Security Guidelines for Implementing Homomorphic Encryption

Jean-Philippe Bossuat[1], Rosario Cammarota[2], Jung Hee Cheon[3], Ilaria Chillotti, Benjamin R. Curtis[4(✉)], Wei Dai[5], Huijing Gong[2(✉)], Erin Hales[6], Duhyeong Kim[2], Bryan Kumara[7], Changmin Lee[8], Xianhui Lu[9], Carsten Maple[7,10], Alberto Pedrouzo-Ulloa[11], Rachel Player[6(✉)], Luis Antonio Ruiz Lopez[12], Yongsoo Song[3], Donggeon Yhee[13], and Bahattin Yildiz[14]

[1] jeanphilippe.bossuat@gmail.com
[2] Intel Labs
{rosario.cammarota, huijing.gong, duhyeong.kim}@intel.com
[3] Seoul National University
{jhcheon,y.song}@snu.ac.kr
[4] Zama
ben.curtis@zama.ai
[5] TikTok Inc.
weidai3141@gmail.com
[6] Royal Holloway, University of London
{erin.hales.2018@live.,rachel.player@}rhul.ac.uk
[7] The Alan Turing Institute
bkumara@turing.ac.uk
[8] Korea Institute for Advanced Study
changminlee@kias.re.kr
[9] Chinese Academy of Sciences
luxianhui@iie.ac.cn
[10] University of Warwick
CM@warwick.ac.uk
[11] atlanTTic, Universidade de Vigo
apedrouzo@gts.uvigo.es
[12] Lorica Cybersecurity
luis@loricacyber.com
[13] dgyhee@gmail.com
[14] LG Electronics
bahattin.yildiz@lge.com

Abstract. Fully Homomorphic Encryption (FHE) is a cryptographic primitive that allows performing arbitrary operations on encrypted data. Since the conception of the idea in [RAD78], it was considered a holy grail of cryptography. After the first construction in 2009 [Gen09], it has evolved to become a practical primitive with strong security guarantees. Most modern constructions are based on well-known lattice problems such as Learning with Errors (LWE). Besides its academic appeal, in recent years FHE has also attracted significant attention from industry, thanks to its applicability to a considerable number of real-world use-cases. An upcoming standardization effort by ISO/IEC aims to support the wider adoption of these techniques. However, one of the main challenges that standards bodies, developers, and end users usually encounter is establishing parameters. This is particularly hard in the case of FHE because the parameters are not only related to the security level of the system, but also to the type of operations that the system is able to handle. In this paper we provide examples of parameter sets for LWE targeting particular security levels, that can be used in the context of FHE constructions. We also give examples of complete FHE parameter sets, including the parameters relevant for correctness and performance, alongside those relevant for security. As an additional contribution, we survey the parameter selection support offered in open source FHE libraries.

✉ Corresponding authors

1 Introduction

An encryption scheme is said to be *fully homomorphic* if arbitrary computations can be conducted on encrypted inputs without knowledge of the decryption key, and thus without access to the plaintext input. From the time the first construction was proposed in [Gen09], there has been a significant effort to improve fully homomorphic encryption (FHE) schemes in terms of both efficiency and security. The study of its potential application started as early as [RAD78]. In fact, FHE supports many applications [KL21], including computation over data stored on private clouds [BY88], private information retrieval [MCR21], and secure inference [JVC18].

There has been significant academic and commercial effort towards developing real-world applications for FHE. As a result, a community initiative towards standardizing FHE called HomomorphicEncryption.org was launched in 2017. More recently, there is an ongoing effort to formally standardize FHE schemes by ISO/IEC. The schemes expected to be standardized are BFV [Bra12,FV12], BGV [BGV12], CKKS [CKKS17] and CGGI [CGGI16] with their variants. These FHE schemes are based on variants of the Learning with Errors (LWE) problem [Reg05], including Ring-LWE (RLWE) [SSTX09,LPR10] and General-LWE (GLWE) [BGV12,CGGI17].[15] To assess the concrete security of FHE schemes, we must therefore estimate the concrete hardness of the underlying variant of LWE. Every instance of RLWE and GLWE can be interpreted as an LWE instance. Moreover, it is not known how to cryptanalytically exploit the algebraic structures of RLWE and GLWE. For this reason, it is appropriate to restrict focus to the concrete security of LWE.

The purpose of this document is to support the ISO/IEC effort towards the standardization of FHE and its goal is two-fold. The first goal is to present LWE parameter sets that can be used in FHE implementations that target particular levels of security. These parameter sets are presented in Section 4.1. They are developed using the prevailing methodology to establish parameters for LWE-based cryptography, following works such as [APS15] and the Lattice Estimator[16]. We make available our code for estimating the security of these parameters sets at `https://github.com/gong-cr/FHE-Security-Guidelines/`.

Our second goal is to present examples of functional parameter sets that could be used for particular FHE schemes in different contexts. These parameter sets, presented in Section 4.2, mention not only those parameters that are relevant for security but also those relevant for correctness and performance. These parameter sets are necessarily exemplar and may not suit all implementations in all application contexts. Thus, in Section 4.3, we also survey the parameter selection support offered in open source FHE libraries.

1.1 Comparison to prior work [ACC+19]

Our approach builds upon the efforts in the prior work of HomomorphicEncryption.org [ACC+19], by updating and expanding the LWE parameter sets for FHE schemes that target specific levels of security. While their work provided valuable insights, it had certain limitations. Specifically, it did not consider parameter sets commonly used in schemes like [CGGI16] and similar ones. Additionally, it overlooked binary secret distributions, which are often used in practical applications. Furthermore, the LWE dimensions considered in [ACC+19] are limited to a range of $n = 1024$ to $n = 32768$, despite larger dimensions being employed in practice nowadays. Since currently there is no scientific evidence against including these parameter sets, we overcome these limitations in this document. In addition, the parameter sets provided in [ACC+19] may now be considered somewhat outdated, due to recent cryptanalytic advance-

[15] GLWE is also referred to as *Module* LWE (MLWE) in the literature [BGV12,LS15], but we will use the terminology "GLWE" in this document for consistency.

[16] `https://github.com/malb/lattice-estimator`.

ments that may have implications on the concrete hardness of LWE instances used in FHE applications [CHHS19,SC19,EJK20,GJ21,BLLW22,MAT22,CST22,DP23b,PS23,DP23a,XWW+24].

It is important to note that the goals of this document and [ACC+19] are different. In addition to presenting wider ranges of LWE parameter sets targeting specific levels of security, we also include functional parameter sets. These functional parameter sets offer examples of complete sets of parameters, rather than presenting only the parameters that are relevant for security. However, we would like to emphasize that the functional parameter tables provided are not exhaustive and should be viewed as examples. In addition, in contrast to [ACC+19], we do not provide details for any particular FHE construction or cryptanalytic attack. Instead, we encourage readers to consult the existing literature for detailed information on these aspects.

1.2 Related work

There are many other works in the literature on subjects that are similar to, but not directly addressed by, this document. Here we present an overview of these topics.

NTRU-based FHE. The NTRU problem [HPS98] can also be seen as a variant of RLWE and indeed is equivalent to RLWE for suitable parameters [SS11]. Several FHE schemes based on NTRU have been proposed [LTV12,BLLN13,Klu22,BIP+22,XZD+23]. However, it is known that the sublattice structure of the NTRU lattice can be used to optimize attacks [ABD16,CJL16,KF17,DvW21], leaving some NTRU-based FHE schemes insecure. Concretely, it was shown in [DvW21] that to avoid the sublattice attacks one should use modulus smaller than $O(n^{2.484})$. This seems to rule out the BGV/BFV-like NTRU-based FHE schemes that require large modulus (e.g., [LTV12]), but not CGGI-like NTRU-based schemes (e.g., [BIP+22]). As the NTRU-based schemes that are secure against the sublattice attacks are relatively new, they are not considered further in this work.

Reductions between LWE and other lattice problems. This document considers the hardness of LWE from the point of view of estimating the concrete security of specific LWE instances. The hardness of LWE can also be established by considering reductions between this and other lattice problems. It is known that solving LWE is at least as hard as quantumly [Reg05,Reg10], or classically [Pei09,BLP+13], solving worst-case lattice hard problems such as the decisional shortest vector problem (Gap-SVP) and the Shortest Independent Vectors Problem (SIVP). While these hardness proofs mainly focused on the case that the secret key is sampled from the uniform distribution, there are also reductions from LWE with uniform secret to LWE with some other secret key distributions, including the error distribution [ACPS09], a uniform binary distribution [BLP+13], and a sparse binary distribution [CHK+16]. RLWE (resp. GLWE) is proved to be at least as hard as worst-case lattice hard problems over ideal (resp. module) lattices [LPR10,PRSD17,LS15]. Algorithms for solving Ideal-SVP are considered in [CDPR16,PHS19,BL21].

Weak instances of RLWE. Although RLWE as originally defined is proved to be at least as hard as worst-case lattice hard problems over ideal lattices, there are variants with particular choices for quotient polynomial and modulus that have been shown to be weak [EHL14,ELOS15,CLS16,CIV16,Pei16]. The RLWE instances in this document are not weak in this sense.

Machine learning attacks. The line of work [WCCL22,LSW+23,LWA+23,SWL+24] shows how a transformer model may sometimes be used to recover secrets from LWE instances with sparse secrets in dimensions $n \leq 1024$ for relatively large modulus q. It is not clear whether the approach would be feasible or competitive for attacking LWE instances that are used in FHE, which would either use a much smaller modulus q than considered in [SWL+24] for $n \leq 1024$, or use a larger dimension n. Hence we do not consider this approach further.

Side channel attacks. Side-channel attacks exploit leakage gained from a specific implementation of an algorithm on a specific computer system, rather than weaknesses in the implemented algorithm itself. The discussion and mitigation of potential side-channel leakages in FHE is not considered in this document. We merely note that prior literature has exploited side channels in certain FHE implementations [PPM17,AKP+22,DP22,AA22], and that any potential side-channel leakage deserves attention since it can amplify the utility of algorithmic approaches for solving LWE [DDGR20,DGHK23].

IND-CPA$^{\mathsf{D}}$ security. The notion of IND-CPA$^{\mathsf{D}}$ security was introduced by Li and Micciancio [LM21] as a stronger assumption than IND-CPA security for schemes on approximate data. More recent work [CSBB24,CCP+24] has demonstrated that the IND-CPA$^{\mathsf{D}}$ security notion is applicable to schemes on exact data with non-negligible decryption failure probability, which includes existing instantiations of exact schemes. Developing approaches to ensure IND-CPA$^{\mathsf{D}}$ security is currently an active area of research. In this work we target IND-CPA security. We note that there may be application scenarios where IND-CPA$^{\mathsf{D}}$ is more appropriate, but we do not discuss this further.

Parameter selection. In Section 4.1 we present LWE parameter sets for FHE that target particular levels of security. Such sets could be used as part of an automatic parameter selection tool or compiler that considers functionality and efficiency alongside security. Approaches for automating the selection of FHE (or partial) parameters were given in e.g. [DKS+20,LHC+22,LCK+23,BBB+23,JCH23]. Similar such sets [ACC+19] have also been used in major FHE libraries to inform default parameters. We will mention this further in Section 4.3.

1.3 Structure of document

The remainder of this document is organised as follows. Section 2 introduces the LWE problem and its algebraic variants used in FHE schemes. Section 3 states the security levels that we target and describes the tools and assumptions that we use to give concrete security estimates of LWE parameter sets. Section 4.1 gives examples of LWE parameter sets chosen to target a given security level that can be used in FHE applications. Section 4.2 presents examples of complete FHE parameter sets. These parameters include the LWE parameters relevant to security, as well as other parameters (such as plaintext modulus) that are relevant for correctness and performance. Section 4.3 surveys the parameter selection support offered in open source FHE libraries.

2 Notation and definitions

In this section, we specify the notation used in the remainder of the document. We define the LWE, RLWE, and GLWE problems. We also specify the secret and error distributions that are used in practice.

Learning With Errors (LWE). The LWE problem is parametrized by $(n, m, q, \chi_{\mathbf{s}}, \chi_{\mathbf{e}})$, where n is the dimension, m is the number of available samples, q is the modulus, $\chi_{\mathbf{s}}$ is the secret distribution over $\mathbb{Z}_q^n$, and $\chi_{\mathbf{e}}$ is the error distribution over $\mathbb{Z}^m$.

Definition 1 (LWE **distribution**). *For a secret $\mathbf{s} \in \mathbb{Z}_q^n$ that is chosen according to $\chi_{\mathbf{s}}$, the* LWE distribution *samples $\mathbf{a} \in \mathbb{Z}_q^n$ uniformly at random, samples $e \in \mathbb{Z}$ from $\chi_{\mathbf{e}}$, computes $b := \mathbf{a} \cdot \mathbf{s} + e \bmod q$, and outputs $(\mathbf{a}, b)$.*

Definition 2 **(Decision** LWE**).** *The Decision* LWE *problem asks to decide whether samples $(\mathbf{a}, b)$ are from the* LWE *distribution or are chosen uniformly at random from $\mathbb{Z}_q^{n+1}$.*

Definition 3 **(Search** LWE**).** *The Search* LWE *problem asks to recover $\mathbf{s}$ (or equivalently $e_1, \ldots, e_m$) given m samples $\{(\mathbf{a}_i, b_i) : i = 1, \ldots, m\}$ from the* LWE *distribution.*

Ring Learning With Errors (RLWE).

Definition 4 (RLWE **distribution**). *Let $\mathcal{R}_q = \mathbb{Z}_q[X]/(f_N(x))$ be a polynomial ring with modulus q, where $f_N(x)$ is an irreducible polynomial of degree N. We often take a power-of-two cyclotomic ring so that N is a power of two and $f_N(x) = x^N + 1$. Let χ_s denote a secret distribution over $\mathcal{R}_q$, and let χ_e denote an error distribution over $\mathcal{R}_q$. For a secret $s \in \mathcal{R}_q$ that is chosen according to χ_s, the* RLWE distribution *samples $a \in \mathcal{R}_q$ uniformly, samples an error $e \in \mathcal{R}_q$ according to χ_e, computes $b := as + e \in \mathcal{R}_q$, and outputs (a, b).*

Definition 5 **(Decision** RLWE**).** *The Decision* RLWE *problem asks to decide whether samples (a, b) are from the* RLWE *distribution or are chosen uniformly at random from $\mathcal{R}_q \times \mathcal{R}_q$.*

Definition 6 **(Search** RLWE**).** *The Search* RLWE *problem asks to recover s given m samples $\{(a_i, b_i = a_i \cdot s + e_i) : i = 1, \ldots, m\}$ from the* RLWE *distribution.*

General Learning With Errors (GLWE).

Definition 7 (GLWE **distribution**). *We again let $\mathcal{R}_q$ be an (e.g. cyclotomic) polynomial ring with modulus q. We overload notation to let χ_s denote a secret distribution over $\mathcal{R}_q^k$, and to let χ_e denote an error distribution over $\mathcal{R}_q$. For a secret $\mathbf{s} \in \mathcal{R}_q^k$ that is chosen according to χ_s, sample $\mathbf{a} \in \mathcal{R}_q^k$ uniformly, and sample an error $e \in \mathcal{R}_q$ from χ_e. The* GLWE distribution *computes $b := \mathbf{a} \cdot \mathbf{s} + e \in \mathcal{R}_q$, and outputs $(\mathbf{a}, b)$.*

Definition 8 **(Decision** GLWE**).** *The Decision* GLWE *problem asks to decide whether samples $(\mathbf{a}, b)$ are from the* GLWE *distribution or are chosen uniformly at random from $\mathcal{R}_q^{k+1}$.*

Definition 9 **(Search** GLWE**).** *The Search* GLWE *problem asks to recover $\mathbf{s}$ given m samples $\{(\mathbf{a}_i, b_i) : i = 1, \ldots, m\}$ from the* GLWE *distribution.*

Error distributions. If the standard deviation of the error distribution is $\Omega(\sqrt{n})$, the best-known algorithm to solve the LWE problem requires exponential time [Reg10]. In practice, implementations of $\mathsf{RLWE}/\mathsf{GLWE}$-based homomorphic encryption schemes typically choose much narrower distributions. For RLWE-based schemes with an underlying power-of-two cyclotomic ring, each coordinate of the error polynomial is independently sampled from a Gaussian distribution centered at 0 with standard deviation σ. A very common choice is $\sigma \approx 3.2$ [ACC+19,HS20]. For RLWE-based schemes where the underlying ring is the k^{th} cyclotomic ring (where k is not a power of two), each coordinate of the error polynomial is sampled from Gaussian distribution centered at 0 with standard deviation $\sigma\sqrt{k}$ [HS20]. As an alternative, the FIPS 203 (draft) [oST23] makes use of a centered binomial distribution as the error distribution. For example, a centered binomial distribution resulted from 42 fair coin tosses centers at 0 and has standard deviation 3.24. Constant-time sampling from a centered binomial distribution can be more efficient than that from a discrete Gaussian distribution when σ is small.

Secret distributions. Various choices are used in practice for the secret key distribution. Below we list some examples.

- The coefficients of the secret polynomial s are chosen uniformly at random from $\mathbb{Z}_q$: this is known as *uniform secret*.
- The secret polynomial s is chosen according to the error distribution χ_e: this is known as *normal form secret*.
- The coefficients of the secret polynomial s are chosen uniformly at random from $\{-1, 0, 1\}$: this is known as *ternary secret*.
- The coefficients of the secret polynomial s are chosen uniformly at random from $\{0, 1\}$: this is known as *binary secret*.
- The coefficients of the secret polynomial s are chosen in $\{-1, 0, 1\}$ with a restriction that exactly h of them are 1 or -1, and the rest are all zeros: this is known as *fixed Hamming weight secret*. The exact method for sampling the nonzero entries may vary depending on the implementation.
- For a fixed Hamming weight secret such that the Hamming weight is small (e.g., $h < 0.25 \cdot n$), keys chosen from this distribution are called *sparse secret* keys. We discuss sparse secrets in the following subsection. The LWE parameter sets presented in this document do not have sparse secrets.

Sparse secrets. Sparse secrets were first used in homomorphic encryption to reduce the complexity of recryption, a part of bootstrapping [HS21]. For certain schemes, the multiplicative depth of bootstrapping depends on the Hamming weight of the secret key [CH18]. For others, the bootstrapping approach relates the Hamming weight of the secret key to the approximation interval of a sine function, and consequently this Hamming weight must be bounded and somewhat small for these algorithms [CHK+18,CCS19,HK20] (see also Appendix A). For these reasons, some implementations of BFV, BGV, and CKKS bootstrapping use sparse secret keys [CHK+18,CH18,CCS19,HK20] or temporarily switch the ciphertext to a sparse secret [BTPH22]. However, more recent works have achieved correct and efficient bootstrapping with non-sparse keys for CKKS [BMTPH21] and some instances of BFV [OPP23].

Reductions exist for the sparse secret variant of LWE, denoted as spLWE. The reduction [CHK+16] shows that spLWE can be reduced from standard LWE. However, the reduction is not sufficiently tight to provide useful insight into parameter setting.

Many attacks leverage properties of sparse secrets [NMW+24,HKLS22,May21,CHHS19,CP19,HG07]. Estimates of the cost of these attacks are not currently implemented in the Lattice Estimator, which is the tool used to estimate security in this work, so it is difficult to assess the security of parameter sets for which these attacks are applicable. Some of these attacks (e.g. [HKLS22,May21]) are not yet applicable to

FHE parameter sets. However, given the pace of development in this area, we expect future improvements in algorithms for solving spLWE.

3 Concrete security estimation

In this section we state the security levels that the parameter sets in Section 4.1 target, and we outline the assumptions under which we give estimates for the concrete security of those parameter sets.

3.1 Security Levels

We define three classical security levels, and corresponding quantum security levels according to Appendix A of FIPS 203 (draft) [oST23], as follows.

Category 128, 192, 256: Any algorithm that solves the underlying LWE instance must require (classical) computational resources comparable to or greater than those required for key search on a block cipher with a 128-bit, respectively 192-bit, respectively 256-bit key.

Category 128Q, 192Q, 256Q: Any algorithm that solves the underlying LWE instance must require quantum computational resources comparable to or greater than those required for key search on a block cipher with a 128-bit, respectively 192-bit, respectively 256-bit key.

3.2 The Lattice Estimator

We estimate concrete security of the FHE parameter sets given in Section 4.1 using the open-source Lattice Estimator tool [APS15]. The Lattice Estimator is widely used in estimating the security of FHE parameter sets [ACC+19] as well as more broadly in lattice-based cryptography.

Algorithms for solving LWE, that are currently supported in the Lattice Estimator, include the primal attack [BG14,ADPS16], the dual attack [MR09,Alb17,GJ21,MAT22], decoding attacks [LN13], Coded-BKW [GJS15,KF15], and algebraic algorithms [AG11,ACF+15]. Some combinatorial algorithms, including hybrid combinatorial and lattice algorithms [How07,ACW19,CHHS19,EJK20] are also supported.

However, it is important to note that some cryptanalytic algorithms applicable to LWE instances, including those typical of FHE applications, are not supported in the Lattice Estimator. This includes some combinatorial and hybrid approaches [May21,HKLS22,BLLW22,EGMS23]. Moreover, recent work suggests the success probability of the dual attack may be overestimated in some cases, which may impact the utility of the dual attack estimates in the Lattice Estimator [DP23b].

3.3 Lattice reduction algorithms and cost models

Since several of the algorithms for solving LWE rely on a lattice reduction subroutine (most commonly instantiated as BKZ), it is important to specify the cost model used for lattice reduction. There are several cost models available in the Lattice Estimator and there is not consensus in the literature as to a universally preferred cost model (see e.g. [ACD+18]). Following [ACC+19], our estimates for the security for the parameters presented in Section 4.1 are derived using the following cost models. To estimate the cost of BKZ with block size β in a lattice of dimension d, in the classical setting, we use:

$$T_{\mathsf{BKZ}}(\beta, d) = 8d \cdot 2^{0.292\beta+16.4}.$$

In the quantum setting, we use:

$$T_{\mathsf{BKZ}}(\beta, d) = 8d \cdot 2^{0.265\beta+16.4}.$$

To configure this in the Lattice Estimator, we set `RC.BDGL16` [BDGL16] as the cost model in the classical setting and `RC.LaaMosPol14` [LMvdP14] as the cost model in the quantum setting. We further set `red_shape_model = "gsa"` as the behaviour model for BKZ.

3.4 Computational cost metric

To assess whether we have met a target security level as defined in Section 3.1, we need to define a metric for the "computational resources". Multiple such metrics exist (see e.g. [ADPS16,ABD+20]) and their refinement is the subject of ongoing research. Since we use the Lattice Estimator to estimate the concrete cost of algorithms for solving LWE, we use the unit of computation used in the Estimator: "ring operations". That is, we will estimate that a particular parameter set meets Category 128 (respectively 128Q) if the Lattice Estimator estimates that all algorithms cost greater than 2^{128} ring operations when using a classical (respectively quantum) lattice reduction cost model. Note that "ring operations" can be converted into CPU cycles for classical computers.

4 Tables of parameters

In this section, we provide examples of parameter sets for FHE, targeting security (Section 4.1) and functionality (Section 4.2). We also review the parameter selection support offered in some of the major open-source FHE libraries. The notation used in Sections 4.1 and 4.2 is summarised in Table 4.1.

4.1 Parameter sets that target particular security levels

In this section, we give in Table 4.2 and 4.3 examples of LWE parameter sets that can be used in FHE applications. These LWE parameter sets target particular security levels as defined in Section 3.1 using the Lattice Estimator under the assumptions stated in Section 3.3 and 3.4. As such, the tables in this section are similar to those presented in [ACC+19]. The concrete security of the parameter sets is assessed by estimating the cost of primal_usvp, primal_bdd, hybrid_bdd, and hybrid_dual using commit 00ec72c of the Lattice Estimator.

Table 4.2 presents the maximal log (base 2) of the modulus q that can be used in dimension N, for Gaussian error distribution with standard deviation $\sigma = 3.19$, and for secret distributions that are either uniform ternary or Gaussian with standard deviation $\sigma = 3.19$, to give LWE parameter sets that target the Category 128 or 128Q, 192 or 192Q, 256 and 256Q security levels. This table is suitable in but not limited to the BFV/BGV/CKKS application settings where the error distribution standard deviation $\sigma = 3.19$ is typically fixed, but the modulus q can be varied.

In the CGGI setting, q is typically fixed to either 32-bit or 64-bit, and the error standard deviation can be varied. Thus, in Table 4.3, we present the minimal log (base 2) of the error distribution standard deviation σ, that can be used in dimension $n = k \cdot N$, for modulus q, and for secret distributions that are either uniform binary, uniform ternary, or Gaussian, to give LWE parameter sets that target the Category 128, 192, 256, 128Q, 192Q, or 256Q security levels.

Parameter	Definition
λ	Security level (classical or quantum) of the parameter set.
N	Dimension of the RLWE instance.
n	Dimension of the LWE instance, $n = kN$ when modelling GLWE.
q	LWE modulus. Largest ciphertext modulus for BGV, BFV, CGGI.
Q	Largest modulus of the ciphertext space, for CKKS.
P	Multiplication modulus for CKKS, with $q = PQ$ bounded according to security level.
t	BGV/BFV/CGGI plaintext modulus.
$\chi_{\mathbf{s}}$	Probability distribution of the LWE secret.
$\chi_{\mathbf{e}}$	Probability distribution of the error of a fresh LWE sample.
σ	Standard deviation of the LWE error distribution, also target standard deviation of the error distribution for ciphertexts after CKKS bootstrapping.
L	Level, number of maximal repeated multiplication supported.
Scaling Factor	CKKS scaling factor.
Base prime size	Number of significant bits for CKKS.
Precision Bit	Evaluated by logarithmically transforming the difference between results from standard (cleartext) calculation and those computed homomorphically.

Table 4.1: Notation used in Tables 4.2 4.3 4.4 4.5 4.6 and 4.7.

n	$\log_2(q)$ (Classical)		$\log_2(q)$ (Quantum)	
	Ternary	Gaussian	Ternary	Gaussian
		$\lambda = 128$		
1024	26	29	25	27
2048	54	56	50	52
4096	108	110	101	103
8192	217	219	203	205
16384	438	439	409	411
32768	881	883	825	827
65536	1776	1778	1663	1665
131072	3576	3578	3348	3351
		$\lambda = 192$		
2048	37	39	34	36
4096	75	77	70	72
8192	151	153	141	143
16384	304	306	283	285
32768	611	613	570	572
65536	1229	1230	1145	1147
131072	2469	2471	2302	2304
		$\lambda = 256$		
2048	28	30	26	28
4096	58	60	54	56
8192	117	119	109	111
16384	237	239	220	222
32768	475	477	442	444
65536	955	957	889	890
131072	1918	1920	1784	1786

Table 4.2: Maximal log (base 2) of the modulus q that can be used in dimension N, for Gaussian error distribution with standard deviation $\sigma = 3.19$, and for secret distributions $\chi_{\mathbf{s}}$ that are either uniform ternary or Gaussian with standard deviation $\sigma = 3.19$, to give LWE parameter sets that target the Category 128, 192, 256 (‘Classical’), 128Q, 192Q, or 256Q (‘Quantum’) security levels.

n	$\log_2(q)$	$\log_2(\sigma)$ (Classical)			$\log_2(\sigma)$ (Quantum)		
		Binary	Ternary	Gaussian	Binary	Ternary	Gaussian
$\lambda = 128$							
630	32	17.9	16.6	14.2	18.9	17.7	15.4
1024		7.6	6.3	4.5	9.2	8.0	6.3
≥ 2048		2.0	2.0	2.0	2.0	2.0	2.0
630	64	49.9	48.6	46.2	50.9	49.7	47.4
750		46.8	45.5	43.0	48.0	46.7	44.4
870		43.7	42.4	39.9	45.0	43.8	41.4
1024		39.6	38.3	36.1	41.2	40.0	37.9
2048		12.6	11.4	9.4	16.0	14.8	12.7
≥ 4096		2.0	2.0	2.0	2.0	2.0	2.0
$\lambda = 192$							
630	32	23.6	22.2	19.7	24.3	23.0	20.6
1024		16.3	15.0	12.4	17.5	16.2	13.8
≥ 2048		2.0	2.0	2.0	2.0	2.0	2.0
630	64	55.6	54.2	51.7	56.3	55.0	52.6
750		53.4	52.0	49.5	54.2	52.9	50.5
870		51.2	49.8	47.3	52.2	50.9	48.5
1024		48.3	47.0	44.4	49.5	48.2	45.8
2048		29.4	28.1	25.5	31.9	30.6	28.2
≥ 4096		2.0	2.0	2.0	2.0	2.0	2.0
$\lambda = 256$							
1024	32	21.0	19.6	16.9	21.9	20.6	18.1
2048		6.2	4.8	2.4	8.1	6.8	4.6
≥ 4096		2.0	2.0	2.0	2.0	2.0	2.0
1024	64	53.0	51.6	48.9	53.9	52.6	50.1
2048		38.2	36.8	34.2	40.1	38.8	36.3
4096		8.9	7.2	4.8	12.5	11.3	8.8
≥ 8192		2.0	2.0	2.0	2.0	2.0	2.0

Table 4.3: Minimal log (base 2) of the error distribution standard deviation σ, that can be used in dimension $n = kN$ and for secret distributions $\chi_{\mathbf{s}}$ that are either uniform binary, uniform ternary, or Gaussian with standard deviation $\sigma_s = 4$, to give LWE parameter sets that target the Category 128, 192, 256 ('Classical'), 128Q, 192Q or 256Q ('Quantum') security level. Since CGGI considers LWE ciphertexts, the dimension n is not restricted to a power of two, and therefore other values of n can be used (similarly, other values of q can be used). In both cases, we the value of $\log_2(\sigma)$ should be adapted accordingly.

4.2 Functional parameter sets

In this section, we give examples of SHE and FHE parameters sets that could be used for BGV, BFV, CGGI, or CKKS applications. These parameter sets include the LWE parameters relevant to security, as well as other parameters (such as plaintext modulus for BGV or BFV) that are relevant for correctness and performance.

Note that the parameter sets presented herein are intended as illustrative examples and may not necessarily represent optimal configurations to the individual libraries, and they are not intended for comparison among libraries.

Functional parameters for BGV and BFV. Table 4.4 provides examples of parameter sets for (RNS variants of) BGV/BFV in an SHE setting, i.e., without bootstrapping. In Table 4.4 there are parameters that are estimated to meet the Category 128, 192, 256, 128Q, 192Q or 256Q security levels. The parameters in Table 4.4 were generated[17] using Microsoft SEAL [SEA23]. The notation used is described in Table 4.1. Since BFV/BGV bootstrapping has seen a lot of recent developments and improvements [GV23,GIKV23,OPP23,Gee24,KSS24,KDE+24,MHWW24,LW24], we choose not to present example parameters for BFV/BGV with bootstrapping.

λ	128	192	256	128Q	192Q	256Q
$\log_2(N)$	14	15	16	14	15	16
$\log_2(q)$	424	585	920	391	562	880
$\log_2(t)$	20	20	20	20	20	20
χ_{s}	Ternary	Ternary	Ternary	Ternary	Ternary	Ternary
σ (χ_{e})	3.2	3.2	3.2	3.2	3.2	3.2
L (BFV)	10	14	23	9	13	22
L (BGV)	5	8	13	4	8	12

Table 4.4: Sample parameters for BFV/BGV without bootstrapping.

Sample parameters for CGGI. In Table 4.5 we present examples of parameters for CGGI that are estimated to meet the Category 128 security level. The notation used in Table 4.5 is as defined in Table 4.1, with the following additions: $(\chi_{\mathsf{LWE}}, \sigma_{\mathsf{LWE}})$ denote the secret key distribution, and the standard deviation of the Gaussian error used in LWE ciphertexts; $(\chi_{\mathsf{GLWE}}, \sigma_{\mathsf{GLWE}})$ denote the secret key distribution and the standard deviation of the Gaussian error used in GLWE ciphertexts; $(\beta_{\mathsf{ks}}, \ell_{\mathsf{ks}})$ denote the decomposition parameters used in key-switching keys; and $(\beta_{\mathsf{pbs}}, \ell_{\mathsf{pbs}})$ denote the decomposition parameters used in the bootstrapping keys. Finally, p_{error} denotes the error probability for a single bootstrapping operation. Parameters in Table 4.5 were generated using the optimization techniques found in Concrete [BBB+23].

[17] Table 4.4 can be reproduced using a script available at `https://github.com/WeiDaiWD/SEAL-Depth-Estimator`.

λ	128	128	128	128
n	742	777	630	512
$\log_2(N)$	11	9	10	10
k	1	3	1	1
q	2^{64}	2^{64}	2^{32}	2^{27} / 2^{14}
t	2^4	2	2	2
χ_{LWE}	Binary	Binary	Binary	Ternary
χ_{GLWE}	Binary	Binary	Binary	Ternary
β_{ks}	2^{23}	2^{18}	2^7	128
ℓ_{ks}	5	3	3	-
β_{pbs}	2^{23}	2^{18}	2^2	2^7
ℓ_{pbs}	1	1	8	-
σ_{LWE}	$2^{-17.11}$	$2^{-18.03}$	2^{-15}	3.2
σ_{GLWE}	$2^{-51.60}$	$2^{-38.08}$	2^{-25}	3.2
p_{error}	2^{-40}	2^{-40}	2^{-165}	2^{-52}

Table 4.5: Sample parameters for CGGI. The first two parameter sets (with $n = 742$ and 777) are parameter sets from the TFHE-rs library. The third parameter set (with $n = 630$) is from TFHElib, and the fourth parameter set (with $n = 512$) is taken from the parameters reccomended for OpenFHE in [MP21]. Note that the failure probabilities p_{error} are computed using varying techniques. Note that the parameter t, plaintext modulus, is sometimes referred to as p in the literature.

Sample parameters for RNS-CKKS. In Table 4.6, respectively Table 4.7, we present example parameter sets for (an RNS variant) of CKKS without, respectively with, bootstrapping. The parameters in Table 4.6 are estimated to meet the Category 128, 192, 256, 128Q, 192Q, 256Q levels of security. The parameters in Table 4.7 are estimated to meet the Category 128 and 192 levels of security.

The parameters in Table 4.6 were selected using OpenFHE v1.1.3 (commit `7b08ce1`) [BBB+22]. The parameters in Table 4.7 are selected[18] using Lattigo v5.0.2 [Tun23][19] for Set I and using OpenFHE v1.1.3 (commit `7b08ce1`) [BBB+22] for Sets II and III. The rescale method for OpenFHE is set to `FLEXIBLEAUTO`. Both libraries contain implementation of several bootstrapping algorithms, including [CHK+18,CCS19,HK20,BMTPH21,BCC+22].

The total cost in levels of CKKS bootstrapping can be broken down into several specific building blocks, with the most resource-intensive steps being: (1) CoeffsToSlots, (2) EvalMod and (3) SlotsToCoeffs. Table 4.7 provides the number of consumed levels for the execution of each of these blocks.

[18] Tables 4.6 and 4.7 can be reproduced using scripts available at `https://github.com/gong-cr/FHE-Security-Guidelines/`.
[19] Lattigo also provides support by default for the sparse secret encapsulation technique [BTPH22], but this feature was disabled to instead use a dense secret.
[20] Number of Slots refers to the number of complex numbers that are encrypted in each separate ciphertext.
[21] This scaling factor does not affect bootstrapping as Lattigo uses different independent internal scaling factors for each step of the bootstrapping circuit.
[22] Detailed explanation on this bootstrapping failure probability and the parameter K can be found in Appendix A.
[23] Following [BCC+22], Iterations corresponds to the number of repetitions applied to improve the final precision. Here, Iterations set to 1 means that no additional bootstrapping repetitions are applied.

λ	128	192	256	128Q	192Q	256Q
$\log_2(N)$	14	15	15	14	15	15
$\chi_{\mathbf{s}}$	Ternary	Ternary	Ternary	Ternary	Ternary	Ternary
σ ($\chi_{\mathbf{e}}$)	3.19	3.19	3.19	3.19	3.19	3.19
Base Prime Size	40	44	40	40	44	40
L	7	9	8	6	8	7
$\log_2(PQ)$	426	602	472	388	560	434
$\log_2(Q)$	306	422	352	268	380	313
$\log_2(P)$	120	180	120	120	180	120
$\log_2$ (Scaling Factor)	38	42	39	38	42	39
Precision Bit	25.1	26.7	22.4	24.0	28.2	23.5

Table 4.6: Sample parameters for RNS-CKKS without bootstrapping.

	Set I	Set II	Set III
λ	128	128	192
$\log_2(N)$	16	16	17
Number of Slots[20]	32768	32768	65536
$\chi_{\mathbf{s}}$	Ternary	Ternary	Ternary
σ ($\chi_{\mathbf{e}}$)	3.19	3.19	3.19
Base Prime Size	45	60	60
L (after bootstrapping)	11	5	14
$\log_2$(Scaling Factor)	35 [21]	55	55
$\log_2(PQ)$	1769	1750	2425
$\log_2(Q)$	1464	1270	1765
$\log_2(P)$	305	480	660
Level cost of SlotsToCoeffs	4	2	2
Level cost of EvalMod	12	13	13
$\log_2(\mathsf{Pr}[\|I(X)\| > K])$[22]	-37.65	-37.65	-11.66
K	512	512	512
Level cost of CoeffsToSlots	3	2	2
Iterations[23]	1	1	1
Precision Bit	15.9	9.4	7.5

Table 4.7: Sample parameters for RNS-CKKS with bootstrapping.

4.3 Parameter selection on open source libraries and compilers

Most FHE libraries lack a systematic process to select parameters for a desired application. However, external tools have been developed to help with this task for some of the most popular libraries. Table 4.8 lists some of the available open source FHE libraries and the schemes they support. In this section, we will overview parameter selection approaches in some of the major FHE libraries.

Library	Link	BFV	BGV	CKKS	CGGI	Note
blyss	`blyssprivacy/sdk`					Combines GSW and basic LWE.
Cingulata	`CEA-LIST/Cingulata`	✓				Also a compiler toolchain for its own BFV implementation and for TFHElib.
Cupcake	`facebookresearch/Cupcake`					Only implements of the additive version of BFV.
FHE-DECK	`FHE-Deck/fhe-deck-core`					Contains only the basics for RLWE and NTRU infrastructure.
FHELib	`Crypto-TII/fhelib`		✓			
HEaaN	`cryptolabinc/heaan`		✓	✓		Proprietary. Free for non-commercial usage.
HELib	`homenc/HElib`		✓	✓		
HEHub	`primihub/hehub`		✓	✓	✓	
HEU	`secretflow/heu`			✓	✓	Contains additive homomorphic encryption. FHE algorithms still in development.
Lattigo	`tuneinsight/lattigo`	✓	✓	✓	✓	
Liberate. FHE	`Desilo/liberate-fhe`			✓		
NFLLib	`quarkslab/NFLlib`	✓				
OpenFHE	`openfheorg`	✓	✓	✓	✓	
Parmesan	`crates/parmesan`					Builds on TFHE-rs.
Phantom	`encryptorion-lab/ phantom-fhe`	✓	✓	✓		
Poseidon	`luhang-HPU/Poseidon`	✓	✓	✓		
REDcuFHE	`TrustworthyComputing/ REDcuFHE`				✓	
SEAL	`microsoft/SEAL`	✓	✓	✓		
TFHE-rs	`zama-ai/tfhe-rs`				✓	
TFHElib	`tfhe/tfhe`				✓	

Table 4.8: Open source homomorphic encryption libraries and the algorithms they support.

OpenFHE. OpenFHE [BBB+22] supports the schemes BFV, BGV, CKKS, and the CGGI-like scheme FHEW. For each of BFV, BGV, and CKKS, the authors of the library provide a process to select parameters, depending on various factors such as desired security level, depth support, batch size, key-switching mechanism, etc. The library then finds[24] the appropriate parameters based on the tables in [ACC+19].

SEAL and EVA. Microsoft's SEAL [SEA23] supports BFV, BGV and CKKS. The main library does not have an elaborate system to find optimal parameters for the desired application. Nonetheless, it does provide[25] a list of upper bounds for the ciphertext modulus depending on the dimension of the ring, the desired security level and the distribution of the secret key. This list follows the tables from [ACC+19]. It is worth noting that SEAL uses, by default, a centered binomial distribution for the generation of LWE samples. Microsoft's EVA [DKS+20] is a compiler for homomorphic encryption built to work with the SEAL library. It contains a mechanism[26] to select an adequate decomposition of the ciphertext modulus depending on the desired application.

Lattigo. Tune Insight's Lattigo [Tun23] contains implementations of BFV, BGV and CKKS as well as support for the CGGI-like scheme FHEW. The library allows the user to set their own parameters, only providing a method to verify that the parameters are valid, i.e., that the parameters follow the hypotheses required for the construction to work and that they do not lead to a zero secret or error.

TFHE-rs and Concrete. Zama's TFHE-rs [Zam22b] implements a variant of the CGGI scheme. The library offers parameter sets for different configurations depending on the application. Zama's Concrete [Zam22a] is a compiler for CGGI built on top of THFE-rs. It contains an optimizing tool[27] to find appropriate parameters for a given FHE computation. It makes use of the Lattice Estimator to find the security level of the parameters.

HECATE and ELASM. Besides EVA, there have been other efforts proposing automatic scale management schemes through compilers. For instance, HECATE [LHC+22] and ELASM [LCK+23] target CKKS implementations. HECATE explores the scale management space to optimize for latency, while ELASM additionally considers the error/latency tradeoff.

5 Conclusion

This work provides example LWE parameter sets that can be used in FHE implementations to target particular levels of security. We also make available the code used to estimate the security of these parameter sets. We recognize the dynamic nature of cryptographic attacks and the necessity of updating our parameters in response to significant advancements in lattice cryptanalysis. We anticipate if these advancements are integrated into the Lattice Estimator, then using our methods and code will enable users

[24] The relevant code can be found in files `bfvrns-parametergeneration.cpp`, `bgvrns-parametergeneration.cpp`, and `ckksrns-parametergeneration.cpp` (Retrieved from OpenFHE v1.1.4 – commit `94fd76a`).

[25] The relevant code can be found in the file `hestdparms.h` (Retrieved from SEAL v4.1.1 – commit `206648d`).

[26] The relevant code can be found in the file `encryption_parameter_selector.h` (Retrieved from EVA v1.0.1 – commit `4cd3254`).

[27] Documentation on the optimizer can be found in the file `optimizer.md` (Retrieved from Concrete v2.5.0 – commit `240ae2d`).

to independently update these parameter sets as necessitated by new developments. Furthermore, as the field of FHE matures and expands, we hope that more types of FHE schemes, diverse secret distributions, and comprehensive attack scenarios can be integrated into future guidelines.

This work provides examples of functional parameter sets that could be used for particular FHE schemes in different contexts, and reviews parameter selection support in some of the major FHE libraries. In practice, it is not only security that must be considered, but also functional correctness and efficiency; and the optimal choice of parameters may be application- and library-dependent. An advanced parameter selection framework for FHE that takes into account all these aspects is an important direction for future research.

Acknowledgements

The authors would like to thank Alexander Viand for the valuable discussion on automatic parameter management using compilers.

Erin Hales was supported by the EPSRC and the UK Government as part of the Centre for Doctoral Training in Cyber Security at Royal Holloway, University of London (EP/P009301/1).

Alberto Pedrouzo-Ulloa is partially funded by the EU H2020 under TRUMPET (proj. no. 101070038), by FEDER and Xunta de Galicia under "Grupos de Referencia Competitiva" (ED431C 2021/47), by FEDER and MCIN/AEI under FELDSPAR (TED2021-130624B-C21), by "NextGenerationEU/PRTR" under TRUFFLES, and under a Margarita Salas grant of Universidade de Vigo.

Wei Dai contributed to this work partly during a previous employment at Microsoft Research.

Ilaria Chillotti contributed to this work during a previous employment at Zama.

Donggeon Yhee contributed to this work during a previous employment at Seoul National University.

Bahattin Yildiz contributed to most part of this work during a previous employment at Intel Labs.

References

AA22. Furkan Aydin and Aydin Aysu. Exposing side-channel leakage of SEAL homomorphic encryption library. In *Proceedings of the 2022 Workshop on Attacks and Solutions in Hardware Security*, ASHES'22, page 95–100, New York, NY, USA, 2022. Association for Computing Machinery.

ABD16. Martin R. Albrecht, Shi Bai, and Léo Ducas. A subfield lattice attack on overstretched NTRU assumptions - cryptanalysis of some FHE and graded encoding schemes. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I*, volume 9814 of *Lecture Notes in Computer Science*, pages 153–178. Springer, 2016.

ABD+20. Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. kyber-specification-round3. `https://pq-crystals.org/kyber/data/kyber-specification-round3.pdf`, October 2020. (Accessed on 04/18/2023).

ACC+19. Martin Albrecht, Melissa Chase, Hao Chen, Jintai Ding, Shafi Goldwasser, Sergey Gorbunov, Shai Halevi, Jeffrey Hoffstein, Kim Laine, Kristin Lauter, Satya Lokam, Daniele Micciancio, Dustin Moody, Travis Morrison, Amit Sahai, and Vinod Vaikuntanathan. Homomorphic encryption standard. Cryptology ePrint Archive, Paper 2019/939, 2019. `https://eprint.iacr.org/2019/939`.

ACD$^+$18. Martin R. Albrecht, Benjamin R. Curtis, Amit Deo, Alex Davidson, Rachel Player, Eamonn W. Postlethwaite, Fernando Virdia, and Thomas Wunderer. Estimate all the {LWE, NTRU} schemes! In Dario Catalano and Roberto De Prisco, editors, *Security and Cryptography for Networks - 11th International Conference, SCN 2018, Amalfi, Italy, September 5-7, 2018, Proceedings*, volume 11035 of *Lecture Notes in Computer Science*, pages 351–367. Springer, 2018.

ACF$^+$15. Martin R. Albrecht, Carlos Cid, Jean-Charles Faugère, Robert Fitzpatrick, and Ludovic Perret. Algebraic algorithms for LWE problems. *ACM Commun. Comput. Algebra*, 49(2):62, 2015.

ACPS09. Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In Shai Halevi, editor, *Advances in Cryptology – CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 595–618, Santa Barbara, CA, USA, August 16–20, 2009. Springer, Heidelberg, Germany.

ACW19. Martin R. Albrecht, Benjamin R. Curtis, and Thomas Wunderer. Exploring trade-offs in batch bounded distance decoding. In Kenneth G. Paterson and Douglas Stebila, editors, *Selected Areas in Cryptography - SAC 2019 - 26th International Conference, Waterloo, ON, Canada, August 12-16, 2019, Revised Selected Papers*, volume 11959 of *Lecture Notes in Computer Science*, pages 467–491. Springer, 2019.

ADPS16. Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange - A new hope. In Thorsten Holz and Stefan Savage, editors, *USENIX Security 2016: 25th USENIX Security Symposium*, pages 327–343, Austin, TX, USA, August 10–12, 2016. USENIX Association.

AG11. Sanjeev Arora and Rong Ge. New algorithms for learning in presence of errors. In Luca Aceto, Monika Henzinger, and Jirí Sgall, editors, *Automata, Languages and Programming - 38th International Colloquium, ICALP 2011, Zurich, Switzerland, July 4-8, 2011, Proceedings, Part I*, volume 6755 of *Lecture Notes in Computer Science*, pages 403–415. Springer, 2011.

AKP$^+$22. Furkan Aydin, Emre Karabulut, Seetal Potluri, Erdem Alkim, and Aydin Aysu. RevEAL: Single-trace side-channel leakage of the SEAL homomorphic encryption library. In *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1527–1532, 2022.

Alb17. Martin R. Albrecht. On dual lattice attacks against small-secret LWE and parameter choices in helib and SEAL. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part II*, volume 10211 of *Lecture Notes in Computer Science*, pages 103–129, 2017.

APS15. Martin Albrecht, Rachel Player, and Samuel Scott. On the concrete hardness of learning with errors. *Journal of Mathematical Cryptology*, 9(3):169–203, 10 2015.

BBB$^+$22. Ahmad Al Badawi, Jack Bates, Flávio Bergamaschi, David Bruce Cousins, Saroja Erabelli, Nicholas Genise, Shai Halevi, Hamish Hunt, Andrey Kim, Yongwoo Lee, Zeyu Liu, Daniele Micciancio, Ian Quah, Yuriy Polyakov, R. V. Saraswathy, Kurt Rohloff, Jonathan Saylor, Dmitriy Suponitsky, Matthew Triplett, Vinod Vaikuntanathan, and Vincent Zucca. Openfhe: Open-source fully homomorphic encryption library. In Michael Brenner, Anamaria Costache, and Kurt Rohloff, editors, *Proceedings of the 10th Workshop on Encrypted Computing & Applied Homomorphic Cryptography, Los Angeles, CA, USA, 7 November 2022*, pages 53–63. ACM, 2022.

BBB$^+$23. Loris Bergerat, Anas Boudi, Quentin Bourgerie, Ilaria Chillotti, Damien Ligier, Jean-Baptiste Orfila, and Samuel Tap. Parameter optimization and larger precision for (T)FHE. *J. Cryptol.*, 36(3):28, 2023.

BCC$^+$22. Youngjin Bae, Jung Hee Cheon, Wonhee Cho, Jaehyung Kim, and Taekyung Kim. Meta-bts: Bootstrapping precision beyond the limit. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, CCS '22, page 223–234, New York, NY, USA, 2022. Association for Computing Machinery.

BDGL16. Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 10–24. SIAM, 2016.

BG14. Shi Bai and Steven D. Galbraith. Lattice decoding attacks on binary LWE. In Willy Susilo and Yi Mu, editors, *Information Security and Privacy - 19th Australasian Conference, ACISP 2014, Wollongong, NSW, Australia, July 7-9, 2014. Proceedings*, volume 8544 of *Lecture Notes in Computer Science*, pages 322–337. Springer, 2014.

BGV12. Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, page 309–325, New York, NY, USA, 2012. Association for Computing Machinery.

BIP+22. Charlotte Bonte, Ilia Iliashenko, Jeongeun Park, Hilder V. L. Pereira, and Nigel P. Smart. FINAL: faster FHE instantiated with NTRU and LWE. In Shweta Agrawal and Dongdai Lin, editors, *Advances in Cryptology - ASIACRYPT 2022 - 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5-9, 2022, Proceedings, Part II*, volume 13792 of *Lecture Notes in Computer Science*, pages 188–215. Springer, 2022.

BL21. Daniel J. Bernstein and Tanja Lange. Non-randomness of s-unit lattices. *IACR Cryptol. ePrint Arch.*, page 1428, 2021.

BLLN13. Joppe W. Bos, Kristin E. Lauter, Jake Loftus, and Michael Naehrig. Improved security for a ring-based fully homomorphic encryption scheme. In Martijn Stam, editor, *Cryptography and Coding - 14th IMA International Conference, IMACC 2013, Oxford, UK, December 17-19, 2013. Proceedings*, volume 8308 of *Lecture Notes in Computer Science*, pages 45–64. Springer, 2013.

BLLW22. Lei Bi, Xianhui Lu, Junjie Luo, and Kunpeng Wang. Hybrid dual and meet-LWE attack. In Khoa Nguyen, Guomin Yang, Fuchun Guo, and Willy Susilo, editors, *ACISP 22: 27th Australasian Conference on Information Security and Privacy*, volume 13494 of *Lecture Notes in Computer Science*, pages 168–188, Wollongong, NSW, Australia, November 28–30, 2022. Springer, Heidelberg, Germany.

BLP+13. Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th Annual ACM Symposium on Theory of Computing*, pages 575–584, Palo Alto, CA, USA, June 1–4, 2013. ACM Press.

BMTPH21. Jean-Philippe Bossuat, Christian Mouchet, Juan Troncoso-Pastoriza, and Jean-Pierre Hubaux. Efficient bootstrapping for approximate homomorphic encryption with non-sparse keys. In *Advances in Cryptology–EUROCRYPT 2021: 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17–21, 2021, Proceedings, Part I*, pages 587–617. Springer, 2021.

Bra12. Zvika Brakerski. Fully homomorphic encryption without modulus switching from classical GapSVP. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 868–886, Santa Barbara, CA, USA, August 19–23, 2012. Springer, Heidelberg, Germany.

BTPH22. Jean-Philippe Bossuat, Juan Troncoso-Pastoriza, and Jean-Pierre Hubaux. Bootstrapping for approximate homomorphic encryption with negligible failure-probability by using sparse-secret encapsulation. In Giuseppe Ateniese and Daniele Venturi, editors, *Applied Cryptography and Network Security*, pages 521–541, Cham, 2022. Springer International Publishing.

BY88. Ernest F. Brickell and Yacov Yacobi. On privacy homomorphisms (extended abstract). In David Chaum and Wyn L. Price, editors, *Advances in Cryptology — EUROCRYPT' 87*, pages 117–125, Berlin, Heidelberg, 1988. Springer Berlin Heidelberg.

CCP+24. Jung Hee Cheon, Hyeongmin Choe, Alain Passelègue, Damien Stehlé, and Elias Suvanto. Attacks against the indcpa-d security of exact fhe schemes. Cryptology ePrint Archive, Paper 2024/127, 2024. https://eprint.iacr.org/2024/127.

CCS19. Hao Chen, Ilaria Chillotti, and Yongsoo Song. Improved bootstrapping for approximate homomorphic encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 34–54. Springer, 2019.

CDPR16. Ronald Cramer, Léo Ducas, Chris Peikert, and Oded Regev. Recovering short generators of principal ideals in cyclotomic rings. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II*, volume 9666 of *Lecture Notes in Computer Science*, pages 559–585. Springer, 2016.

CGGI16. Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology – ASIACRYPT 2016, Part I*, volume 10031 of *Lecture Notes in Computer Science*, pages 3–33, Hanoi, Vietnam, December 4–8, 2016. Springer, Heidelberg, Germany.

CGGI17. Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. Faster packed homomorphic operations and efficient circuit bootstrapping for TFHE. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017, Part I*, volume 10624 of *Lecture Notes in Computer Science*, pages 377–408, Hong Kong, China, December 3–7, 2017. Springer, Heidelberg, Germany.

CH18. Hao Chen and Kyoohyung Han. Homomorphic lower digits removal and improved fhe bootstrapping. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 315–337. Springer, 2018.

CHHS19. Jung Hee Cheon, Minki Hhan, Seungwan Hong, and Yongha Son. A hybrid of dual and meet-in-the-middle attack on sparse and ternary secret lwe. *IEEE Access*, 7:89497–89506, 2019.

CHK$^+$16. Jung Hee Cheon, Kyoohyung Han, Jinsu Kim, Changmin Lee, and Yongha Son. A practical post-quantum public-key cryptosystem based on spLWE. In *International Conference on Information Security and Cryptology*, pages 51–74. Springer, 2016.

CHK$^+$18. Jung Hee Cheon, Kyoohyung Han, Andrey Kim, Miran Kim, and Yongsoo Song. Bootstrapping for approximate homomorphic encryption. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018*, pages 360–384, Cham, 2018. Springer International Publishing.

CIV16. Wouter Castryck, Ilia Iliashenko, and Frederik Vercauteren. Provably weak instances of ring-lwe revisited. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I*, volume 9665 of *Lecture Notes in Computer Science*, pages 147–167. Springer, 2016.

CJL16. Jung Hee Cheon, Jinhyuck Jeong, and Changmin Lee. An algorithm for NTRU problems and cryptanalysis of the GGH multilinear map without an encoding of zero. *IACR Cryptol. ePrint Arch.*, page 139, 2016.

CKKS17. Jung Hee Cheon, Andrey Kim, Miran Kim, and Yong Soo Song. Homomorphic encryption for arithmetic of approximate numbers. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017, Part I*, volume 10624 of *Lecture Notes in Computer Science*, pages 409–437, Hong Kong, China, December 3–7, 2017. Springer, Heidelberg, Germany.

CLS16. Hao Chen, Kristin E. Lauter, and Katherine E. Stange. Security considerations for galois non-dual RLWE families. In Roberto Avanzi and Howard M. Heys, editors, *Selected Areas in Cryptography - SAC 2016 - 23rd International Conference, St. John's, NL, Canada, August 10-12, 2016, Revised Selected Papers*, volume 10532 of *Lecture Notes in Computer Science*, pages 443–462. Springer, 2016.

CP19. Benjamin R Curtis and Rachel Player. On the feasibility and impact of standardising sparse-secret lwe parameter sets for homomorphic encryption. In *Proceedings of the 7th ACM Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, pages 1–10, 2019.

CSBB24. Marina Checri, Renaud Sirdey, Aymen Boudguiga, and Jean-Paul Bultel. On the practical cpad security of "exact" and threshold fhe schemes and libraries. Cryptology ePrint Archive, Paper 2024/116, 2024. `https://eprint.iacr.org/2024/116`.

CST22. Kevin Carrier, Yixin Shen, and Jean-Pierre Tillich. Faster dual lattice attacks by using coding theory. Cryptology ePrint Archive, Paper 2022/1750, 2022. `https://eprint.iacr.org/2022/1750`.

DDGR20. Dana Dachman-Soled, Léo Ducas, Huijing Gong, and Mélissa Rossi. LWE with side information: Attacks and concrete security estimation. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020, Part II*, volume 12171 of *Lecture Notes in Computer Science*, pages 329–358, Santa Barbara, CA, USA, August 17–21, 2020. Springer, Heidelberg, Germany.

DGHK23. Dana Dachman-Soled, Huijing Gong, Tom Hanson, and Hunter Kippen. Revisiting security estimation for lwe with hints from a geometric perspective. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology – CRYPTO 2023*, pages 748–781, Cham, 2023. Springer Nature Switzerland.

DKS$^+$20. Roshan Dathathri, Blagovesta Kostova, Olli Saarikivi, Wei Dai, Kim Laine, and Madan Musuvathi. EVA: an encrypted vector arithmetic language and compiler for efficient homomorphic computation. In Alastair F. Donaldson and Emina Torlak, editors, *Proceedings of the 41st ACM SIGPLAN International Conference on Programming Language Design and Implementation, PLDI 2020, London, UK, June 15-20, 2020*, pages 546–561. ACM, 2020.

DP22. Nir Drucker and Tomer Pelleg. Timing leakage analysis of non-constant-time NTT implementations with Harvey butterflies. In Shlomi Dolev, Jonathan Katz, and Amnon Meisels, editors, *Cyber Security, Cryptology, and Machine Learning*, pages 99–117, Cham, 2022. Springer International Publishing.

DP23a. Léo Ducas and Ludo N. Pulles. Accurate score prediction for dual-sieve attacks. Cryptology ePrint Archive, Paper 2023/1850, 2023. `https://eprint.iacr.org/2023/1850`.

DP23b. Léo Ducas and Ludo N. Pulles. Does the dual-sieve attack on learning with errors even work? In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part III*, volume 14083 of *Lecture Notes in Computer Science*, pages 37–69. Springer, 2023.

DvW21. Léo Ducas and Wessel P. J. van Woerden. NTRU fatigue: How stretched is overstretched? In Mehdi Tibouchi and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2021 - 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6-10, 2021, Proceedings, Part IV*, volume 13093 of *Lecture Notes in Computer Science*, pages 3–32. Springer, 2021.

EGMS23. Andre Esser, Rahul Girme, Arindam Mukherjee, and Santanu Sarkar. Memory-efficient attacks on small LWE keys. In Jian Guo and Ron Steinfeld, editors, *Advances in Cryptology - ASIACRYPT 2023 - 29th International Conference on the Theory and Application of Cryptology and Information Security, Guangzhou, China, December 4-8, 2023, Proceedings, Part IV*, volume 14441 of *Lecture Notes in Computer Science*, pages 72–105. Springer, 2023.

EHL14. Kirsten Eisenträger, Sean Hallgren, and Kristin E. Lauter. Weak instances of PLWE. In Antoine Joux and Amr M. Youssef, editors, *Selected Areas in Cryptography - SAC 2014 - 21st International Conference, Montreal, QC, Canada, August 14-15, 2014, Revised Selected Papers*, volume 8781 of *Lecture Notes in Computer Science*, pages 183–194. Springer, 2014.

EJK20. Thomas Espitau, Antoine Joux, and Natalia Kharchenko. On a dual/hybrid approach to small secret LWE - A dual/enumeration technique for learning with errors and application to security estimates of FHE schemes. In Karthikeyan Bhargavan, Elisabeth Oswald, and Manoj Prabhakaran, editors, *Progress in Cryptology - INDOCRYPT 2020 - 21st International Conference on Cryptology in India, Bangalore, India, December 13-16, 2020, Proceedings*, volume 12578 of *Lecture Notes in Computer Science*, pages 440–462. Springer, 2020.

ELOS15. Yara Elias, Kristin E. Lauter, Ekin Ozman, and Katherine E. Stange. Provably weak instances of ring-lwe. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 63–92. Springer, 2015.

FV12. Junfeng Fan and Frederik Vercauteren. Somewhat practical fully homomorphic encryption. *IACR Cryptol. ePrint Arch.*, 2012:144, 2012.

Gee24. Robin Geelen. Revisiting the slot-to-coefficient transformation for BGV and BFV. *IACR Cryptol. ePrint Arch.*, page 153, 2024.

Gen09. Craig Gentry. Computing on encrypted data (invited talk). In Juan A. Garay, Atsuko Miyaji, and Akira Otsuka, editors, *CANS 09: 8th International Conference on Cryptology and Network Security*, volume 5888 of *Lecture Notes in Computer Science*, page 477, Kanazawa, Japan, December 12–14, 2009. Springer, Heidelberg, Germany.

GIKV23. Robin Geelen, Ilia Iliashenko, Jiayi Kang, and Frederik Vercauteren. On polynomial functions modulo p^{e} and faster bootstrapping for homomorphic encryption. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part III*, volume 14006 of *Lecture Notes in Computer Science*, pages 257–286. Springer, 2023.

GJ21. Qian Guo and Thomas Johansson. Faster dual lattice attacks for solving LWE with applications to CRYSTALS. In Mehdi Tibouchi and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2021 - 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6-10, 2021, Proceedings, Part IV*, volume 13093 of *Lecture Notes in Computer Science*, pages 33–62. Springer, 2021.

GJS15. Qian Guo, Thomas Johansson, and Paul Stankovski. Coded-bkw: Solving LWE using lattice codes. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 23–42. Springer, 2015.

GV23. Robin Geelen and Frederik Vercauteren. Bootstrapping for BGV and BFV revisited. *J. Cryptol.*, 36(2):12, 2023.

HG07. Nick Howgrave-Graham. A hybrid lattice-reduction and meet-in-the-middle attack against ntru. In *Advances in Cryptology-CRYPTO 2007: 27th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2007. Proceedings 27*, pages 150–169. Springer, 2007.

HK20. Kyoohyung Han and Dohyeong Ki. Better bootstrapping for approximate homomorphic encryption. In *Topics in Cryptology–CT-RSA 2020: The Cryptographers' Track at the RSA Conference 2020, San Francisco, CA, USA, February 24–28, 2020, Proceedings*, pages 364–390. Springer, 2020.

HKLS22. Minki Hhan, Jiseung Kim, Changmin Lee, and Yongha Son. How to meet ternary lwe keys on babai's nearest plane. *Cryptology ePrint Archive*, 2022.

How07. Nick Howgrave-Graham. A hybrid lattice-reduction and meet-in-the-middle attack against NTRU. In Alfred Menezes, editor, *Advances in Cryptology - CRYPTO 2007, 27th Annual International*

Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2007, Proceedings, volume 4622 of *Lecture Notes in Computer Science*, pages 150–169. Springer, 2007.

HPS98. Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A ring-based public key cryptosystem. In Joe Buhler, editor, *Algorithmic Number Theory, Third International Symposium, ANTS-III, Portland, Oregon, USA, June 21-25, 1998, Proceedings*, volume 1423 of *Lecture Notes in Computer Science*, pages 267–288. Springer, 1998.

HS20. Shai Halevi and Victor Shoup. Design and implementation of helib: a homomorphic encryption library. Cryptology ePrint Archive, Paper 2020/1481, 2020. `https://eprint.iacr.org/2020/1481`.

HS21. Shai Halevi and Victor Shoup. Bootstrapping for helib. *Journal of Cryptology*, 34(1):7, 2021.

JCH23. Sergio Pastrana José Cabrero-Holgueras. Towards automated homomorphic encryption parameter selection with fuzzy logic and linear programming. In Binshan Li, editor, *Expert Systems with Applications, Volume 229 Part A*, volume 229 of *Lecture Notes in Computer Science*, pages 13–20. Springer, 2023.

JVC18. Chiraag Juvekar, Vinod Vaikuntanathan, and Anantha Chandrakasan. Gazelle: A low latency framework for secure neural network inference. In *Proceedings of the 27th USENIX Conference on Security Symposium*, SEC'18, page 1651–1668, USA, 2018. USENIX Association.

KDE+24. Andrey Kim, Maxim Deryabin, Jieun Eom, Rakyong Choi, Yongwoo Lee, Whan Ghang, and Donghoon Yoo. General bootstrapping approach for rlwe-based homomorphic encryption. *IEEE Transactions on Computers*, 73(01):86–96, jan 2024.

KF15. Paul Kirchner and Pierre-Alain Fouque. An improved BKW algorithm for LWE with applications to cryptography and lattices. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 43–62. Springer, 2015.

KF17. Paul Kirchner and Pierre-Alain Fouque. Revisiting lattice attacks on overstretched NTRU parameters. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part I*, volume 10210 of *Lecture Notes in Computer Science*, pages 3–26, 2017.

KL21. Kim Laine Kristin Lauter, Wei Dai, editor. *Protecting Privacy through Homomorphic Encryption.* Springer Cham, December 2021.

Klu22. Kamil Kluczniak. Ntru-v-um: Secure fully homomorphic encryption from NTRU with small modulus. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 2022*, pages 1783–1797. ACM, 2022.

KSS24. Jaehyung Kim, Jinyeong Seo, and Yongsoo Song. Simpler and faster bfv bootstrapping for arbitrary plaintext modulus from ckks. Cryptology ePrint Archive, Paper 2024/109, 2024. `https://eprint.iacr.org/2024/109`.

LCK+23. Yongwoo Lee, Seonyoung Cheon, Dongkwan Kim, Dongyoon Lee, and Hanjun Kim. ELASM: Error-Latency-Aware scale management for fully homomorphic encryption. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 4697–4714, Anaheim, CA, August 2023. USENIX Association.

LHC+22. Yongwoo Lee, Seonyeong Heo, Seonyoung Cheon, Shinnung Jeong, Changsu Kim, Eunkyung Kim, Dongyoon Lee, and Hanjun Kim. Hecate: Performance-aware scale optimization for homomorphic encryption compiler. In Jae W. Lee, Sebastian Hack, and Tatiana Shpeisman, editors, *CGO 2022 - Proceedings of the 2022 IEEE/ACM International Symposium on Code Generation and Optimization*, CGO 2022 - Proceedings of the 2022 IEEE/ACM International Symposium on Code Generation and Optimization, pages 193–204, United States, 2022. Institute of Electrical and Electronics Engineers Inc. Publisher Copyright: © 2022 IEEE.; 20th IEEE/ACM International Symposium on Code Generation and Optimization, CGO 2022 ; Conference date: 02-04-2022 Through 06-04-2022.

LM21. Baiyu Li and Daniele Micciancio. On the security of homomorphic encryption on approximate numbers. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology – EUROCRYPT 2021, Part I*, volume 12696 of *Lecture Notes in Computer Science*, pages 648–677, Zagreb, Croatia, October 17–21, 2021. Springer, Heidelberg, Germany.

LMvdP14. Thijs Laarhoven, Michele Mosca, and Joop van de Pol. Finding shortest lattice vectors faster using quantum search. *IACR Cryptol. ePrint Arch.*, page 907, 2014.

LN13. Mingjie Liu and Phong Q. Nguyen. Solving BDD by enumeration: An update. In Ed Dawson, editor, *Topics in Cryptology - CT-RSA 2013 - The Cryptographers' Track at the RSA Conference 2013,* *San Francisco,CA, USA, February 25-March 1, 2013. Proceedings*, volume 7779 of *Lecture Notes in Computer Science*, pages 293–309. Springer, 2013.

LPR10. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 1–23, French Riviera, May 30 – June 3, 2010. Springer, Heidelberg, Germany.

LS15. Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *Des. Codes Cryptography*, 75(3):565–599, jun 2015.

LSW+23. Cathy Yuanchen Li, Jana Sotáková, Emily Wenger, Mohamed Malhou, Evrard Garcelon, François Charton, and Kristin E. Lauter. SalsaPicante: A machine learning attack on LWE with binary secrets. In Weizhi Meng, Christian Damsgaard Jensen, Cas Cremers, and Engin Kirda, editors, *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, CCS 2023, Copenhagen, Denmark, November 26-30, 2023*, pages 2606–2620. ACM, 2023.

LTV12. Adriana López-Alt, Eran Tromer, and Vinod Vaikuntanathan. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In Howard J. Karloff and Toniann Pitassi, editors, *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 1219–1234. ACM, 2012.

LW24. Zeyu Liu and Yunhao Wang. Relaxed functional bootstrapping: A new perspective on bgv/bfv bootstrapping. Cryptology ePrint Archive, Paper 2024/172, 2024. `https://eprint.iacr.org/2024/172`.

LWA+23. Cathy Yuanchen Li, Emily Wenger, Zeyuan Allen-Zhu, François Charton, and Kristin E. Lauter. SALSA VERDE: a machine learning attack on LWE with sparse small secrets. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

MAT22. MATZOV. Report on the security of lwe: Improved dual lattice attack, 2022.

May21. Alexander May. How to meet ternary LWE keys. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part II*, volume 12826 of *Lecture Notes in Computer Science*, pages 701–731. Springer, 2021.

MCR21. Muhammad Haris Mughees, Hao Chen, and Ling Ren. Onionpir: Response efficient single-server pir. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21, page 2292–2306, New York, NY, USA, 2021. Association for Computing Machinery.

MHWW24. Shihe Ma, Tairong Huang, Anyu Wang, and Xiaoyun Wang. Faster bgv bootstrapping for power-of-two cyclotomics through homomorphic ntt. Cryptology ePrint Archive, Paper 2024/164, 2024. `https://eprint.iacr.org/2024/164`.

MP21. Daniele Micciancio and Yuriy Polyakov. Bootstrapping in fhew-like cryptosystems. In *Proceedings of the 9th on Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, WAHC '21, page 17–28, New York, NY, USA, 2021. Association for Computing Machinery.

MR09. Daniele Micciancio and Oded Regev. Lattice-based cryptography. In Daniel J. Bernstein, Johannes Buchmann, and Erik Dahmen, editors, *Post-Quantum Cryptography*, pages 147–191. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

NMW+24. Niklas Nolte, Mohamed Malhou, Emily Wenger, Samuel Stevens, Cathy Yuanchen Li, Francois Charton, and Kristin Lauter. The cool and the cruel: separating hard parts of lwe secrets. Cryptology ePrint Archive, Paper 2024/443, 2024. `https://eprint.iacr.org/2024/443`.

OPP23. Hiroki Okada, Rachel Player, and Simon Pohmann. Homomorphic polynomial evaluation using galois structure and applications to BFV bootstrapping. In Jian Guo and Ron Steinfeld, editors, *Advances in Cryptology - ASIACRYPT 2023 - 29th International Conference on the Theory and Application of Cryptology and Information Security, Guangzhou, China, December 4-8, 2023, Proceedings, Part VI*, volume 14443 of *Lecture Notes in Computer Science*, pages 69–100. Springer, 2023.

oST23. National Institute of Standards and Technology. Module-lattice-based key-encapsulation mechanism standard. Technical Report Federal Information Processing Standards Publications (FIPS PUBS) 203 (Draft) August 24, 2023, U.S. Department of Commerce, Washington, D.C., 2023.

Pei09. Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In Michael Mitzenmacher, editor, *41st Annual ACM Symposium on Theory of Computing*, pages 333–342, Bethesda, MD, USA, May 31 – June 2, 2009. ACM Press.

Pei16. Chris Peikert. How (not) to instantiate ring-lwe. In Vassilis Zikas and Roberto De Prisco, editors, *Security and Cryptography for Networks - 10th International Conference, SCN 2016, Amalfi, Italy,* *August 31 - September 2, 2016, Proceedings*, volume 9841 of *Lecture Notes in Computer Science*, pages 411–430. Springer, 2016.

PHS19. Alice Pellet-Mary, Guillaume Hanrot, and Damien Stehlé. Approx-svp in ideal lattices with pre-processing. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part II*, volume 11477 of *Lecture Notes in Computer Science*, pages 685–716. Springer, 2019.

PPM17. Robert Primas, Peter Pessl, and Stefan Mangard. Single-trace side-channel attacks on masked lattice-based encryption. In Wieland Fischer and Naofumi Homma, editors, *Cryptographic Hardware and Embedded Systems – CHES 2017*, volume 10529 of *Lecture Notes in Computer Science*, pages 513–533, Taipei, Taiwan, September 25–28, 2017. Springer, Heidelberg, Germany.

PRSD17. Chris Peikert, Oded Regev, and Noah Stephens-Davidowitz. Pseudorandomness of ring-lwe for any ring and modulus. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, page 461–473, New York, NY, USA, 2017. Association for Computing Machinery.

PS23. Amaury Pouly and Yixin Shen. Provable dual attacks on learning with errors. Cryptology ePrint Archive, Paper 2023/1508, 2023. Accepted for publication at Eurocrypt 2024. Available at `https://eprint.iacr.org/2023/1508`.

RAD78. R L Rivest, L Adleman, and M L Dertouzos. On data banks and privacy homomorphisms. *Foundations of Secure Computation, Academia Press*, pages 169–179, 1978.

Reg05. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *2005 ACM 37th Annual Conference on Theory of Computing*, pages 84–93, 2005.

Reg10. Oded Regev. The learning with errors problem (invited survey). In *2010 IEEE 25th Annual Conference on Computational Complexity*, pages 191–204, 2010.

SC19. Yongha Son and Jung Hee Cheon. Revisiting the hybrid attack on sparse secret lwe and application to he parameters. In *Proceedings of the 7th ACM Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, WAHC'19, page 11–20, New York, NY, USA, 2019. Association for Computing Machinery.

SEA23. Microsoft SEAL (release 4.1). `https://github.com/Microsoft/SEAL`, January 2023. Microsoft Research, Redmond, WA.

SS11. Damien Stehlé and Ron Steinfeld. Making NTRU as secure as worst-case problems over ideal lattices. In Kenneth G. Paterson, editor, *Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings*, volume 6632 of *Lecture Notes in Computer Science*, pages 27–47. Springer, 2011.

SSTX09. Damien Stehlé, Ron Steinfeld, Keisuke Tanaka, and Keita Xagawa. Efficient public key encryption based on ideal lattices. In Mitsuru Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 617–635, Tokyo, Japan, December 6–10, 2009. Springer, Heidelberg, Germany.

SWL+24. Samuel Stevens, Emily Wenger, Cathy Yuanchen Li, Niklas Nolte, Eshika Saxena, Francois Charton, and Kristin Lauter. Salsa fresca: Angular embeddings and pre-training for ml attacks on learning with errors. Cryptology ePrint Archive, Paper 2024/150, 2024. `https://eprint.iacr.org/2024/150`.

Tun23. Tune Insight. *Lattigo v5*, November 2023. EPFL-LDS, Tune Insight SA.

WCCL22. Emily Wenger, Mingjie Chen, François Charton, and Kristin E. Lauter. SALSA: attacking lattice cryptography with transformers. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

XWW+24. Wenwen Xia, Leizhang Wang, Geng Wang, Dawu Gu, and Baocang Wang. A refined hardness estimation of LWE in two-step mode. Cryptology ePrint Archive, Paper 2024/067, 2024. Accepted for publication at PKC 2024. Available at `https://eprint.iacr.org/2024/067`.

XZD+23. Binwu Xiang, Jiang Zhang, Yi Deng, Yiran Dai, and Dengguo Feng. Fast blind rotation for bootstrapping fhes. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part IV*, volume 14084 of *Lecture Notes in Computer Science*, pages 3–36. Springer, 2023.

Zam22a. Zama. Concrete: TFHE Compiler that converts python programs into FHE equivalent, 2022. `https://github.com/zama-ai/concrete`.

Zam22b. Zama. TFHE-rs: A Pure Rust Implementation of the TFHE Scheme for Boolean and Integer Arithmetics Over Encrypted Data, 2022. https://github.com/zama-ai/tfhe-rs.

A CKKS bootstrapping failure probability

In this Appendix we give more details about the failure probability in CKKS bootstrapping as briefly mentioned in Table 4.7. We omit a full description of CKKS bootstrapping and refer the reader to e.g. [CHK+18,CCS19,HK20,BMTPH21,BCC+22] for more details.

The bootstrapping failure probability plays a crucial role in the practicality of CKKS bootstrapping and it is related to the EvalMod step. The EvalMod step of the bootstrapping takes as input the message $I(Y) \cdot Q + \Delta m(Y)$ with $Y = X^{N/2M}$ (M being the number of complex slots) and aims to vanish the integer polynomial $I(Y)$ by homomorphically evaluating the function $f_{\mathsf{mod}} = x \mod 1$ in the union of intervals $\cup_{i=-K}^{K}[i-\epsilon, i+\epsilon]$, with $[-\epsilon, \epsilon]$ being the expected interval where the original message lies. The coefficients of the polynomial $I(Y)$ are the sum of $h+1$ uniform random variables in $[-0.5, 0.5)$, with h the Hamming weight of the secret.

Remark 1. There have been many works proposing different approaches for the EvalMod step. However, all practical approaches follow the same blueprint, which is to find a good polynomial approximation of f_{mod}. Which function is chosen to closely match f_{mod} and how the polynomial approximation is done has no effect on the failure probability. Only the interval in which it is approximated, i.e. the parameter K, affects the failure probability.

If $\|I(Y)\| > K$, then the EvalMod step returns an unusable corrupted plaintext. This failure probability is defined as $f_{\mathsf{fail}}(K, h, M) = \mathsf{Pr}\left[\|I(Y)\| > K\right]$ by [BTPH22] and they show how to compute it by adapting the Irwin Hall cumulative distribution function:

$$f_{\mathsf{fail}}(K, h, M) \colon 1 - \left(\frac{2}{(h+1)!} \left(\sum_{i=0}^{\lfloor K+0.5(h+1) \rfloor} (-1)^i \binom{h+1}{i} (K + 0.5(h+1) - i)^{h+1} \right) - 1 \right)^{2M}. \tag{1}$$

Usually the bootstrapping parameters are instantiated using a secret with fixed Hamming weight h, which allows to get an exact estimation of $f_{\mathsf{fail}}(K, h, M)$, and thus to choose K according to the desired failure probability. However, in our case we have a ternary secret with coefficients sampled with probability $[p/2, 1-p, p/2]$ and $p = 2/3$, thus the exact value of h is unknown and this prevents from being able to estimate the exact failure probability. We provide a procedure to find a suitable K in such case given N, p and M and a desired failure probability 2^{δ} for some $\delta < 0$:

1. Estimate K based on $\mathsf{E}[h]$: This step is straightforward and can be done with a binary search on K by successive evaluations of $f_{\mathsf{fail}}(K, \mathsf{E}[h], M)$.
2. Estimate a correction factor K' such that $1 - \mathsf{Pr}[f_{\mathsf{fail}}(K + K', h, M) \leq 2^{\delta}] \leq 2^{\delta}$: Since I follows an Irwin Hall distribution, it is $\mathcal{O}(\sqrt{h})$ and we have

$$K = \left\lceil \kappa \cdot \sqrt{\mathsf{E}[h] + 1} \right\rceil,$$

from which we obtain κ. Let now $\sigma_h = \sqrt{Np(1-p)}$, then the value K will increase by $d\frac{\kappa\sigma_h}{\sqrt{\mathsf{E}[h]+1}} \approx \kappa\sqrt{1-p}$ if h deviates by $d\sigma_h$ of $\mathsf{E}[h]$. [28] Therefore

$$\mathsf{Pr}\big[h \leq \mathsf{E}[h] + d\sigma_h\big] = \mathsf{erf}\left(\frac{d\sigma_h}{\sqrt{2}\sigma_h}\right) = \mathsf{erf}\left(\frac{d}{\sqrt{2}}\right).$$

[28] We assume that d is positive since the converse would not have a negative impact on the failure probability.

Thus given $1 - \mathsf{erf}\left(\frac{d}{\sqrt{2}}\right) \leq 2^{\delta}$ we have $K' = \lceil d\kappa\sqrt{1-p} \rceil$.

3. Set $K = K + K'$.

Following the procedure described above, we implemented the following two helper functions:

1. $\texttt{Probability}(\texttt{Xs}, K, \log_2(N), \log_2(M)) \rightarrow \delta$: given Xs the secret distribution, K, $\log_2(N)$ and $\log_2(M)$ returns $\delta = \log_2(\mathsf{Pr}\left[\|I(Y)\| > K\right])$.
2. $\texttt{FindSuitableK}(\texttt{Xs}, \log_2(N), \log_2(M), \delta) \rightarrow K$: given given Xs the secret distribution, $\log_2(N)$ and $\log_2(M)$ and δ, returns K such that $\mathsf{Pr}\left[\|I(Y)\| > K\right]) \leq 2^{\delta}$.

Both **1.** and **2.** take into account the correction factor K' if Xs is specified as a probability density. The code is available at `https://github.com/gong-cr/FHE-Security-Guidelines/blob/main/RNS-CKKS-examples/lattigo/templates/bootstrapping/failure/failure_probability.go`.

Remark 2. Equation 1 require arbitrary precision arithmetic of precision $2h$ to produce accurate results due to (i) the alternating sum over $K + h/2$ and (ii) the exponentiation by $h + 1$. Thus evaluating 1 is $\mathcal{O}(h^3)$, making it prohibitively expensive for large values of h. Instead, we can pre-compute a table of (K, δ) for a fixed large enough h (e.g. 8192) and a range of δ that are likely to be used in practice (e.g. $0 > \delta > -512$). Then the value K' for some other h' can be approximated by using the relation $\kappa \approx K/\sqrt{h+1} \approx K'/\sqrt{h'+1}$ for a given δ.