^2 - 4 \cdot B \cdot \sigma_s^2 \cdot \sigma_c^2}}{2 \cdot B}, $$

$$ U = \frac{\sigma_s^2 \cdot \sigma_c^2}{B_{\text{max}}}, $$

$$ G = \sigma_s^2 (2t + 7 \sqrt{2t} + 24.5) \cdot (n \sigma_h^2) \cdot 2 \cdot \sigma_s^2 + \sigma_c^2 (2t + 7 \sqrt{2t} + 24.5) \cdot (n \sigma_h^2) \cdot 2 \cdot \sigma_c^2 + \sigma_s^2 + \sigma_c^2, $$

$$ B_{\text{max}} = \frac{\sigma_s^2 \cdot \sigma_c^2 \cdot (2t + 7 \sqrt{2t} + 24.5)^2 \cdot n \sigma_h^2 \cdot \sigma_h^2 + \sigma_c^2 (2t + 7 \sqrt{2t} + 24.5) \cdot n \sigma_h^2}{4 \cdot \sigma_c^2 + \sigma_s^2 (2t + 7 \sqrt{2t} + 24.5) \cdot (n \sigma_h^2) + 1}. $$

Proof. Lemma 6.1 showed that with probability $0.99 - 3n \cdot e^{-12.25}$ over choice of hint vectors, $\text{Tr}(\Sigma') \leq T$.

Let $\alpha_1, \ldots, \alpha_g$ be the $g$ minimum eigenvalues of $\Sigma'$. Using the Eigenvalue Interlacing Theorem [23], we have that $\det(\Sigma'') \leq \frac{\det(\Sigma')}{\prod_{i=1}^{g} \alpha_i}$. We therefore need a lower bound on $\alpha_1 \cdots \alpha_g$.

We consider $\alpha_i' \cdots \alpha_g'$ 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 $\alpha_i' \geq \frac{L}{\alpha_i}$ for all $i \in [g]$. $\alpha_1 \cdots \alpha_g$ can then be lower bounded by $(\frac{T}{B})^g$, which implies that

$$ \det(\Sigma'') \leq \frac{\det(\Sigma')}{(\frac{T}{B})^g}. $$

Specifically, assuming the bounds from the proof of Lemma 6.1, 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,$^9$

$$ R_{1,i} \leq (2t + 7 \sqrt{2t} + 24.5) \cdot n \sigma_h^2, \quad R_{2,i} \leq (2t + 7 \sqrt{2t} + 24.5) \cdot n \sigma_h^2, $$

for the parameter sets with $n = 131072$, we increase $7$ to $7.5$, $24.5$ to $28.125$ and increase the probability to $0.99 - 4n \cdot e^{-14}$.

$^8$ For the parameter sets with $n = 131072$, we increase $7$ to $7.5$, $24.5$ to $28.125$ and increase the probability to $0.99 - 4n \cdot e^{-14}$.

$^9$ For the parameter sets with $n = 131072$, we increase $7$ to $7.5$, $24.5$ to $28.125$ and increase the probability to $1 - 2 \cdot e^{-14}$. 
we have that
\[
-q_b, i = \sigma_s^2 \cdot \sigma_e^2 \cdot (2t + 7\sqrt{2}t + 24.5) \cdot (n\sigma_h^2) 2 \cdot \sigma_e^2 \\
+ \sigma_e^2 \cdot \sigma_s^2 \cdot (2t + 7\sqrt{2}t + 24.5) \cdot (n\sigma_h^2) 2 \cdot \sigma_s^2 + \sigma_e^2 + \sigma_s^2 \\
= G.
\]

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

Using the same upper bounds from (12) we also have the following upper bound on \(q_{a,i}^{10}:
\[
q_{a,i} = \frac{\sigma_s^2 \cdot \sigma_e^2 \cdot R_{1,i} \cdot R_{2,i}}{4 \cdot \sigma_s^2} + \frac{\sigma_s^2 \cdot R_{1,i} \cdot R_{2,i}}{2 \cdot \sigma_s^2} + \frac{\sigma_e^2 \cdot R_{2,i}}{2 \cdot \sigma_e^2} + 1 - \frac{\sigma_s^2 \cdot \sigma_e^2 \cdot R_{3,i}}{4 \cdot \sigma_s^2} - \frac{\sigma_s^2 \cdot \sigma_e^2 \cdot R_{4,i}}{4 \cdot \sigma_s^2}
\leq \frac{\sigma_s^2 \cdot \sigma_e^2 \cdot (2t + 7\sqrt{2}t + 24.5)^2 \cdot n^2 \sigma_h^2 \cdot \sigma_s^2 \cdot \sigma_h^2}{2 \cdot \sigma_s^2} + \frac{\sigma_e^2 \cdot (2t + 7\sqrt{2}t + 24.5) \cdot n \sigma_h^2}{2 \cdot \sigma_e^2} + 1
\]
\[
= B_{\text{max}}.
\]

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

Combining Lemma 6.1 with Theorem 6.2 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
\[
\det(\Sigma'') \leq \frac{\det(\Sigma')}{(\frac{1}{p})^g}.
\]

We note that for up to \(n = 32768, 4n \cdot e^{-12.25} \leq 0.63, \) As before, \(E[\det(\Sigma')]\) can be computed via Lemma 5.1. 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 4.1, and use the log-volume and dimension to estimate the hardness of the residual instance (after guesses) under a lattice reduction attack.
8 Extending to Larger Classes of Circuits

8.1 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 $\delta_i^L \cdot s + \delta_i^R$ (see Section 3.5.2), where $\delta_i^L, \delta_i^R$ 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 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 $\frac{\ell}{12}$, which is the setting for which our Lemma 5.1 applies.

Specifically, the lattice reduction attack for circuits of this class can be analyzed by instantiating Lemma 5.1 with the following parameter settings.

- $\sigma_{b+}^2 = \frac{\ell}{12}$
- $\sigma_{b-}^2 = 0$
- $\sigma_{e}^2$ is set to the noise-flooding noise. The variance of the noise already present in the ciphertext can be computed by taking the noise in each ciphertext before addition (which Section 3.5.2 provides) and multiplying by $\ell$.

8.2 The second class of circuits and lattice reduction attacks

In Figure 1b we present the second class of circuits we consider. The circuits $C^L_1, \ldots, C^L_\ell, C^R_1, \ldots, C^R_\ell$ 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 $C^L_1, \ldots, C^L_\ell, C^R_1, \ldots, C^R_\ell$ is a multiplication with rescale. Note that the noise after multiplication with rescale in circuit $C^L_i$ (resp. $C^R_i$) is dominated by $\delta_i^{L,i} \cdot s + \delta_i^{R,i}$ (resp. $\delta_i^{R,i} \cdot s + \delta_i^{R,i}$) (see Section 3.5.2), where $\delta_i^{L,i}, \delta_i^{R,i}$ (resp. $\delta_i^{R,i}, \delta_i^{R,i}$) 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 $\frac{m}{2r-1} \cdot (G_{L,1} + G_{R,1}) \cdot s$. Further, the dominating linear coefficients of $s$ are again (well approximated by) a Gaussian of variance $\sigma_{e}^2 = \frac{\ell}{5} \cdot (\frac{m'}{2r-1})^2$. Since the error term does not include information about $e$, we can set $\sigma_{e}^2 = 0$.

For the parameter sets with $n = 131072$, we increase 7 to 7.5, and 24.5 to 28.

And for $n = 131072$, $4n \cdot e^{-14} \leq 0.44$. 

---

Fig. 1: A pictorial representation of the two classes of circuits we consider.
We compute the noise variance that is already present in the ciphertext, as a contribution of the following terms \( \frac{m^r}{\sqrt{n}} \cdot (G_{L,0} + G_{R,0}) \), \( \frac{m^r}{\sqrt{n}} \cdot (G_{L,1} + G_{R,1}) \cdot s \), \( (G_{L,1} \cdot G_{R,1}) \cdot s^2 \), \( (G_{L,0} \cdot G_{R,1}) \cdot s \), \( (G_{L,1} \cdot G_{R,0}) \cdot s \), \( G_{L,0} \cdot G_{R,0} \). 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 \( \ell \).

Since the arithmetic is coordinate-wise on the canonical space, it suffices to consider the arithmetic of \( i \)-th component of each vector in \( \mathbb{C}^{2^n} \). 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 \( \frac{m^r}{\sqrt{n}} \cdot (G_{L,0} + G_{R,0}) \).** The variance is immediately computed as \( 2(\frac{m^r}{\sqrt{n}})^2 \cdot \ell \cdot \sigma_s^2 \).

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

\[
\text{Var}[G_{L,1,i} \cdot \text{Re} s_i, \text{Re} - G_{L,0,i} \cdot \text{Im} s_i, \text{Im}]
= 2 \cdot \frac{n}{2} \cdot \frac{\ell}{12} \cdot \frac{n \cdot \sigma_s^4}{2}
\]

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

**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.

\[
\text{Var}
\left[
(G_{L,1,i} \cdot \text{Re} G_{R,1,i} \cdot \text{Re} - G_{L,1,i} \cdot \text{Im} G_{R,1,i} \cdot \text{Im})
(s_i^{2, \text{Re}} - s_i^{2, \text{Im}})
- 2 s_i^{\text{Re} \cdot \text{Im}} (G_{L,1,i} \cdot \text{Re} G_{R,1,i} \cdot \text{Im} + G_{L,1,i} \cdot \text{Im} G_{R,1,i} \cdot \text{Re})
\right]
= \text{Var}\left[
G_{L,1,i} \cdot \text{Re} G_{R,1,i} \cdot \text{Re} s_i^{2, \text{Re}} - G_{L,1,i} \cdot \text{Re} G_{R,1,i} \cdot \text{Re} s_i^{2, \text{Im}} - G_{L,1,i} \cdot \text{Im} G_{R,1,i} \cdot \text{Re} s_i^{2, \text{Re}} + G_{L,1,i} \cdot \text{Im} G_{R,1,i} \cdot \text{Im} s_i^{2, \text{Re}}
+ 2 s_i^{\text{Re} \cdot \text{Im}} (G_{L,1,i} \cdot \text{Re} G_{R,1,i} \cdot \text{Im} G_{R,1,i} \cdot \text{Re} - G_{L,1,i} \cdot \text{Im} G_{R,1,i} \cdot \text{Re} s_i^{2, \text{Re}} + G_{L,1,i} \cdot \text{Im} G_{R,1,i} \cdot \text{Im} s_i^{2, \text{Re}})
\right]
= 4 \text{E}\left[G_{L,1,i}^{2, \text{Re}} (G_{R,1,i}^{2, \text{Re}})^2 s_i^{2, \text{Re}}\right] + 8 \text{E}\left[(s_i^{2, \text{Re}})^2 G_{L,1,i}^{2, \text{Im}} (G_{R,1,i}^{2, \text{Im}})^2\right]
= 4 \left(\frac{n}{2} \cdot \frac{\ell}{12}\right)^2 \cdot \frac{3n^2}{4} \sigma_s^4 + 8 \left(\frac{n}{2} \cdot \frac{\sigma_s^4}{2}\right)^2 \left(\frac{n}{2} \cdot \frac{\ell}{12}\right)^2
= \frac{5n^4}{4} \left(\frac{\ell}{12}\right)^2 \sigma_s^4
\]

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

**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.

\[
\text{Var}
\left[
G_{L,0,i} \cdot \text{Re} G_{R,1,i} \cdot \text{Re} s_i^{\text{Re}} - G_{L,0,i} \cdot \text{Im} G_{R,1,i} \cdot \text{Im} s_i^{\text{Re}}
- G_{L,0,i} \cdot \text{Im} G_{R,1,i} \cdot \text{Re} s_i^{\text{Im}} - G_{L,0,i} \cdot \text{Re} G_{R,1,i} \cdot \text{Im} s_i^{\text{Im}}
\right]
= 4 \cdot \left(\frac{n}{2} \cdot \frac{\ell}{12}\right)^2 \cdot \frac{n \sigma_s^2}{2}
\]

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

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

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:

\[
2 \left( \frac{m^r}{\Delta^{-1}} \right)^2 \cdot \frac{\ell}{12} \cdot \sigma_s^2 + \left( \frac{m^s}{\Delta^{-1}} \right)^2 \cdot \frac{n \cdot \ell}{6} \cdot \sigma_s^2 + 5 \cdot \frac{\sigma_s^4}{2} + 2 \cdot \frac{n^2}{24} + \frac{\ell^2}{12} \cdot \sigma_s^2 + n \cdot \left( \frac{\ell}{12} \right)^2
\]

**Obtaining the hardness estimates.** We can now apply Lemma 5.1 with the following parameter settings:

- \( \sigma_{y,s} = \frac{\ell}{6} \cdot \left( \frac{m^r}{\Delta^{-1}} \right)^2 \)
- \( \sigma_{h,s} = 0 \)
- \( \sigma_s^2 \) is set to the noise-flooding noise plus an additional \( \frac{5}{2} \cdot n^3 \cdot \left( \frac{\ell}{12} \right)^2 \cdot \sigma_s^4 + 2 \cdot n^2 \cdot \left( \frac{\ell}{12} \right)^2 \cdot \sigma_s^2 \), 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 \( \frac{5}{2} \cdot n^3 \cdot \left( \frac{\ell}{12} \right)^2 \cdot \sigma_s^4 + 2 \cdot n^2 \cdot \left( \frac{\ell}{12} \right)^2 \cdot \sigma_s^2 \), since the noise already in the ciphertext is larger than this quantity. Thus, for \( n \in \mathbb{N} \), when

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

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 \( \frac{5}{2} \cdot n^3 \cdot \left( \frac{\ell}{12} \right)^2 \cdot \sigma_s^4 + 2 \cdot n^2 \cdot \left( \frac{\ell}{12} \right)^2 \cdot \sigma_s^2 \). 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 5.1 to obtain the hardness estimates for these circuits under a lattice reduction attack.

### 8.3 Guessing Attack for Class 1 and 2 Circuits

Now that we have determined \( \sigma_{y,s}^2, \sigma_{h,s}^2 \), and \( \sigma_s^2 \) 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 \( \sigma_s^2, \ldots, \sigma_s^2 \) and the upper left hand \( n \times n \) submatrix has \( n \) eigenvalues of the form \([\alpha_{2i+1}, \alpha_{2i+2}]\) for all \( i \in [n/2] \), \( \alpha_{2i+1} = \alpha_{2i+2} \). Further, for each \( i \in [n/2] \),

\[
\alpha_{2i+1} = \frac{\sigma_s^2}{1 + \frac{\sigma_{2i,R,i}^2}{2\sigma_s^2}}.
\]

Since with all but \( e^{-11} \) probability\(^\text{12} \), \( R_{1,i} \geq (2t - 6.63\sqrt{2t}) \cdot n \sigma_{h,s}^2 \), we have that with probability \( 1 - n/2 \cdot e^{-11} \)
all eigenvalues are less than

\[
\sigma_{max}^2 \leq \frac{\sigma_s^2}{1 + \frac{\sigma_s^2 (2t - 6.63 \sqrt{2t}) \cdot n \sigma_{h,s}^2}{2\sigma_s^2}}.
\]

\(^\text{12} \) For the parameter sets with \( n = 131072 \), we increase 6.63 below to 7.2 and decrease the probability to \( e^{-13} \).
We note that for the maximum setting of parameters $n$, $\sigma_{\text{max}}^2$, and $\mu$ guesses are greater than probability for some probability threshold $\mu$ implies that $\sigma_{\text{max}}^2$. Thus the total success probability of the attack is $-\text{erf} \left( \frac{-0.5}{\sqrt{2\sigma_{\text{max}}^2}} \right)^n - n/2 \cdot e^{-11}$. We note that for up to parameter $n = 32768$, $n/2 \cdot e^{-11} \leq 0.28$.  

### 8.4 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 $\sigma_{\text{max}}^2$, $\sigma_{\text{max}}^2$, and $\sigma_{\text{max}}^2$ in the two cases.

The guessing strategy for the hybrid attack is as follows: For each $i$, the adversary computes $e_i \Sigma_i e_i^T$ and guesses $\mu_e$ for the $g$ number of indices $i$ with the minimum values of $e_i \Sigma_i e_i^T$, where $g$ is the maximum value such that

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

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 $\Sigma_i$ of dimension $(n - g) \times (n - g)$, which we denote by $\Sigma_i''$. 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 $\Sigma_i''$. Using the Eigenvalue Interlacing Theorem [23], we have that $\text{det}(\Sigma_i'') \leq \frac{\text{det}(\Sigma)}{\alpha_1 \cdots \alpha_g}$. We therefore need a lower bound on $\alpha_1 \cdots \alpha_g$. Since with all but $e^{-11}$ probability$^{14}$, $R_{1,i} \leq (2t + 6.63\sqrt{2t} + 22) \cdot n\sigma_{\text{max}}^2$, we have that with probability $1 - n/2 \cdot e^{-11}$ all eigenvalues are greater than

$$ L = \frac{\sigma_i^2}{1 + \frac{\sigma_i^2}{\sqrt{2(2t + 6.63\sqrt{2t} + 22)} \cdot n\sigma_{\text{max}}^2}}. \tag{20} $$

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

$$ \text{det}(\Sigma_i'') \leq \frac{\text{det}(\Sigma)}{L^g}. \tag{21} $$

We note that for the maximum setting of parameters $n = 32768$, $n \cdot e^{-11} \leq 0.55$. Further, $\text{det}(\Sigma'')$ can be computed by plugging the parameter settings from Sections 8.1 and 8.2 into Lemma 5.1. Thus, we can use (21) to estimate the hardness of the residual instance (after guesses) under a lattice reduction attack.

### 9 Experiments

#### 9.1 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 [17]. 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 [28].

---

$^{13}$ And for $n = 131072$, $n/2 \cdot e^{-13} \leq 0.15$.

$^{14}$ For the parameter sets with $n = 131072$, we increase 6.63 below to 7.2 and decrease the probability to $e^{-13}$.

$^{15}$ And for $n = 131072$, $n \cdot e^{-13} \leq 0.30$.

$^{16}$ Our analysis may give slightly different concrete hardness estimates than the LWE Estimator [4], since [17] takes into account the ellipsoidal distribution of the original secret/error.
Experimental validation. Before the experiments on the concrete security estimation of CKKS, we first provide experimental validation of Lemma 5.1, in Section 9.2. We also provide concrete security estimation for provably secure (statistical) noise-flooding, as presented in [27]. We provide these as a baseline, and to validate our methods. Since there is no reduction in security when applying statistical noise-flooding, the results of those experiments are presented in Appendix A.

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 5, 6 and 7, 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 8.

Noise-flooding countermeasures. We use the results of [15] to estimate the output variance of the noise $\rho_{\text{circ}}^2$, where $\text{circ}$ is one of Identity, C1 or C2. We then consider noise-flooding by $\rho_{\text{circ}}^2$, 100 · $\rho_{\text{circ}}^2$ and $t$ · $\rho_{\text{circ}}^2$, where $t$ is the number of decryption queries. For guessing attacks, we do not include results for noise-flooding variance of $t$ · $\rho_{\text{circ}}^2$, 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 $t$ · $\rho_{\text{resh}}^2$, since no coordinates can be guessed with high confidence for any parameter set, and so the attack is equivalent to a lattice reduction attack.

9.2 Experimental Validation of Lemma 5.1

We first provide a verification of the theoretical results from Section 5, to demonstrate that the estimations hold in practice. In particular, Lemma 5.1 assumes that the distribution of the coefficients of $\epsilon_j^2$ and $\nu^j$ are independent Gaussians, while in practice this is not the case. The quantity of interest is $\det(\Sigma \sim)$, as defined in Section 5. In the proof of Lemma 5.1, we use the following fact:

$$\det(\Sigma \sim) = \frac{\det(H \Sigma H^T + \Sigma_c)}{\det(\Sigma_c) \det(\Sigma)} = \frac{\det(I_{2n} + \frac{1}{\sigma^2} \Sigma^{1/2} H H^T \Sigma^{1/2})}{\det(\Sigma)}.$$  \hspace{1cm} (22)

In order to validate the canonical embedding transformation used in the analysis of Lemma 5.1, we sample a random hint matrix $H$, directly compute $I_{2n} + \Sigma^{1/2} H H^T \Sigma^{1/2}/\sigma^2$, and calculate its determinant. In order to construct the hint matrix, we sample $\epsilon_j^2 \sim \chi$ and $\nu^j \sim S$ as defined in Appendix 3. 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 5.1. 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 compute 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 ($n = 256, t = 16$).

9.3 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 3. To calculate the concrete hardness, we apply Lemma 5.1 to obtain the expected volume and dimension of the lattice after hints are integrated and homogenization/isotropization.

\footnote{After $\sim 200$ million decryption queries, the estimated variance does not go below 3.6 for identity circuits, and after $\sim 100$ million decryption queries does not go below 0.33 and 0.36 for C1 and C2 circuits, respectively.}
<table>
<thead>
<tr>
<th>Dim</th>
<th>Num Hints</th>
<th>Predicted Determinant</th>
<th>Experimental Determinant</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>16</td>
<td>708.60</td>
<td>708.76</td>
</tr>
<tr>
<td>64</td>
<td>32</td>
<td>799.19</td>
<td>799.28</td>
</tr>
<tr>
<td>64</td>
<td>64</td>
<td>888.87</td>
<td>889.14</td>
</tr>
<tr>
<td>64</td>
<td>128</td>
<td>978.08</td>
<td>978.10</td>
</tr>
<tr>
<td>64</td>
<td>256</td>
<td>1067.04</td>
<td>1067.00</td>
</tr>
<tr>
<td>64</td>
<td>512</td>
<td>1155.89</td>
<td>1155.87</td>
</tr>
<tr>
<td>128</td>
<td>16</td>
<td>1594.55</td>
<td>1591.58</td>
</tr>
<tr>
<td>128</td>
<td>32</td>
<td>1175.78</td>
<td>1175.55</td>
</tr>
<tr>
<td>128</td>
<td>64</td>
<td>1955.17</td>
<td>1954.82</td>
</tr>
<tr>
<td>128</td>
<td>128</td>
<td>2133.59</td>
<td>2133.49</td>
</tr>
<tr>
<td>128</td>
<td>256</td>
<td>2311.52</td>
<td>2311.44</td>
</tr>
<tr>
<td>128</td>
<td>512</td>
<td>2489.22</td>
<td>2489.23</td>
</tr>
<tr>
<td>256</td>
<td>16</td>
<td>3543.88</td>
<td>3539.04</td>
</tr>
</tbody>
</table>

Fig. 2: **Summary of results for experimental validation of Lemma 5.1.** Each parameter set is specified by the dimension of the LWE secret and error (column 1) and the number of hints applied (column 2). The third column indicates the (natural log of) the expected value of the determinant as predicted by Lemma 5.1. The final column reports the determinant calculated by performing the experiment, as averaged over 256 trials.

is completed. As in [17], 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.

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

We next 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 4. In this attack, the adversary requests enough decryptions so that a 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 6.1 and (10).

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

We next 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 5. 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 [17]). 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 [17], 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.

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

**Class 1:** This is the same attack as in Section 9.3, except the adversary requests decryptions of ciphertexts that correspond to the evaluation of a Class 1 circuit (see Section 8.1 for the definition of this class) on fresh ciphertexts. To calculate the concrete hardness, we apply Lemma 5.1 to obtain the expected volume and dimension of the lattice after hints are integrated with the parameter settings for $\sigma_{h_s}, \sigma_{h_e}, \sigma_\epsilon$ given in Section 8.1. The results are reported in Figure 10 in Appendix A.
<table>
<thead>
<tr>
<th>Parameter Set</th>
<th>Original Security</th>
<th>Noise Variance</th>
<th>Num Queries (t)</th>
<th>Final Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\log_2 n = 10, \log_2 q_L = 25)</td>
<td>128/102 (p_{\text{fresh}}^2)</td>
<td>1000</td>
<td>247 bikz (\approx 65) bits</td>
<td></td>
</tr>
<tr>
<td>(\log_2 n = 10, \log_2 q_L = 17)</td>
<td>192/170 (p_{\text{fresh}}^2)</td>
<td>1000</td>
<td>336 bikz (\approx 89) bits</td>
<td></td>
</tr>
<tr>
<td>(\log_2 n = 10, \log_2 q_L = 13)</td>
<td>256/234 (p_{\text{fresh}}^2)</td>
<td>1000</td>
<td>374 bikz (\approx 99) bits</td>
<td></td>
</tr>
<tr>
<td>(\log_2 n = 11, \log_2 q_L = 51)</td>
<td>128/97 (p_{\text{fresh}}^2)</td>
<td>1000</td>
<td>366 bikz (\approx 97) bits</td>
<td></td>
</tr>
<tr>
<td>(\log_2 n = 11, \log_2 q_L = 35)</td>
<td>192/162 (p_{\text{fresh}}^2)</td>
<td>1000</td>
<td>536 bikz (\approx 142) bits</td>
<td></td>
</tr>
<tr>
<td>(\log_2 n = 11, \log_2 q_L = 27)</td>
<td>256/226 (p_{\text{fresh}}^2)</td>
<td>1000</td>
<td>615 bikz (\approx 163) bits</td>
<td></td>
</tr>
<tr>
<td>(\log_2 n = 12, \log_2 q_L = 101)</td>
<td>128/97 (p_{\text{fresh}}^2)</td>
<td>1000</td>
<td>599 bikz (\approx 159) bits</td>
<td></td>
</tr>
<tr>
<td>(\log_2 n = 12, \log_2 q_L = 70)</td>
<td>192/161 (p_{\text{fresh}}^2)</td>
<td>1000</td>
<td>761 bikz (\approx 201) bits</td>
<td></td>
</tr>
<tr>
<td>(\log_2 n = 12, \log_2 q_L = 54)</td>
<td>256/227 (p_{\text{fresh}}^2)</td>
<td>1000</td>
<td>845 bikz (\approx 224) bits</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 3: 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 [17]. The third column indicates the noise-flooding noise added before returning the decryption. \(p_{\text{fresh}}^2\) is the variance of the noise that is already present in a fresh ciphertext (see Section 3.5.1). 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 [17]) and bit-security.

**Class 2**: This is the same attack as in Section 9.3, except the adversary requests decryptions of ciphertexts that correspond to the evaluation of a Class 2 circuit (see Section 8.2 for the definition of this class) on fresh ciphertexts. To calculate the concrete hardness, we apply Lemma 5.1 to obtain the expected volume and dimension of the lattice after hints are integrated with the parameter settings for \(\sigma^2_{h}, \sigma^2_{h}, \sigma^2_{e}\) given in Section 8.2. The results are reported in Figure 11 in Appendix A.

**9.7 Concrete Security of Guessing Attacks on Class 1 and 2 Circuits**

**Class 1**: This is the same attack as in Section 9.4, except the adversary requests decryptions of ciphertexts that correspond to the evaluation of a Class 1 circuit (see Section 8.1 for the definition of this class) 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 \(\sigma_{h}, \sigma_{h}, \sigma_{e}\) given in Section 8.1. The results for various noise-flooding levels are reported in Figure 12 in Appendix A.

**Class 2**: This is the same attack as in Section 9.4, except the adversary requests decryptions of ciphertexts that correspond to the evaluation of a Class 2 circuit (see Section 8.2 for the definition of this class) on
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 [17]. The third column indicates the noise-flooding noise added before returning the decryption. $\rho_{\text{fresh}}$ is the variance of the noise that is already present in a fresh ciphertext (see Section 3.5.1). 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 [17]) and bit-security.

fresh ciphertexts. To determine the number of decryptions required to recover the LWE secret with high probability, we apply (18) with the settings of $\sigma_{h_x}, \sigma_{h_y}, \sigma_\epsilon^2$ given in Section 8.2. The results for various noise-flooding levels are reported in Figure 13 in Appendix A.

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

**Class 1:** This is the same attack as in Section 9.5, except the adversary requests decryptions of ciphertexts that correspond to the evaluation of a Class 1 circuit (see Section 8.1 for the definition of this class) on fresh ciphertexts. For a fixed number of decryptions, we use (11), with the settings of $\sigma_{h_2}, \sigma_{h_2}^2$, and $\sigma_\epsilon^2$ given in Section 8.1, to determine the number of guesses $g$ that can be made such that all guesses are correct with

<table>
<thead>
<tr>
<th>Parameter Set</th>
<th>Original Security</th>
<th>Original Noise Variance</th>
<th>Num Queries ($t$)</th>
<th>Final Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\log_2 n = 13, \log_2 q_L = 202$</td>
<td>$\rho_{\text{fresh}}^2$</td>
<td>1000</td>
<td>$340 \text{ bikz} \approx 90 \text{ bits}$</td>
<td>$128/96$</td>
</tr>
<tr>
<td>$\log_2 n = 13, \log_2 q_L = 141$</td>
<td>$\rho_{\text{fresh}}^2$</td>
<td>1000</td>
<td>$356 \text{ bikz} \approx 94 \text{ bits}$</td>
<td>$192/159$</td>
</tr>
<tr>
<td>$\log_2 n = 13, \log_2 q_L = 109$</td>
<td>$\rho_{\text{fresh}}^2$</td>
<td>1000</td>
<td>$361 \text{ bikz} \approx 96 \text{ bits}$</td>
<td>$256/225$</td>
</tr>
<tr>
<td>$\log_2 n = 14, \log_2 q_L = 411$</td>
<td>$\rho_{\text{fresh}}^2$</td>
<td>1000</td>
<td>$553 \text{ bikz} \approx 146 \text{ bits}$</td>
<td>$187/186$</td>
</tr>
<tr>
<td>$\log_2 n = 14, \log_2 q_L = 284$</td>
<td>$\rho_{\text{fresh}}^2$</td>
<td>1000</td>
<td>$587 \text{ bikz} \approx 156 \text{ bits}$</td>
<td>$192/158$</td>
</tr>
<tr>
<td>$\log_2 n = 14, \log_2 q_L = 220$</td>
<td>$\rho_{\text{fresh}}^2$</td>
<td>1000</td>
<td>$593 \text{ bikz} \approx 157 \text{ bits}$</td>
<td>$256/222$</td>
</tr>
<tr>
<td>$\log_2 n = 15, \log_2 q_L = 827$</td>
<td>$\rho_{\text{fresh}}^2$</td>
<td>1000</td>
<td>$577 \text{ bikz} \approx 153 \text{ bits}$</td>
<td>$128/92$</td>
</tr>
<tr>
<td>$\log_2 n = 15, \log_2 q_L = 571$</td>
<td>$\rho_{\text{fresh}}^2$</td>
<td>1000</td>
<td>$586 \text{ bikz} \approx 155 \text{ bits}$</td>
<td>$192/156$</td>
</tr>
<tr>
<td>$\log_2 n = 15, \log_2 q_L = 443$</td>
<td>$\rho_{\text{fresh}}^2$</td>
<td>1000</td>
<td>$589 \text{ bikz} \approx 156 \text{ bits}$</td>
<td>$256/220$</td>
</tr>
<tr>
<td>$\log_2 n = 17, \log_2 q_L = 2400$</td>
<td>$\rho_{\text{fresh}}^2$</td>
<td>1000</td>
<td>$548 \text{ bikz} \approx 145 \text{ bits}$</td>
<td>$140/146$</td>
</tr>
<tr>
<td>$\log_2 n = 17, \log_2 q_L = 2000$</td>
<td>$\rho_{\text{fresh}}^2$</td>
<td>1000</td>
<td>$550 \text{ bikz} \approx 146 \text{ bits}$</td>
<td>$193/187$</td>
</tr>
</tbody>
</table>

---

Fig. 3: 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 [17]. The third column indicates the noise-flooding noise added before returning the decryption. $\rho_{\text{fresh}}$ is the variance of the noise that is already present in a fresh ciphertext (see Section 3.5.1). 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 [17]) and bit-security.
For each parameter set, the second column provides the target security and the number of bits of security computed by the tool of [17]. 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 6.1 occurring.

**Class 2:** This is the same attack as in Section 9.5, except the adversary requests decryptions of ciphertexts that correspond to the evaluation of a Class 2 circuit (see Section 8.2 for the definition of this class) on fresh ciphertexts. For a fixed number of decryptions, we use (11), with the settings of \( \sigma_{h_z}^2, \sigma_{h_z}^2, \) and \( \sigma_{\epsilon}^2 \) given in Section 8.1. The results are reported in Figure 14 in Appendix A.

**Discussion of the results**

**Trends for noise-flooding level of \( \rho_{\text{circ}}^2 \):** Our experimental data is summarized via the graphs in Figure 6. Figure 6(a) shows the reduction in bit security for a lattice reduction attack when considering an adversary who obtains 1000 decryptions of identity, Class 1, and Class 2 circuits with noise-flooding level \( \rho_{\text{circ}}^2 \) equal to the noise that is 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 for this is that the hints in identity circuits involve all \( 2n \) coordinates in the LWE secret/error, so the variance of all \( 2n \) coordinates is reduced after each hint. On the other hand, hints in 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 after each hint. We also note that there is a greater security reduction for higher target security level vs. lower target security level. For example, for the lattice reduction attack, we see that for \( \log_2(n) = 10 \), identity circuits, and for

<table>
<thead>
<tr>
<th>Parameter Set</th>
<th>Orig Security</th>
<th>Noise Var</th>
<th>Num Queries</th>
<th>Succ Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \log_2 n = 10, \log_2 qL = 25 )</td>
<td>128/102 ( \rho_{\text{fresh}}^2 )</td>
<td>1160 0.81</td>
<td>100 ( \rho_{\text{fresh}}^2 )</td>
<td>62,180 0.80</td>
</tr>
<tr>
<td>( \log_2 n = 10, \log_2 qL = 17 )</td>
<td>192/170 ( \rho_{\text{fresh}}^2 )</td>
<td>1160 0.81</td>
<td>100 ( \rho_{\text{fresh}}^2 )</td>
<td>62,180 0.80</td>
</tr>
<tr>
<td>( \log_2 n = 10, \log_2 qL = 13 )</td>
<td>256/234 ( \rho_{\text{fresh}}^2 )</td>
<td>1160 0.81</td>
<td>100 ( \rho_{\text{fresh}}^2 )</td>
<td>62,180 0.80</td>
</tr>
<tr>
<td>( \log_2 n = 11, \log_2 qL = 51 )</td>
<td>128/97 ( \rho_{\text{fresh}}^2 )</td>
<td>1220 0.80</td>
<td>100 ( \rho_{\text{fresh}}^2 )</td>
<td>67,950 0.80</td>
</tr>
<tr>
<td>( \log_2 n = 11, \log_2 qL = 35 )</td>
<td>192/162 ( \rho_{\text{fresh}}^2 )</td>
<td>1220 0.80</td>
<td>100 ( \rho_{\text{fresh}}^2 )</td>
<td>67,950 0.80</td>
</tr>
<tr>
<td>( \log_2 n = 11, \log_2 qL = 27 )</td>
<td>256/226 ( \rho_{\text{fresh}}^2 )</td>
<td>1220 0.80</td>
<td>100 ( \rho_{\text{fresh}}^2 )</td>
<td>67,950 0.80</td>
</tr>
<tr>
<td>( \log_2 n = 12, \log_2 qL = 101 )</td>
<td>128/97 ( \rho_{\text{fresh}}^2 )</td>
<td>1290 0.81</td>
<td>100 ( \rho_{\text{fresh}}^2 )</td>
<td>73,760 0.80</td>
</tr>
<tr>
<td>( \log_2 n = 12, \log_2 qL = 70 )</td>
<td>192/161 ( \rho_{\text{fresh}}^2 )</td>
<td>1290 0.81</td>
<td>100 ( \rho_{\text{fresh}}^2 )</td>
<td>73,760 0.80</td>
</tr>
<tr>
<td>( \log_2 n = 12, \log_2 qL = 54 )</td>
<td>256/227 ( \rho_{\text{fresh}}^2 )</td>
<td>1290 0.81</td>
<td>100 ( \rho_{\text{fresh}}^2 )</td>
<td>73,760 0.80</td>
</tr>
</tbody>
</table>

**Fig. 4:** 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 [17]. 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 6.1 occurring.
<table>
<thead>
<tr>
<th>Parameter Set</th>
<th>Orig Security</th>
<th>Noise Var</th>
<th>Num Queries</th>
<th>Succ Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \log_2 n = 13, \log_2 q_L = 202 )</td>
<td>128/96 ( \rho_{\text{fresh}}^2 )</td>
<td>1350</td>
<td>0.80</td>
<td></td>
</tr>
<tr>
<td>( \log_2 n = 13, \log_2 q_L = 141 )</td>
<td>192/159 ( \rho_{\text{fresh}}^2 )</td>
<td>79,600</td>
<td>0.80</td>
<td></td>
</tr>
<tr>
<td>( \log_2 n = 13, \log_2 q_L = 109 )</td>
<td>256/225 ( \rho_{\text{fresh}}^2 )</td>
<td>79,600</td>
<td>0.80</td>
<td></td>
</tr>
<tr>
<td>( \log_2 n = 14, \log_2 q_L = 411 )</td>
<td>128/93 ( \rho_{\text{fresh}}^2 )</td>
<td>1420</td>
<td>0.81</td>
<td></td>
</tr>
<tr>
<td>( \log_2 n = 14, \log_2 q_L = 284 )</td>
<td>192/158 ( \rho_{\text{fresh}}^2 )</td>
<td>85,450</td>
<td>0.80</td>
<td></td>
</tr>
<tr>
<td>( \log_2 n = 14, \log_2 q_L = 220 )</td>
<td>256/222 ( \rho_{\text{fresh}}^2 )</td>
<td>85,450</td>
<td>0.80</td>
<td></td>
</tr>
<tr>
<td>( \log_2 n = 15, \log_2 q_L = 827 )</td>
<td>128/92 ( \rho_{\text{fresh}}^2 )</td>
<td>1480</td>
<td>0.80</td>
<td></td>
</tr>
<tr>
<td>( \log_2 n = 15, \log_2 q_L = 571 )</td>
<td>192/156 ( \rho_{\text{fresh}}^2 )</td>
<td>1480</td>
<td>0.80</td>
<td></td>
</tr>
<tr>
<td>( \log_2 n = 15, \log_2 q_L = 443 )</td>
<td>256/220 ( \rho_{\text{fresh}}^2 )</td>
<td>1480</td>
<td>0.80</td>
<td></td>
</tr>
<tr>
<td>( \log_2 n = 17, \log_2 q_L = 2400 )</td>
<td>140/146 ( \rho_{\text{fresh}}^2 )</td>
<td>103,360</td>
<td>0.80</td>
<td></td>
</tr>
<tr>
<td>( \log_2 n = 17, \log_2 q_L = 2000 )</td>
<td>193/187 ( \rho_{\text{fresh}}^2 )</td>
<td>103,360</td>
<td>0.82</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 4: **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 [17]. 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 6.1 occurring.

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 to be highly correlated with the setting of the 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 6(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 (we believe this occurs because for identity circuits, hints correspond to linear combinations of both the \( s \) and \( e \) variables in the LWE instance, 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 average 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 \( T/(2n) \). On the other hand, for Class 1 and 2 circuits, hints correspond to linear combinations of only the \( s \) variables from the LWE instance. Thus, we restrict our
the ciphertext. As expected, we see that the biggest drop in bit security is observed when noise-flooding by
the number of decryption queries, security level or in the bikz for any parameter setting.

![Trend](Figures/Trend.png) shows the reduction in bit-security is greater for identity circuits versus Class 1 and Class 2 circuits, as can be
each hint slightly reducing the variance). This explains why for approximately the same number of guesses,
instance has lower variance in the case of identity circuits (since more hints have been incorporated, with
is far higher for identity circuits than Class 1 and Class 2 circuits. Thus, after guesses are made, the residual
attention to a subspace with only $n$ eigenvalues that are reduced by obtaining hints. All of these eigenvalues
have the same distribution, and our proof shows that all the eigenvalues are less than maximum value $\sigma_{max}^2$.

Figure 6(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 6(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.

**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 [27]. Our results are presented in
Figures 7, 8 and 9 in Appendix A. 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 $\rho_{\text{circ}}^2$, $100 \cdot \rho_{\text{circ}}^2$, and $t \cdot \rho_{\text{circ}}^2$, where $t$
is the number of decryption queries, circ is one of Identity, C1 or C2, and $\rho_{\text{circ}}$ 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
$\rho_{\text{circ}}$, across all parameter sets and across all circuits.

<table>
<thead>
<tr>
<th>Parameter Set</th>
<th>Orig Security</th>
<th>Noise Var</th>
<th>Num Queries</th>
<th>Num Succ</th>
<th>Final Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\log_2 n = 10, \log_2 q_L = 25$</td>
<td>128/102</td>
<td>$\rho_{\text{fresh}}^2$</td>
<td>1130</td>
<td>793</td>
<td>0.80</td>
</tr>
<tr>
<td>$\log_2 n = 10, \log_2 q_L = 17$</td>
<td>192/170</td>
<td>$\rho_{\text{fresh}}^2$</td>
<td>1130</td>
<td>793</td>
<td>0.80</td>
</tr>
<tr>
<td>$\log_2 n = 10, \log_2 q_L = 13$</td>
<td>256/234</td>
<td>$\rho_{\text{fresh}}^2$</td>
<td>1130</td>
<td>793</td>
<td>0.80</td>
</tr>
<tr>
<td>$\log_2 n = 11, \log_2 q_L = 51$</td>
<td>128/97</td>
<td>$\rho_{\text{fresh}}^2$</td>
<td>1130</td>
<td>801</td>
<td>0.80</td>
</tr>
<tr>
<td>$\log_2 n = 11, \log_2 q_L = 35$</td>
<td>192/162</td>
<td>$\rho_{\text{fresh}}^2$</td>
<td>1130</td>
<td>801</td>
<td>0.80</td>
</tr>
<tr>
<td>$\log_2 n = 11, \log_2 q_L = 27$</td>
<td>256/226</td>
<td>$\rho_{\text{fresh}}^2$</td>
<td>1130</td>
<td>801</td>
<td>0.80</td>
</tr>
<tr>
<td>$\log_2 n = 12, \log_2 q_L = 101$</td>
<td>128/97</td>
<td>$\rho_{\text{fresh}}^2$</td>
<td>1130</td>
<td>807</td>
<td>0.80</td>
</tr>
<tr>
<td>$\log_2 n = 12, \log_2 q_L = 70$</td>
<td>192/161</td>
<td>$\rho_{\text{fresh}}^2$</td>
<td>1130</td>
<td>807</td>
<td>0.80</td>
</tr>
<tr>
<td>$\log_2 n = 12, \log_2 q_L = 54$</td>
<td>256/227</td>
<td>$\rho_{\text{fresh}}^2$</td>
<td>1130</td>
<td>807</td>
<td>0.80</td>
</tr>
</tbody>
</table>

Fig. 5: **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 [17]. 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 6.1 and (15) occurring. The final column
provides the reduced security level after the attack in terms of bikz (see [17]) and bit-security.
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 3 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 3 that for $\log_2(n) = 17$, there is no change in the security level.

**Conclusions:** The first conclusion of this work is that, in practice, it is enough to noise flood by $t \cdot \rho_{\text{circ}}^2$, where $t$ is the number of decryption queries available to the adversary, and $\rho_{\text{circ}}^2$ is the variance of the noise, as predicted by an average-case noise analysis. Perhaps a less cautious approach is to noise flood by $\alpha \cdot t \cdot \rho$, where $0 < \alpha < 1$, if it is acceptable to have the security level drop by a few bits. We note that 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, we note that one can think of increasing $\alpha$ as a way to allow for more decryption queries. 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. Specifically, 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.

---

<table>
<thead>
<tr>
<th>Parameter Set</th>
<th>Orig Security</th>
<th>Noise Var</th>
<th>Num Queries</th>
<th>Num Succ</th>
<th>Final Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\log_2 n = 13$, $\log_2 q_L = 202$</td>
<td>128/96</td>
<td>$\rho_{\text{fresh}}^2$</td>
<td>1130</td>
<td>810</td>
<td>0.80</td>
</tr>
<tr>
<td>$\log_2 n = 13$, $\log_2 q_L = 141$</td>
<td>192/159</td>
<td>$\rho_{\text{fresh}}^2$</td>
<td>1130</td>
<td>810</td>
<td>0.80</td>
</tr>
<tr>
<td>$\log_2 n = 13$, $\log_2 q_L = 109$</td>
<td>256/225</td>
<td>$\rho_{\text{fresh}}^2$</td>
<td>1130</td>
<td>810</td>
<td>0.80</td>
</tr>
<tr>
<td>$\log_2 n = 14$, $\log_2 q_L = 411$</td>
<td>128/93</td>
<td>$\rho_{\text{fresh}}^2$</td>
<td>1130</td>
<td>813</td>
<td>0.80</td>
</tr>
<tr>
<td>$\log_2 n = 14$, $\log_2 q_L = 184$</td>
<td>192/158</td>
<td>$\rho_{\text{fresh}}^2$</td>
<td>1130</td>
<td>813</td>
<td>0.80</td>
</tr>
<tr>
<td>$\log_2 n = 14$, $\log_2 q_L = 220$</td>
<td>256/222</td>
<td>$\rho_{\text{fresh}}^2$</td>
<td>1130</td>
<td>813</td>
<td>0.80</td>
</tr>
<tr>
<td>$\log_2 n = 15$, $\log_2 q_L = 827$</td>
<td>128/92</td>
<td>$\rho_{\text{fresh}}^2$</td>
<td>1130</td>
<td>814</td>
<td>0.80</td>
</tr>
<tr>
<td>$\log_2 n = 15$, $\log_2 q_L = 571$</td>
<td>192/156</td>
<td>$\rho_{\text{fresh}}^2$</td>
<td>1130</td>
<td>814</td>
<td>0.80</td>
</tr>
<tr>
<td>$\log_2 n = 15$, $\log_2 q_L = 443$</td>
<td>256/220</td>
<td>$\rho_{\text{fresh}}^2$</td>
<td>1130</td>
<td>814</td>
<td>0.80</td>
</tr>
<tr>
<td>$\log_2 n = 17$, $\log_2 q_L = 2400$</td>
<td>140/146</td>
<td>$\rho_{\text{fresh}}^2$</td>
<td>1260</td>
<td>1557</td>
<td>0.80</td>
</tr>
<tr>
<td>$\log_2 n = 17$, $\log_2 q_L = 2000$</td>
<td>193/187</td>
<td>$\rho_{\text{fresh}}^2$</td>
<td>1260</td>
<td>1557</td>
<td>0.80</td>
</tr>
</tbody>
</table>

Fig. 5: 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 [17]. The third column indicates the noise-flooding variance added before returning the decryption. $\rho_{\text{fresh}}$ 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 6.1 and (15) occurring. The final column provides the reduced security level after the attack in terms of bikz (see [17]) and bit-security.
Fig. 6: **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 $\rho_{\text{fresh}}$, $\rho_{\text{C1}}$, and $\rho_{\text{C2}}$, 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).
Acknowledgements

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

References


7. Private Communication, anonymized for submission.


### A Additional Experimental Results

For each parameter set, the second column provides the target security as well as the number of bits of security added before returning the decryption to the adversary. \( \rho_{\text{stat}} = 12 \cdot t \cdot 2^\kappa \cdot \rho_{\text{fresh}} \), where \( \rho_{\text{fresh}} \) is the variance of the noise that is already present in a fresh ciphertext (see Section 3.5.1), 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 [17]) and bit-security.

<table>
<thead>
<tr>
<th>Parameter Set</th>
<th>Original Security</th>
<th>Noise Variance</th>
<th>Final Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \log_2 n = 10, \log_2 qL = 25 )</td>
<td>( \frac{192}{102.34} ) (386.21) ( \rho_{\text{stat}} )</td>
<td>386.21 bikz ( \approx 102.34 ) bits</td>
<td></td>
</tr>
<tr>
<td>( \log_2 n = 10, \log_2 qL = 17 )</td>
<td>( \frac{192}{176.04} ) (641.05) ( \rho_{\text{stat}} )</td>
<td>641.05 bikz ( \approx 170.04 ) bits</td>
<td></td>
</tr>
<tr>
<td>( \log_2 n = 10, \log_2 qL = 13 )</td>
<td>( \frac{256}{234.29} ) (884.13) ( \rho_{\text{stat}} )</td>
<td>884.13 bikz ( \approx 234.29 ) bits</td>
<td></td>
</tr>
<tr>
<td>( \log_2 n = 11, \log_2 qL = 51 )</td>
<td>( \frac{128}{96.84} ) (365.43) ( \rho_{\text{stat}} )</td>
<td>365.43 bikz ( \approx 96.84 ) bits</td>
<td></td>
</tr>
<tr>
<td>( \log_2 n = 11, \log_2 qL = 45 )</td>
<td>( \frac{192}{162.31} ) (612.49) ( \rho_{\text{stat}} )</td>
<td>612.49 bikz ( \approx 162.31 ) bits</td>
<td></td>
</tr>
<tr>
<td>( \log_2 n = 11, \log_2 qL = 27 )</td>
<td>( \frac{256}{226.11} ) (853.25) ( \rho_{\text{stat}} )</td>
<td>853.25 bikz ( \approx 226.11 ) bits</td>
<td></td>
</tr>
<tr>
<td>( \log_2 n = 12, \log_2 qL = 101 )</td>
<td>( \frac{128}{96.81} ) (365.34) ( \rho_{\text{stat}} )</td>
<td>365.34 bikz ( \approx 96.81 ) bits</td>
<td></td>
</tr>
<tr>
<td>( \log_2 n = 12, \log_2 qL = 70 )</td>
<td>( \frac{192}{161.41} ) (606.11) ( \rho_{\text{stat}} )</td>
<td>606.11 bikz ( \approx 161.41 ) bits</td>
<td></td>
</tr>
<tr>
<td>( \log_2 n = 12, \log_2 qL = 54 )</td>
<td>( \frac{256}{227.10} ) (856.98) ( \rho_{\text{stat}} )</td>
<td>856.98 bikz ( \approx 227.10 ) bits</td>
<td></td>
</tr>
<tr>
<td>( \log_2 n = 13, \log_2 qL = 202 )</td>
<td>( \frac{128}{96.11} ) (362.66) ( \rho_{\text{stat}} )</td>
<td>362.66 bikz ( \approx 96.11 ) bits</td>
<td></td>
</tr>
<tr>
<td>( \log_2 n = 13, \log_2 qL = 141 )</td>
<td>( \frac{192}{159.40} ) (601.49) ( \rho_{\text{stat}} )</td>
<td>601.49 bikz ( \approx 159.40 ) bits</td>
<td></td>
</tr>
<tr>
<td>( \log_2 n = 13, \log_2 qL = 109 )</td>
<td>( \frac{256}{224.89} ) (848.03) ( \rho_{\text{stat}} )</td>
<td>848.03 bikz ( \approx 224.89 ) bits</td>
<td></td>
</tr>
<tr>
<td>( \log_2 n = 14, \log_2 qL = 411 )</td>
<td>( \frac{128}{93.37} ) (352.34) ( \rho_{\text{stat}} )</td>
<td>352.34 bikz ( \approx 93.37 ) bits</td>
<td></td>
</tr>
<tr>
<td>( \log_2 n = 14, \log_2 qL = 284 )</td>
<td>( \frac{192}{157.62} ) (594.78) ( \rho_{\text{stat}} )</td>
<td>594.78 bikz ( \approx 157.62 ) bits</td>
<td></td>
</tr>
<tr>
<td>( \log_2 n = 14, \log_2 qL = 220 )</td>
<td>( \frac{256}{222.42} ) (839.32) ( \rho_{\text{stat}} )</td>
<td>839.32 bikz ( \approx 222.42 ) bits</td>
<td></td>
</tr>
<tr>
<td>( \log_2 n = 15, \log_2 qL = 827 )</td>
<td>( \frac{128}{92.37} ) (348.55) ( \rho_{\text{stat}} )</td>
<td>348.55 bikz ( \approx 92.37 ) bits</td>
<td></td>
</tr>
<tr>
<td>( \log_2 n = 15, \log_2 qL = 571 )</td>
<td>( \frac{192}{156.35} ) (596.00) ( \rho_{\text{stat}} )</td>
<td>596.00 bikz ( \approx 156.35 ) bits</td>
<td></td>
</tr>
<tr>
<td>( \log_2 n = 15, \log_2 qL = 443 )</td>
<td>( \frac{256}{220.52} ) (832.15) ( \rho_{\text{stat}} )</td>
<td>832.15 bikz ( \approx 220.52 ) bits</td>
<td></td>
</tr>
<tr>
<td>( \log_2 n = 17, \log_2 qL = 2400 )</td>
<td>( \frac{128}{145.88} ) (550.51) ( \rho_{\text{stat}} )</td>
<td>550.51 bikz ( \approx 145.88 ) bits</td>
<td></td>
</tr>
<tr>
<td>( \log_2 n = 17, \log_2 qL = 2000 )</td>
<td>( \frac{192}{187.40} ) (707.17) ( \rho_{\text{stat}} )</td>
<td>707.17 bikz ( \approx 187.40 ) bits</td>
<td></td>
</tr>
</tbody>
</table>
For each parameter set, the second column provides the target security as well as the number of bits of security. 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 [17]) and bit-security.

<table>
<thead>
<tr>
<th>Parameter Set</th>
<th>Original Security</th>
<th>Noise Variance</th>
<th>Final Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\log_2 n = 10, \log_2 q_L = 25)</td>
<td>128/102.34 (386.21)</td>
<td>(\rho_{stat}^{2})</td>
<td>386.21 bikz (\approx 102.34) bits</td>
</tr>
<tr>
<td>(\log_2 n = 10, \log_2 q_L = 17)</td>
<td>192/161.04 (641.65)</td>
<td>(\rho_{stat}^{2})</td>
<td>641.65 bikz (\approx 170.64) bits</td>
</tr>
<tr>
<td>(\log_2 n = 10, \log_2 q_L = 13)</td>
<td>256/234.29 (884.13)</td>
<td>(\rho_{stat}^{2})</td>
<td>884.13 bikz (\approx 234.29) bits</td>
</tr>
<tr>
<td>(\log_2 n = 11, \log_2 q_L = 51)</td>
<td>128/96.84 (365.43)</td>
<td>(\rho_{stat}^{2})</td>
<td>365.43 bikz (\approx 96.84) bits</td>
</tr>
<tr>
<td>(\log_2 n = 11, \log_2 q_L = 35)</td>
<td>192/162.31 (612.49)</td>
<td>(\rho_{stat}^{2})</td>
<td>612.49 bikz (\approx 162.31) bits</td>
</tr>
<tr>
<td>(\log_2 n = 11, \log_2 q_L = 27)</td>
<td>256/226.11 (853.25)</td>
<td>(\rho_{stat}^{2})</td>
<td>853.25 bikz (\approx 226.11) bits</td>
</tr>
<tr>
<td>(\log_2 n = 12, \log_2 q_L = 101)</td>
<td>128/96.81 (365.34)</td>
<td>(\rho_{stat}^{2})</td>
<td>365.34 bikz (\approx 96.81) bits</td>
</tr>
<tr>
<td>(\log_2 n = 12, \log_2 q_L = 70)</td>
<td>192/161.11 (640.11)</td>
<td>(\rho_{stat}^{2})</td>
<td>640.11 bikz (\approx 161.11) bits</td>
</tr>
<tr>
<td>(\log_2 n = 12, \log_2 q_L = 54)</td>
<td>256/227.10 (886.98)</td>
<td>(\rho_{stat}^{2})</td>
<td>886.98 bikz (\approx 227.10) bits</td>
</tr>
<tr>
<td>(\log_2 n = 13, \log_2 q_L = 202)</td>
<td>128/96.11 (362.66)</td>
<td>(\rho_{stat}^{2})</td>
<td>362.66 bikz (\approx 96.11) bits</td>
</tr>
<tr>
<td>(\log_2 n = 13, \log_2 q_L = 141)</td>
<td>192/156.40 (601.40)</td>
<td>(\rho_{stat}^{2})</td>
<td>601.40 bikz (\approx 156.40) bits</td>
</tr>
<tr>
<td>(\log_2 n = 13, \log_2 q_L = 100)</td>
<td>256/224.80 (848.63)</td>
<td>(\rho_{stat}^{2})</td>
<td>848.63 bikz (\approx 224.80) bits</td>
</tr>
<tr>
<td>(\log_2 n = 14, \log_2 q_L = 411)</td>
<td>128/93.37 (352.34)</td>
<td>(\rho_{stat}^{2})</td>
<td>352.34 bikz (\approx 93.37) bits</td>
</tr>
<tr>
<td>(\log_2 n = 14, \log_2 q_L = 284)</td>
<td>192/157.62 (594.78)</td>
<td>(\rho_{stat}^{2})</td>
<td>594.78 bikz (\approx 157.62) bits</td>
</tr>
<tr>
<td>(\log_2 n = 14, \log_2 q_L = 220)</td>
<td>256/222.42 (839.32)</td>
<td>(\rho_{stat}^{2})</td>
<td>839.32 bikz (\approx 222.42) bits</td>
</tr>
<tr>
<td>(\log_2 n = 15, \log_2 q_L = 827)</td>
<td>128/92.37 (348.55)</td>
<td>(\rho_{stat}^{2})</td>
<td>348.55 bikz (\approx 92.37) bits</td>
</tr>
<tr>
<td>(\log_2 n = 15, \log_2 q_L = 571)</td>
<td>192/156.35 (596.00)</td>
<td>(\rho_{stat}^{2})</td>
<td>596.00 bikz (\approx 156.35) bits</td>
</tr>
<tr>
<td>(\log_2 n = 15, \log_2 q_L = 443)</td>
<td>256/226.52 (832.15)</td>
<td>(\rho_{stat}^{2})</td>
<td>832.15 bikz (\approx 226.52) bits</td>
</tr>
<tr>
<td>(\log_2 n = 17, \log_2 q_L = 2400)</td>
<td>140/145.88 (550.51)</td>
<td>(\rho_{stat}^{2})</td>
<td>550.51 bikz (\approx 145.88) bits</td>
</tr>
<tr>
<td>(\log_2 n = 17, \log_2 q_L = 2000)</td>
<td>163/187.40 (707.17)</td>
<td>(\rho_{stat}^{2})</td>
<td>707.17 bikz (\approx 187.40) bits</td>
</tr>
</tbody>
</table>

Fig. 8: 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 [17] (bikz are provided in parenthesis). The third column indicates the noise-flooding noise added before returning the decryption to the adversary. \(\rho_{stat}^{2} = 12 \cdot t \cdot 2^{\kappa} \cdot \rho_{\mathcal{C}_2}\), where \(\rho_{\mathcal{C}_2}\) 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 3.5.1). 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 [17]) and bit-security.
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 \([17]\) (bikz are provided in parenthesis). The third column indicates the noise-flooding noise added before returning the decryption to the adversary. \(\rho_{\text{stat}}^2 = 12 \cdot t \cdot 2^k \cdot \rho_{\text{L2}}^2\), where \(\rho_{\text{L2}}^2\) is the variance of the noise that is already present in a ciphertext obtained from evaluating a Class 2 circuit on fresh encryptions, and \(k = 30\) 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 \([17]\)) and bit-security. The (encoded) message magnitude is equal to \(n \cdot \sqrt{\ell/3}\) in all rows, where \(\ell\) is set to 20.

<table>
<thead>
<tr>
<th>Parameter Set</th>
<th>Original Security</th>
<th>Noise Variance</th>
<th>Final Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\log_2 n = 10, \log_2 qL = 25)</td>
<td>128/102.34 (386.21)</td>
<td>(\rho_{\text{stat}}^2)</td>
<td>386.21 bikz (\approx) 102.34 bits</td>
</tr>
<tr>
<td>(\log_2 n = 10, \log_2 qL = 17)</td>
<td>192/170.04 (741.65)</td>
<td>(\rho_{\text{stat}}^2)</td>
<td>641.65 bikz (\approx) 170.04 bits</td>
</tr>
<tr>
<td>(\log_2 n = 10, \log_2 qL = 13)</td>
<td>256/234.29 (884.13)</td>
<td>(\rho_{\text{stat}}^2)</td>
<td>884.13 bikz (\approx) 234.29 bits</td>
</tr>
<tr>
<td>(\log_2 n = 11, \log_2 qL = 51)</td>
<td>128/96.84 (365.43)</td>
<td>(\rho_{\text{stat}}^2)</td>
<td>365.43 bikz (\approx) 96.84 bits</td>
</tr>
<tr>
<td>(\log_2 n = 11, \log_2 qL = 35)</td>
<td>192/162.31 (612.49)</td>
<td>(\rho_{\text{stat}}^2)</td>
<td>612.49 bikz (\approx) 162.31 bits</td>
</tr>
<tr>
<td>(\log_2 n = 11, \log_2 qL = 27)</td>
<td>256/226.11 (853.25)</td>
<td>(\rho_{\text{stat}}^2)</td>
<td>853.25 bikz (\approx) 226.11 bits</td>
</tr>
<tr>
<td>(\log_2 n = 12, \log_2 qL = 101)</td>
<td>128/96.81 (365.34)</td>
<td>(\rho_{\text{stat}}^2)</td>
<td>365.34 bikz (\approx) 96.81 bits</td>
</tr>
<tr>
<td>(\log_2 n = 12, \log_2 qL = 70)</td>
<td>192/162.31 (612.49)</td>
<td>(\rho_{\text{stat}}^2)</td>
<td>612.49 bikz (\approx) 162.31 bits</td>
</tr>
<tr>
<td>(\log_2 n = 12, \log_2 qL = 54)</td>
<td>256/226.11 (853.25)</td>
<td>(\rho_{\text{stat}}^2)</td>
<td>856.98 bikz (\approx) 227.10 bits</td>
</tr>
<tr>
<td>(\log_2 n = 13, \log_2 qL = 202)</td>
<td>128/96.11 (362.66)</td>
<td>(\rho_{\text{stat}}^2)</td>
<td>362.66 bikz (\approx) 96.11 bits</td>
</tr>
<tr>
<td>(\log_2 n = 13, \log_2 qL = 141)</td>
<td>192/156.40 (601.40)</td>
<td>(\rho_{\text{stat}}^2)</td>
<td>601.40 bikz (\approx) 156.40 bits</td>
</tr>
<tr>
<td>(\log_2 n = 13, \log_2 qL = 100)</td>
<td>256/224.80 (848.63)</td>
<td>(\rho_{\text{stat}}^2)</td>
<td>848.63 bikz (\approx) 224.80 bits</td>
</tr>
<tr>
<td>(\log_2 n = 14, \log_2 qL = 411)</td>
<td>128/93.37 (352.34)</td>
<td>(\rho_{\text{stat}}^2)</td>
<td>352.34 bikz (\approx) 93.37 bits</td>
</tr>
<tr>
<td>(\log_2 n = 14, \log_2 qL = 284)</td>
<td>192/157.62 (594.78)</td>
<td>(\rho_{\text{stat}}^2)</td>
<td>594.78 bikz (\approx) 157.62 bits</td>
</tr>
<tr>
<td>(\log_2 n = 14, \log_2 qL = 229)</td>
<td>256/222.42 (839.32)</td>
<td>(\rho_{\text{stat}}^2)</td>
<td>839.32 bikz (\approx) 222.42 bits</td>
</tr>
<tr>
<td>(\log_2 n = 15, \log_2 qL = 827)</td>
<td>128/92.37 (348.55)</td>
<td>(\rho_{\text{stat}}^2)</td>
<td>348.55 bikz (\approx) 92.37 bits</td>
</tr>
<tr>
<td>(\log_2 n = 15, \log_2 qL = 577)</td>
<td>192/156.35 (594.00)</td>
<td>(\rho_{\text{stat}}^2)</td>
<td>594.00 bikz (\approx) 156.35 bits</td>
</tr>
<tr>
<td>(\log_2 n = 15, \log_2 qL = 443)</td>
<td>256/222.52 (832.15)</td>
<td>(\rho_{\text{stat}}^2)</td>
<td>832.15 bikz (\approx) 222.52 bits</td>
</tr>
<tr>
<td>(\log_2 n = 17, \log_2 qL = 2400)</td>
<td>140/145.88 (550.51)</td>
<td>(\rho_{\text{stat}}^2)</td>
<td>550.51 bikz (\approx) 145.88 bits</td>
</tr>
<tr>
<td>(\log_2 n = 17, \log_2 qL = 2000)</td>
<td>163/187.40 (707.17)</td>
<td>(\rho_{\text{stat}}^2)</td>
<td>707.17 bikz (\approx) 187.40 bits</td>
</tr>
</tbody>
</table>

Fig. 9: Concrete security of lattice reduction attacks after observing 1000 decryptions of Class 2 ciphertexts.
<table>
<thead>
<tr>
<th>Parameter Set</th>
<th>Original Noise Variance (t)</th>
<th>Final Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \log_2 n = 10, \log_2 q_L = 25 )</td>
<td>128/102 100 ( \cdot \rho_{C_1}^2 ) 1000</td>
<td>352 bikz ( \approx 93 ) bits</td>
</tr>
<tr>
<td>( \rho_{C_1}^2 )</td>
<td>1000</td>
<td>208 bikz ( \approx 79 ) bits</td>
</tr>
<tr>
<td>( t \cdot \rho_{C_1}^2 )</td>
<td>1000</td>
<td>376 bikz ( \approx 100 ) bits</td>
</tr>
<tr>
<td>( \log_2 n = 10, \log_2 q_L = 17 )</td>
<td>192/170 100 ( \cdot \rho_{C_1}^2 ) 1000</td>
<td>568 bikz ( \approx 150 ) bits</td>
</tr>
<tr>
<td>( \rho_{C_1}^2 )</td>
<td>1000</td>
<td>619 bikz ( \approx 164 ) bits</td>
</tr>
<tr>
<td>( t \cdot \rho_{C_1}^2 )</td>
<td>1000</td>
<td>619 bikz ( \approx 164 ) bits</td>
</tr>
<tr>
<td>( \log_2 n = 10, \log_2 q_L = 13 )</td>
<td>256/234 100 ( \cdot \rho_{C_1}^2 ) 1000</td>
<td>764 bikz ( \approx 202 ) bits</td>
</tr>
<tr>
<td>( \rho_{C_1}^2 )</td>
<td>1000</td>
<td>847 bikz ( \approx 224 ) bits</td>
</tr>
<tr>
<td>( t \cdot \rho_{C_1}^2 )</td>
<td>1000</td>
<td>847 bikz ( \approx 224 ) bits</td>
</tr>
<tr>
<td>( \log_2 n = 11, \log_2 q_L = 51 )</td>
<td>128/97 100 ( \cdot \rho_{C_1}^2 ) 1000</td>
<td>348 bikz ( \approx 92 ) bits</td>
</tr>
<tr>
<td>( \rho_{C_1}^2 )</td>
<td>1000</td>
<td>360 bikz ( \approx 95 ) bits</td>
</tr>
<tr>
<td>( t \cdot \rho_{C_1}^2 )</td>
<td>1000</td>
<td>360 bikz ( \approx 95 ) bits</td>
</tr>
<tr>
<td>( \log_2 n = 11, \log_2 q_L = 35 )</td>
<td>192/162 100 ( \cdot \rho_{C_1}^2 ) 1000</td>
<td>575 bikz ( \approx 152 ) bits</td>
</tr>
<tr>
<td>( \rho_{C_1}^2 )</td>
<td>1000</td>
<td>601 bikz ( \approx 159 ) bits</td>
</tr>
<tr>
<td>( t \cdot \rho_{C_1}^2 )</td>
<td>1000</td>
<td>601 bikz ( \approx 159 ) bits</td>
</tr>
<tr>
<td>( \log_2 n = 11, \log_2 q_L = 27 )</td>
<td>256/226 100 ( \cdot \rho_{C_1}^2 ) 1000</td>
<td>790 bikz ( \approx 209 ) bits</td>
</tr>
<tr>
<td>( \rho_{C_1}^2 )</td>
<td>1000</td>
<td>834 bikz ( \approx 221 ) bits</td>
</tr>
<tr>
<td>( t \cdot \rho_{C_1}^2 )</td>
<td>1000</td>
<td>834 bikz ( \approx 221 ) bits</td>
</tr>
<tr>
<td>( \log_2 n = 12, \log_2 q_L = 101 )</td>
<td>128/97 100 ( \cdot \rho_{C_1}^2 ) 1000</td>
<td>356 bikz ( \approx 94 ) bits</td>
</tr>
<tr>
<td>( \rho_{C_1}^2 )</td>
<td>1000</td>
<td>363 bikz ( \approx 96 ) bits</td>
</tr>
<tr>
<td>( t \cdot \rho_{C_1}^2 )</td>
<td>1000</td>
<td>363 bikz ( \approx 96 ) bits</td>
</tr>
<tr>
<td>( \log_2 n = 12, \log_2 q_L = 70 )</td>
<td>192/161 100 ( \cdot \rho_{C_1}^2 ) 1000</td>
<td>589 bikz ( \approx 156 ) bits</td>
</tr>
<tr>
<td>( \rho_{C_1}^2 )</td>
<td>1000</td>
<td>603 bikz ( \approx 160 ) bits</td>
</tr>
<tr>
<td>( t \cdot \rho_{C_1}^2 )</td>
<td>1000</td>
<td>603 bikz ( \approx 160 ) bits</td>
</tr>
<tr>
<td>( \log_2 n = 12, \log_2 q_L = 54 )</td>
<td>256/227 100 ( \cdot \rho_{C_1}^2 ) 1000</td>
<td>823 bikz ( \approx 218 ) bits</td>
</tr>
<tr>
<td>( \rho_{C_1}^2 )</td>
<td>1000</td>
<td>847 bikz ( \approx 224 ) bits</td>
</tr>
<tr>
<td>( t \cdot \rho_{C_1}^2 )</td>
<td>1000</td>
<td>847 bikz ( \approx 224 ) bits</td>
</tr>
</tbody>
</table>

Fig. 10: 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 [17]. The third column indicates the noise-flooding noise added before returning the decryption to the adversary. \( \rho_{C_1}^2 \) 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 [17]) and bit-security.
<table>
<thead>
<tr>
<th>Parameter Set</th>
<th>Original Security</th>
<th>Noise Variance</th>
<th>Num Queries ((t))</th>
<th>Final Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\log_2 n = 13, \log_2 q_L = 202)</td>
<td>(128/96)</td>
<td>(100 \cdot \rho_{C1}^2)</td>
<td>1000</td>
<td>350 bikz (\approx) 93 bits</td>
</tr>
<tr>
<td>(\log_2 n = 13, \log_2 q_L = 141)</td>
<td>(192/159)</td>
<td>(100 \cdot \rho_{C1}^2)</td>
<td>1000</td>
<td>574 bikz (\approx) 152 bits</td>
</tr>
<tr>
<td>(\log_2 n = 13, \log_2 q_L = 109)</td>
<td>(256/225)</td>
<td>(100 \cdot \rho_{C1}^2)</td>
<td>1000</td>
<td>81 bikz (\approx) 212 bits</td>
</tr>
<tr>
<td>(\log_2 n = 14, \log_2 q_L = 411)</td>
<td>(128/93)</td>
<td>(100 \cdot \rho_{C1}^2)</td>
<td>1000</td>
<td>352 bikz (\approx) 93 bits</td>
</tr>
<tr>
<td>(\log_2 n = 14, \log_2 q_L = 284)</td>
<td>(192/158)</td>
<td>(100 \cdot \rho_{C1}^2)</td>
<td>1000</td>
<td>590 bikz (\approx) 154 bits</td>
</tr>
<tr>
<td>(\log_2 n = 14, \log_2 q_L = 220)</td>
<td>(256/222)</td>
<td>(100 \cdot \rho_{C1}^2)</td>
<td>1000</td>
<td>831 bikz (\approx) 214 bits</td>
</tr>
<tr>
<td>(\log_2 n = 15, \log_2 q_L = 827)</td>
<td>(128/92)</td>
<td>(100 \cdot \rho_{C1}^2)</td>
<td>1000</td>
<td>347 bikz (\approx) 92 bits</td>
</tr>
<tr>
<td>(\log_2 n = 15, \log_2 q_L = 571)</td>
<td>(192/156)</td>
<td>(100 \cdot \rho_{C1}^2)</td>
<td>1000</td>
<td>588 bikz (\approx) 156 bits</td>
</tr>
<tr>
<td>(\log_2 n = 15, \log_2 q_L = 443)</td>
<td>(256/220)</td>
<td>(100 \cdot \rho_{C1}^2)</td>
<td>1000</td>
<td>831 bikz (\approx) 216 bits</td>
</tr>
<tr>
<td>(\log_2 n = 17, \log_2 q_L = 2400)</td>
<td>(140/146)</td>
<td>(100 \cdot \rho_{C1}^2)</td>
<td>1000</td>
<td>551 bikz (\approx) 146 bits</td>
</tr>
<tr>
<td>(\log_2 n = 17, \log_2 q_L = 2000)</td>
<td>(193/187)</td>
<td>(100 \cdot \rho_{C1}^2)</td>
<td>1000</td>
<td>706 bikz (\approx) 187 bits</td>
</tr>
</tbody>
</table>

Fig. 10: 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 [17]. 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 [17]) and bit-security.
<table>
<thead>
<tr>
<th>Parameter Set</th>
<th>Original Security</th>
<th>Noise Variance</th>
<th>Num Queries (t)</th>
<th>Final Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\log_2 n = 10$, $\log_2 qL = 25$</td>
<td>128/102</td>
<td>$\rho_{C2}^2$</td>
<td>1000</td>
<td>302 bikz $\approx$ 80 bits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$t \cdot \rho_{C2}^2$</td>
<td>1000</td>
<td>354 bikz $\approx$ 94 bits</td>
</tr>
<tr>
<td>$\log_2 n = 10$, $\log_2 qL = 17$</td>
<td>192/170</td>
<td>$\rho_{C2}^2$</td>
<td>1000</td>
<td>377 bikz $\approx$ 100 bits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$t \cdot \rho_{C2}^2$</td>
<td>1000</td>
<td>568 bikz $\approx$ 150 bits</td>
</tr>
<tr>
<td>$\log_2 n = 10$, $\log_2 qL = 13$</td>
<td>256/234</td>
<td>$\rho_{C2}^2$</td>
<td>1000</td>
<td>772 bikz $\approx$ 205 bits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$t \cdot \rho_{C2}^2$</td>
<td>1000</td>
<td>851 bikz $\approx$ 226 bits</td>
</tr>
<tr>
<td>$\log_2 n = 11$, $\log_2 qL = 51$</td>
<td>128/97</td>
<td>$\rho_{C2}^2$</td>
<td>1000</td>
<td>321 bikz $\approx$ 85 bits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$t \cdot \rho_{C2}^2$</td>
<td>1000</td>
<td>349 bikz $\approx$ 93 bits</td>
</tr>
<tr>
<td>$\log_2 n = 11$, $\log_2 qL = 35$</td>
<td>192/162</td>
<td>$\rho_{C2}^2$</td>
<td>1000</td>
<td>361 bikz $\approx$ 96 bits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$t \cdot \rho_{C2}^2$</td>
<td>1000</td>
<td>577 bikz $\approx$ 153 bits</td>
</tr>
<tr>
<td>$\log_2 n = 11$, $\log_2 qL = 27$</td>
<td>256/226</td>
<td>$\rho_{C2}^2$</td>
<td>1000</td>
<td>794 bikz $\approx$ 210 bits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$t \cdot \rho_{C2}^2$</td>
<td>1000</td>
<td>836 bikz $\approx$ 222 bits</td>
</tr>
<tr>
<td>$\log_2 n = 12$, $\log_2 qL = 101$</td>
<td>128/97</td>
<td>$\rho_{C2}^2$</td>
<td>1000</td>
<td>342 bikz $\approx$ 91 bits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$t \cdot \rho_{C2}^2$</td>
<td>1000</td>
<td>357 bikz $\approx$ 95 bits</td>
</tr>
<tr>
<td>$\log_2 n = 12$, $\log_2 qL = 70$</td>
<td>192/161</td>
<td>$\rho_{C2}^2$</td>
<td>1000</td>
<td>363 bikz $\approx$ 96 bits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$t \cdot \rho_{C2}^2$</td>
<td>1000</td>
<td>591 bikz $\approx$ 157 bits</td>
</tr>
<tr>
<td>$\log_2 n = 12$, $\log_2 qL = 54$</td>
<td>256/227</td>
<td>$\rho_{C2}^2$</td>
<td>1000</td>
<td>848 bikz $\approx$ 225 bits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$t \cdot \rho_{C2}^2$</td>
<td>1000</td>
<td>825 bikz $\approx$ 219 bits</td>
</tr>
</tbody>
</table>

Fig. 11: 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 [17]. 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 final column provides the reduced security level after the attack in terms of bikz (see [17]) and bit-security. The (encoded) message magnitude is equal to $n \cdot \sqrt{\ell/3}$ in all rows, where $\ell$ is set to 20.
<table>
<thead>
<tr>
<th>Parameter Set</th>
<th>Original Security</th>
<th>Noise Variance</th>
<th>Queries ($t$)</th>
<th>Final Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\log_2 n = 13, \log_2 q L = 202$</td>
<td>$\rho^2_{C_2}$</td>
<td>1000</td>
<td>351 bikz $\approx$ 93 bits</td>
<td></td>
</tr>
<tr>
<td>$\log_2 n = 13, \log_2 q L = 141$</td>
<td>$\rho^2_{C_2}$</td>
<td>1000</td>
<td>362 bikz $\approx$ 96 bits</td>
<td></td>
</tr>
<tr>
<td>$\log_2 n = 13, \log_2 q L = 109$</td>
<td>$\rho^2_{C_2}$</td>
<td>1000</td>
<td>575 bikz $\approx$ 152 bits</td>
<td></td>
</tr>
<tr>
<td>$\log_2 n = 14, \log_2 q L = 411$</td>
<td>$\rho^2_{C_2}$</td>
<td>1000</td>
<td>346 bikz $\approx$ 92 bits</td>
<td></td>
</tr>
<tr>
<td>$\log_2 n = 14, \log_2 q L = 284$</td>
<td>$\rho^2_{C_2}$</td>
<td>1000</td>
<td>350 bikz $\approx$ 93 bits</td>
<td></td>
</tr>
<tr>
<td>$\log_2 n = 14, \log_2 q L = 220$</td>
<td>$\rho^2_{C_2}$</td>
<td>1000</td>
<td>351 bikz $\approx$ 93 bits</td>
<td></td>
</tr>
<tr>
<td>$\log_2 n = 15, \log_2 q L = 827$</td>
<td>$\rho^2_{C_2}$</td>
<td>1000</td>
<td>348 bikz $\approx$ 92 bits</td>
<td></td>
</tr>
<tr>
<td>$\log_2 n = 15, \log_2 q L = 571$</td>
<td>$\rho^2_{C_2}$</td>
<td>1000</td>
<td>348 bikz $\approx$ 92 bits</td>
<td></td>
</tr>
<tr>
<td>$\log_2 n = 15, \log_2 q L = 443$</td>
<td>$\rho^2_{C_2}$</td>
<td>1000</td>
<td>348 bikz $\approx$ 92 bits</td>
<td></td>
</tr>
<tr>
<td>$\log_2 n = 17, \log_2 q L = 2400$</td>
<td>$\rho^2_{C_2}$</td>
<td>1000</td>
<td>549 bikz $\approx$ 145 bits</td>
<td></td>
</tr>
<tr>
<td>$\log_2 n = 17, \log_2 q L = 2000$</td>
<td>$\rho^2_{C_2}$</td>
<td>1000</td>
<td>707 bikz $\approx$ 187 bits</td>
<td></td>
</tr>
</tbody>
</table>

*Fig. 11: 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 [17]. The third column indicates the noise-flooding noise added before returning the decryption to the adversary. $\rho^2_{C_2}$ 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 [17]) and bit-security. The (encoded) message magnitude is equal to $n \cdot \sqrt{\ell / 3}$ in all rows, where $\ell$ is set to 20.
<table>
<thead>
<tr>
<th>Parameter Set</th>
<th>Orig Security</th>
<th>Noise Var</th>
<th>Num Queries</th>
<th>Succ Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\log_2 n = 10, \log_2 q_L = 25$</td>
<td>128/102</td>
<td>$\rho_{C1}^2$</td>
<td>77</td>
<td>0.81</td>
</tr>
<tr>
<td>$\log_2 n = 10, \log_2 q_L = 17$</td>
<td>192/170</td>
<td>$\rho_{C1}^2$</td>
<td>77</td>
<td>0.81</td>
</tr>
<tr>
<td>$\log_2 n = 10, \log_2 q_L = 13$</td>
<td>256/234</td>
<td>$\rho_{C1}^2$</td>
<td>77</td>
<td>0.81</td>
</tr>
<tr>
<td>$\log_2 n = 11, \log_2 q_L = 51$</td>
<td>128/97</td>
<td>$\rho_{C1}^2$</td>
<td>82</td>
<td>0.82</td>
</tr>
<tr>
<td>$\log_2 n = 11, \log_2 q_L = 35$</td>
<td>192/162</td>
<td>$\rho_{C1}^2$</td>
<td>82</td>
<td>0.82</td>
</tr>
<tr>
<td>$\log_2 n = 11, \log_2 q_L = 27$</td>
<td>256/226</td>
<td>$\rho_{C1}^2$</td>
<td>82</td>
<td>0.82</td>
</tr>
<tr>
<td>$\log_2 n = 12, \log_2 q_L = 101$</td>
<td>128/97</td>
<td>$\rho_{C1}^2$</td>
<td>86</td>
<td>0.80</td>
</tr>
<tr>
<td>$\log_2 n = 12, \log_2 q_L = 70$</td>
<td>192/161</td>
<td>$\rho_{C1}^2$</td>
<td>86</td>
<td>0.80</td>
</tr>
<tr>
<td>$\log_2 n = 12, \log_2 q_L = 54$</td>
<td>256/227</td>
<td>$\rho_{C1}^2$</td>
<td>86</td>
<td>0.80</td>
</tr>
</tbody>
</table>

Fig. 12: 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 [17]. 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.
<table>
<thead>
<tr>
<th>Parameter Set</th>
<th>Orig Security</th>
<th>Noise Var</th>
<th>Num Queries</th>
<th>Succ Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \log_2 n = 13, \log_2 q_L = 202 )</td>
<td>128/96</td>
<td>( \rho_{C1}^2 )</td>
<td>91</td>
<td>0.81</td>
</tr>
<tr>
<td>( \log_2 n = 13, \log_2 q_L = 141 )</td>
<td>192/159</td>
<td>( 100 \cdot \rho_{C1}^2 )</td>
<td>4930</td>
<td>0.80</td>
</tr>
<tr>
<td>( \log_2 n = 13, \log_2 q_L = 109 )</td>
<td>256/225</td>
<td>( 100 \cdot \rho_{C1}^2 )</td>
<td>4930</td>
<td>0.80</td>
</tr>
<tr>
<td>( \log_2 n = 14, \log_2 q_L = 411 )</td>
<td>128/93</td>
<td>( \rho_{C1}^2 )</td>
<td>96</td>
<td>0.82</td>
</tr>
<tr>
<td>( \log_2 n = 14, \log_2 q_L = 284 )</td>
<td>192/158</td>
<td>( 100 \cdot \rho_{C1}^2 )</td>
<td>5290</td>
<td>0.80</td>
</tr>
<tr>
<td>( \log_2 n = 14, \log_2 q_L = 220 )</td>
<td>256/222</td>
<td>( 100 \cdot \rho_{C1}^2 )</td>
<td>5290</td>
<td>0.80</td>
</tr>
<tr>
<td>( \log_2 n = 15, \log_2 q_L = 827 )</td>
<td>128/92</td>
<td>( \rho_{C1}^2 )</td>
<td>100</td>
<td>0.80</td>
</tr>
<tr>
<td>( \log_2 n = 15, \log_2 q_L = 571 )</td>
<td>192/156</td>
<td>( 100 \cdot \rho_{C1}^2 )</td>
<td>5660</td>
<td>0.80</td>
</tr>
<tr>
<td>( \log_2 n = 15, \log_2 q_L = 443 )</td>
<td>256/220</td>
<td>( 100 \cdot \rho_{C1}^2 )</td>
<td>5660</td>
<td>0.80</td>
</tr>
<tr>
<td>( \log_2 n = 17, \log_2 q_L = 2400 )</td>
<td>140/146</td>
<td>( \rho_{C1}^2 )</td>
<td>6420</td>
<td>0.80</td>
</tr>
<tr>
<td>( \log_2 n = 17, \log_2 q_L = 2000 )</td>
<td>193/187</td>
<td>( 100 \cdot \rho_{C1}^2 )</td>
<td>6420</td>
<td>0.80</td>
</tr>
</tbody>
</table>

Fig. 12: **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 [17]. 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.
Fig. 13: 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 [17]. The third column indicates the noise-flooding noise added before returning the decryption to the adversary. \( \rho_C^2 \) 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.
Fig. 13: 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 [17]. 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 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.
<table>
<thead>
<tr>
<th>Parameter Set</th>
<th>Orig. Security</th>
<th>Noise Var</th>
<th>Num. Queries</th>
<th>Num. Succ</th>
<th>Final Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \log_2 n = 10, \log_2 q_L = 25 )</td>
<td>128/102</td>
<td>475</td>
<td>531</td>
<td>0.80</td>
<td>72 bikz ( \approx ) 19 bits</td>
</tr>
<tr>
<td>( \log_2 n = 10, \log_2 q_L = 17 )</td>
<td>192/170</td>
<td>350</td>
<td>531</td>
<td>0.80</td>
<td>150 bikz ( \approx ) 40 bits</td>
</tr>
<tr>
<td>( \log_2 n = 10, \log_2 q_L = 13 )</td>
<td>256/234</td>
<td>475</td>
<td>531</td>
<td>0.80</td>
<td>273 bikz ( \approx ) 72 bits</td>
</tr>
<tr>
<td>( \log_2 n = 11, \log_2 q_L = 51 )</td>
<td>128/97</td>
<td>75</td>
<td>825</td>
<td>0.80</td>
<td>178 bikz ( \approx ) 47 bits</td>
</tr>
<tr>
<td>( \log_2 n = 11, \log_2 q_L = 35 )</td>
<td>192/162</td>
<td>350</td>
<td>534</td>
<td>0.80</td>
<td>231 bikz ( \approx ) 61 bits</td>
</tr>
<tr>
<td>( \log_2 n = 11, \log_2 q_L = 27 )</td>
<td>256/226</td>
<td>75</td>
<td>825</td>
<td>0.80</td>
<td>395 bikz ( \approx ) 105 bits</td>
</tr>
<tr>
<td>( \log_2 n = 12, \log_2 q_L = 101 )</td>
<td>128/97</td>
<td>75</td>
<td>827</td>
<td>0.80</td>
<td>394 bikz ( \approx ) 118 bits</td>
</tr>
<tr>
<td>( \log_2 n = 12, \log_2 q_L = 70 )</td>
<td>192/161</td>
<td>350</td>
<td>535</td>
<td>0.80</td>
<td>836 bikz ( \approx ) 157 bits</td>
</tr>
<tr>
<td>( \log_2 n = 12, \log_2 q_L = 54 )</td>
<td>256/227</td>
<td>75</td>
<td>827</td>
<td>0.80</td>
<td>615 bikz ( \approx ) 105 bits</td>
</tr>
</tbody>
</table>

Fig. 14: 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 [17]. The third column indicates the noise-flooding noise added before returning the decryption to the adversary. \( \rho^2_{C_1} \) 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 [17]) and bit-security.
<table>
<thead>
<tr>
<th>Parameter Set</th>
<th>Orig. Security</th>
<th>Noise Var</th>
<th>Num Queries</th>
<th>Num Guess</th>
<th>Num Succ</th>
<th>Final Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \log_2 n = 13, \log_2 q_L = 202 )</td>
<td>128/96</td>
<td>( \rho^2_{C_1} )</td>
<td>75</td>
<td>100 \cdot \rho^2_{C_1}</td>
<td>3500</td>
<td>828</td>
</tr>
<tr>
<td>( \log_2 n = 13, \log_2 q_L = 141 )</td>
<td>192/159</td>
<td>( \rho^2_{C_1} )</td>
<td>75</td>
<td>100 \cdot \rho^2_{C_1}</td>
<td>3500</td>
<td>828</td>
</tr>
<tr>
<td>( \log_2 n = 13, \log_2 q_L = 109 )</td>
<td>256/225</td>
<td>( \rho^2_{C_1} )</td>
<td>75</td>
<td>100 \cdot \rho^2_{C_1}</td>
<td>3500</td>
<td>828</td>
</tr>
<tr>
<td>( \log_2 n = 14, \log_2 q_L = 411 )</td>
<td>128/93</td>
<td>( \rho^2_{C_1} )</td>
<td>75</td>
<td>100 \cdot \rho^2_{C_1}</td>
<td>3500</td>
<td>828</td>
</tr>
<tr>
<td>( \log_2 n = 14, \log_2 q_L = 284 )</td>
<td>192/158</td>
<td>( \rho^2_{C_1} )</td>
<td>75</td>
<td>100 \cdot \rho^2_{C_1}</td>
<td>3500</td>
<td>828</td>
</tr>
<tr>
<td>( \log_2 n = 14, \log_2 q_L = 220 )</td>
<td>256/222</td>
<td>( \rho^2_{C_1} )</td>
<td>75</td>
<td>100 \cdot \rho^2_{C_1}</td>
<td>3500</td>
<td>828</td>
</tr>
<tr>
<td>( \log_2 n = 15, \log_2 q_L = 827 )</td>
<td>128/92</td>
<td>( \rho^2_{C_1} )</td>
<td>75</td>
<td>100 \cdot \rho^2_{C_1}</td>
<td>3500</td>
<td>829</td>
</tr>
<tr>
<td>( \log_2 n = 15, \log_2 q_L = 571 )</td>
<td>192/156</td>
<td>( \rho^2_{C_1} )</td>
<td>75</td>
<td>100 \cdot \rho^2_{C_1}</td>
<td>3500</td>
<td>829</td>
</tr>
<tr>
<td>( \log_2 n = 15, \log_2 q_L = 443 )</td>
<td>256/220</td>
<td>( \rho^2_{C_1} )</td>
<td>75</td>
<td>100 \cdot \rho^2_{C_1}</td>
<td>3500</td>
<td>829</td>
</tr>
<tr>
<td>( \log_2 n = 17, \log_2 q_L = 2400 )</td>
<td>140/146</td>
<td>( \rho^2_{C_1} )</td>
<td>85</td>
<td>100 \cdot \rho^2_{C_1}</td>
<td>4000</td>
<td>1722</td>
</tr>
<tr>
<td>( \log_2 n = 17, \log_2 q_L = 2000 )</td>
<td>193/187</td>
<td>( \rho^2_{C_1} )</td>
<td>85</td>
<td>100 \cdot \rho^2_{C_1}</td>
<td>4000</td>
<td>1722</td>
</tr>
</tbody>
</table>

Fig. 14: 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 [17]. The third column indicates the noise-flooding noise added before returning the decryption to the adversary. \( \rho^2_{C_1} \) 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 [17]) and bit-security.
### Fig. 15: 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 [17]. The third column indicates the noise-flooding noise added before returning the decryption to the adversary. $\rho_{2}^{2}$ 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 [17]) and bit-security. The (encoded) message magnitude is equal to $n \cdot \sqrt{\ell/3}$ in all rows, where $\ell$ is set to 20.

<table>
<thead>
<tr>
<th>Parameter Set</th>
<th>Orig Security</th>
<th>Noise Var</th>
<th>Num Queries</th>
<th>Num Guess</th>
<th>Succ Prob</th>
<th>Final Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\log_2 n = 10, \log_2 q_L = 25$</td>
<td>128/102</td>
<td>$\rho_{2}^{2}$</td>
<td>95</td>
<td>892</td>
<td>0.80</td>
<td>53 bikz $\approx 14$ bits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$100 \cdot \rho_{2}^{2}$</td>
<td>4300</td>
<td>620</td>
<td>0.80</td>
<td>125 bikz $\approx 33$ bits</td>
</tr>
<tr>
<td>$\log_2 n = 10, \log_2 q_L = 17$</td>
<td>192/170</td>
<td>$\rho_{2}^{2}$</td>
<td>95</td>
<td>892</td>
<td>0.80</td>
<td>134 bikz $\approx 36$ bits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$100 \cdot \rho_{2}^{2}$</td>
<td>4300</td>
<td>620</td>
<td>0.80</td>
<td>235 bikz $\approx 62$ bits</td>
</tr>
<tr>
<td>$\log_2 n = 10, \log_2 q_L = 13$</td>
<td>256/234</td>
<td>$\rho_{2}^{2}$</td>
<td>95</td>
<td>892</td>
<td>0.80</td>
<td>216 bikz $\approx 57$ bits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$100 \cdot \rho_{2}^{2}$</td>
<td>4300</td>
<td>620</td>
<td>0.80</td>
<td>345 bikz $\approx 91$ bits</td>
</tr>
<tr>
<td>$\log_2 n = 11, \log_2 q_L = 51$</td>
<td>128/97</td>
<td>$\rho_{2}^{2}$</td>
<td>95</td>
<td>896</td>
<td>0.80</td>
<td>166 bikz $\approx 44$ bits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$100 \cdot \rho_{2}^{2}$</td>
<td>4300</td>
<td>622</td>
<td>0.80</td>
<td>214 bikz $\approx 57$ bits</td>
</tr>
<tr>
<td>$\log_2 n = 11, \log_2 q_L = 35$</td>
<td>192/162</td>
<td>$\rho_{2}^{2}$</td>
<td>95</td>
<td>896</td>
<td>0.80</td>
<td>297 bikz $\approx 79$ bits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$100 \cdot \rho_{2}^{2}$</td>
<td>4300</td>
<td>622</td>
<td>0.80</td>
<td>370 bikz $\approx 98$ bits</td>
</tr>
<tr>
<td>$\log_2 n = 11, \log_2 q_L = 27$</td>
<td>256/226</td>
<td>$\rho_{2}^{2}$</td>
<td>95</td>
<td>896</td>
<td>0.80</td>
<td>424 bikz $\approx 122$ bits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$100 \cdot \rho_{2}^{2}$</td>
<td>4300</td>
<td>622</td>
<td>0.80</td>
<td>521 bikz $\approx 138$ bits</td>
</tr>
<tr>
<td>$\log_2 n = 12, \log_2 q_L = 101$</td>
<td>128/97</td>
<td>$\rho_{2}^{2}$</td>
<td>95</td>
<td>897</td>
<td>0.80</td>
<td>251 bikz $\approx 67$ bits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$100 \cdot \rho_{2}^{2}$</td>
<td>4300</td>
<td>623</td>
<td>0.80</td>
<td>281 bikz $\approx 74$ bits</td>
</tr>
<tr>
<td>$\log_2 n = 12, \log_2 q_L = 70$</td>
<td>192/161</td>
<td>$\rho_{2}^{2}$</td>
<td>95</td>
<td>897</td>
<td>0.80</td>
<td>425 bikz $\approx 113$ bits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$100 \cdot \rho_{2}^{2}$</td>
<td>4300</td>
<td>623</td>
<td>0.80</td>
<td>471 bikz $\approx 125$ bits</td>
</tr>
<tr>
<td>$\log_2 n = 12, \log_2 q_L = 54$</td>
<td>256/227</td>
<td>$\rho_{2}^{2}$</td>
<td>95</td>
<td>897</td>
<td>0.80</td>
<td>601 bikz $\approx 159$ bits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$100 \cdot \rho_{2}^{2}$</td>
<td>4300</td>
<td>623</td>
<td>0.80</td>
<td>664 bikz $\approx 176$ bits</td>
</tr>
</tbody>
</table>
Fig. 15: 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 [17]. The third column indicates the noise-flooding noise added before returning the decryption to the adversary. $\rho_2^2$ 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 $L$ WE 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 [17]) and bit-security. The (encoded) message magnitude is equal to $n \cdot \sqrt{\ell/3}$ in all rows, where $\ell$ is set to 20.