Security Guidelines for Implementing Homomorphic Encryption

Jean-Philippe Bossuat[1], Rosario Cammarota[2], Ilaria Chillotti, Benjamin R. Curtis[3(✉)], Wei Dai[4], Huijing Gong[2(✉)], Erin Hales[5,6], Duhyeong Kim[2], Bryan Kumara[7], Changmin Lee[8], Xianhui Lu[9], Carsten Maple[7,10], Alberto Pedrouzo-Ulloa[11], Rachel Player[6(✉)], Yuriy Polyakov[12], Luis Antonio Ruiz Lopez[13], Yongsoo Song[14], and Donggeon Yhee[15]

[1] Gauss Labs Pte. Ltd.
jeanphilippe.bossuat@gmail.com
[2] Intel Labs
{rosario.cammarota, huijing.gong, duhyeong.kim}@intel.com
[3] Zama
ben.curtis@zama.ai
[4] TikTok Inc.
weidai3141@gmail.com
[5] Royal Holloway, University of London
{erin.hales.2018@live.,rachel.player@}rhul.ac.uk
[6] University of Edinburgh
[7] The Alan Turing Institute
{cmaple, 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] Duality Technologies
ypolyakov@dualitytech.com
[13] Lorica Cybersecurity
luis@loricacyber.com
[14] Seoul National University
y.song@snu.ac.kr
[15] dgyhee@gmail.com

Keywords: Fully Homomorphic Encryption, Homomorphic Encryption, Learning With Errors, Parameter Selection, Concrete Security, BFV, BGV, CKKS, CGGI, DM

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 has been 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

✉ Corresponding authors

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.

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], DM [DM15], and CGGI [CGGI16]. A new FHE scheme [LMK+23], which is regarded as a more efficient alternative to DM [AAB+22], is included in this document under the DM umbrella term[16]. These FHE schemes are based on well-known variants of the Learning With Errors (LWE) problem [Reg05], including Ring-LWE (RLWE) [SSTX09, LPR10] and General-LWE (GLWE) [BGV12, CGGI17][17]. 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 main 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 5.1. They are developed using the prevailing methodology to establish parameters for LWE-based cryptography, following works such as [APS15a] and the Lattice Estimator[18]. 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 5.2, mention not only those parameters that are relevant for security but also those relevant for correctness and functionalities. These parameter sets are necessarily exemplar and may not suit all implementations in all application contexts. Thus, in Section 5.3, we also survey the parameter selection support offered in open source FHE libraries.

[16] We note that elsewhere in the literature the CGGI, DM, and LMK+ schemes are sometimes thought of as the same, whilst utilising differing blind rotation algorithms, e.g. in [XZD+23].

[17] 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.

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

1.1 Comparison to prior work [ACC+19]

Our approach builds upon the efforts from previous work by HomomorphicEncryption.org [ACC+19] (later published as [ACC+21]), 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 [DM15, CGGI16, LMK+23] and similar ones [BR15, BDF18, KS22]. 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 security of the parameter sets provided in [ACC+19] was estimated using the (classical) cost model [BDGL16][19] with the LWE Estimator [APS15b], which is an old version of the currently maintained Lattice Estimator [APS15a]. The parameter sets provided in [ACC+19] may now be considered somewhat outdated, due to recent cryptanalytic advancements that may have implications on the concrete hardness of LWE instances used in FHE applications [CHHS19, SC19, EJK20, GJ21, BLLW22, MAT22, CST22, DP23b, PS24, DP23a, XWW+24]. In particular, the security of the parameter sets provided in this work is estimated using the classical cost model [MAT22] in the Lattice Estimator[20]. Despite these differences, both [ACC+19] and our work provide bounds of concrete parameters for certain security levels in the form of lookup tables, and focus on specifying concrete parameters for power-of-two cyclotomic fields for RLWE schemes.

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] is another widely used assumption in lattice-based cryptography. It has been shown that RLWE-like encryption can be built using statistically hard instances of NTRU [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.

[19] Known as `BKZ.sieve` in the LWE Estimator.
[20] Known as `RC.MATZOV` in the Lattice Estimator.

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].

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].

Parameter selection. In Section 5.1 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, CHP23]. Similar such sets [ACC+19] have also been used in major FHE libraries as a lookup table to inform default parameters. We will mention this further in Section 5.3. Efforts have also explored frameworks or formulas as alternatives to lookup tables for selecting FHE parameters, e.g. [BBB+23, MML+23, KMR24].

1.3 Structure of document

The remainder of this document is organized as follows. Section 2 introduces the LWE problem and its algebraic variants used in FHE schemes. Section 3 discusses several security notions relevant to protocols making use of FHE. Section 4 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 5.1 gives examples of LWE parameter sets chosen to target a given security level that can be used in FHE applications. Section 5.2 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 5.3 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).

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 $\mathcal{R}_q$.

Definition 4 **(RLWE distribution).** *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).

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 $\mathcal{R}_q$.

Definition 7 (GLWE **distribution).** *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* $\{(\mathbf{a}_i, b_i) : i = 1, \dots, m\}$ *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 [AG11]. In practice, implementations of RLWE/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 [oST24] makes use of a Centered Binomial Distribution as the error distribution. For example, a Centered Binomial Distribution resulted from 42 fair coin tosses centered at 0 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* or *Gaussian secret.*
– The coefficients of the secret polynomial s are chosen uniformly at random from $\{-1, 0, 1\}$: this is known as *uniform ternary secret.*
– The coefficients of the secret polynomial s are chosen uniformly at random from $\{0, 1\}$: this is known as *uniform 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 LWE-based homomorphic encryption to reduce the complexity of recryption, a part of bootstrapping [HS21], and were previously used to support bootstrapping in Gentry's original scheme [Gen09]. 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 or to the degree of an interpolation polynomial, and consequently this Hamming weight must be bounded and somewhat small [CHK+18, CCS19, HK20, MHWW24] (see also Appendix A). For these reasons, many 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, some implementations of CKKS [BMTPH21] and BFV [OPP23] have correct and reasonably efficient bootstrapping with non-sparse keys.

Reductions exist for the sparse secret variant of LWE, denoted as spLWE. It has been shown that spLWE can be reduced from standard LWE [GKPV10, BLP+13, CHK+16]. As is the case for reductions for LWE with uniform binary and ternary secrets, the reduction is not sufficiently tight to provide useful insight into FHE parameter setting based on uniform-secret LWE hardness.

Many attacks and analyses leverage properties of sparse secrets [How07, CP19, CHHS19, May21, CSY22, HKLS22, LLW24, NMW+24] and thus may be applicable to FHE parameter sets with sparse secrets. Some of these works provide their own tools for estimating the cost of these attacks for specific parameters. However, the Lattice Estimator—the tool we use—currently does not support these cost estimates. As a result, we have opted not to include parameter sets with sparse secrets in the current study, leaving the discussion for future work. We encourage the integration of these attack cost estimates into the Lattice Estimator to enable a more rigorous and equitable evaluation of the concrete security of parameter sets for which these attacks are applicable.

3 Security notions

In this section, we discuss the essential security notions relevant to homomorphic encryption protocols. Designing a protocol using homomorphic encryption requires a comprehensive review by cryptography experts, as the interactions within a protocol define the adversary model and introduce potential attack vectors. To establish the security of a cryptosystem, one must first identify the resources and capabilities available to an attacker and define the criteria for a successful attack. These concepts are typically encapsulated in a security model.

Informally, in security modelling, IND refers to the adversary's goal of distinguishing an encryption of a message from a collection. The adversary is typically given a challenge, that is, an encryption of a random message from the collection, and its task is to identify what message is encrypted by the challenge. In a *chosen plaintext attack* (CPA) the adversary has access to an encryption oracle, and it is allowed to choose any two plaintexts to form the challenge ciphertext. In a *chosen ciphertext attack* (CCA) the adversary also has access to a decryption oracle. There are two standard versions of IND-CCA. In CCA1, the adversary only has access to the decryption oracle before it selects the plaintexts to form the challenge. On the other hand, in CCA2, the adversary also has access to the decryption oracle after it receives the challenge, with the restriction of not being allowed to query the challenge ciphertext itself.

It is well known that IND-CCA2 cannot be satisfied by any cryptosystem with homomorphic properties. For instance, in an additive encryption scheme, simply adding an encryption of 0 to the challenge ciphertext allows the adversary to submit a valid query to the decryption oracle. FHE schemes that are IND-CCA1-secure, or target security against other types of active attacker, have been considered in several works [LMSV12, BSW12, FHR22, AGHV22, MN24]. While theoretically possible, achieving IND-CCA1-secure FHE is currently impractical. In addition, most approaches for achieving IND-CCA1 would require a cryptosystem to never share encrypted key material since it can be queried to the decryption oracle, the response to which would reveal this material in plaintext [LMSV12]. All modern FHE constructions, including those considered in this document, make use of encrypted key material, such as relinearization keys, bootstrapping keys, etc. For the above reasons, IND-CPA has historically been the standard security notion for FHE constructions.

In recent years, there have been several new attacks on all the schemes considered in this paper. The first one of these attacks was described by Li and Micciancio against CKKS in [LM21]. To perform this attack, the adversary must first gain access to decrypted results from valid ciphertexts. The original decryption circuit for CKKS [CKKS17] outputs an approximate version of the encrypted message, thus containing information about the underlying encryption error. To capture this attack, Li and Micciancio proposed the notion of $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$, where the adversary is allowed to request decryptions of ciphertexts for which it knows the underlying message. Exact scheme instantiations with non-negligible probability of decryption failure (i.e. probability of decryption failure greater than $1/2^{\Omega(s)}$ for a statistical security parameter[21] s) are not exempt from similar attacks. Recent works [CSBB24, CCP+24, ML24] have proposed attacks on BFV, BGV, DM, and CGGI, which work by exploiting potential decryption errors[22].

There have been several measures proposed to counteract this type of attack. In the case of CKKS, the most common technique is noise flooding [LM21, LMSS22], which consists of adding a large noise in $2^{\Omega(s)}$ to the message during the decryption step, effectively hiding the noise. Other mitigations such as rounding and adding a deterministic noise have also been proposed [LM21] and implemented in several libraries [CHK20]. For exact encryption schemes, the attack can be mitigated by reducing the probability of decryption failure to negligible levels (i.e., less than $1/2^{\Omega(s)}$). Further attacks against provably $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$ secure instantiations have been proposed in [CSBB24, CCP+24, GNSJ24], and countermeasures have been proposed in [ABMP24, BCM+24, ML24].

The development of definitions and methods to model and guarantee security for FHE schemes is currently an active area of research, and is beyond the scope of this paper. Hence, in this work we mainly focus on providing (computational) IND-CPA security for FHE. We leave the consideration of advanced security notions for future work.

4 Concrete security estimation

In this section we state the security levels that the parameter sets in Section 5.1 target, and we outline the assumptions under which we give estimates for the concrete security of those parameter sets.

4.1 Security Levels

We define three classical security levels according to the NIST Special Publication 800-57 Part 1 [Bar20], 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.

[21] We use the notation s here to distinguish from the computational security parameter λ that is used elsewhere in the paper. See e.g. [LMSS22] for further details of the statistical security parameter in this context.

[22] Other attacks exploiting decryption failure in cryptography more broadly, and for lattice-based cryptography and FHE specifically, had been previously known (see e.g. [HGS99, LMSV12, BDPS14, DGJ+19]).

4.2 The Lattice Estimator

We estimate concrete security of the FHE parameter sets given in Section 5.1 using the open-source Lattice Estimator tool [APS15a]. 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].

4.3 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]). For configuration in the Lattice Estimator, we choose `RC.MATZOV` [MAT22] as the cost model in the classical setting.

Quantum cost models. In a prior version of this work, we also considered a quantum sieving cost model to target security against adversaries with quantum computational resources. This presentation paralleled that of [ACC+19], who also gave tables developed using classical and quantum sieving cost models. After feedback from an earlier draft of this work, we decided to remove the parameter sets targeting specific security levels against quantum adversaries, whose concrete security was estimated using quantum sieving cost models. The main reason for this is that estimates in [AGPS20] of the concrete performance of quantum sieving algorithms indicates only a mild improvement over classical sieving even when very optimistic assumptions are made about the cost of quantum random memory access and quantum error correction. Indeed, it is shown in [JR23] that assuming quantum random access memory is cheap may be a very strong assumption. Moreover, it is argued in [AS22] that quantum algorithms "can effectively be ignored when setting parameters" in lattice-based cryptography.

This decision also makes Tables 5.2 and 5.3 easier to use: for example, in Table 5.2, there is now a clear maximal bitsize of ciphertext modulus for a fixed choice of ring dimension and secret distribution. As all our tables are reproducible, users can separately run estimates for any other cost model implemented in the Lattice Estimator, including a quantum sieving cost model, if so desired. To make this simpler, in the code that accompanies our work, we have included code for a quantum sieving estimate based on [CL21].

4.4 Computational cost metric

To assess whether we have met a target security level as defined in Section 4.1, 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 if the Lattice Estimator estimates that all algorithms cost greater than 2^{128} ring operations when using a classical lattice reduction cost model. Note that "ring operations" can be converted into CPU cycles for classical computers.

5 Tables of parameters

In this section, we provide examples of parameter sets for FHE, targeting security (Section 5.1) and functionality (Section 5.2). We also review the parameter selection support offered in some of the major open-source FHE libraries. The notation used in Sections 5.1 and 5.2 is summarised in Table 5.1.

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, CKKS, DM and CGGI.
q_{ks}	LWE modulus used for key switching in DM and CGGI when $\sigma = 3.19$.
Q	Largest modulus of the ciphertext space, for BGV, BFV, CKKS.
P	Auxiliary (hybrid key switching) modulus for BGV, BFV, CKKS, with $q = PQ$ bounded according to security level.
t	BGV/BFV/DM/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 multiplications supported.
d_{num}	Number of digits used for hybrid key switching.
Scaling Factor	CKKS scaling factor.
Base prime size	Smallest modulus of the ciphertext space for CKKS.

Table 5.1: Notation used in Tables 5.2, 5.3, 5.4, 5.6, 5.7 and 5.8.

5.1 Parameter sets that target particular security levels

In this section, we give in Table 5.2 and 5.3 examples of LWE parameter sets that can be used in FHE applications.

These LWE parameter sets target particular security levels as defined in Section 4.1 using the Lattice Estimator under the assumptions stated in Section 4.3 and 4.4. 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 (for dimension $N \leq 2^{14}$), and hybrid_dual using commit `8f1ff7e` of the Lattice Estimator, dated Aug 27, 2024.

We want to emphasize that these tables are estimated to meet the target security levels, under the assumptions we have outlined. The estimated security of these parameter sets may be impacted by future advancements in cryptanalysis. It may also be affected by implementation choices in the Lattice Estimator, such as the chosen cost model. We make available scripts that we used to generate the tables at `https://github.com/gong-cr/FHE-Security-Guidelines/`, which could be re-run with subsequent versions of the Lattice Estimator if desired.

Table 5.2 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, 192, and 256 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.

We note that the Lattice Estimator models all error distributions as Gaussians of a given standard deviation. So, using a different fixed error distribution with standard deviation close to $\sigma = 3.19$, such as a Centered Binomial Distribution resulting from 42 fair coin tosses centered at 0, would yield similar values for the maximal $\log_2(q)$ as in Table 5.2.

In the DM/CGGI setting, q is typically fixed to either 32-bit or 64-bit, and the error standard deviation can be varied. Thus, in Table 5.3, 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, and 256 security levels.

N	$\log_2(q)$	
	Ternary	Gaussian
	$\lambda = 128$	
1024	26	28
2048	53	55
4096	106	108
8192	214	216
16384	430	432
32768	868	870
65536	1747	1749
131072	3523	3525
	$\lambda = 192$	
2048	36	38
4096	73	75
8192	147	149
16384	297	299
32768	597	599
65536	1199	1201
131072	2411	2413
	$\lambda = 256$	
2048	27	30
4096	56	58
8192	114	116
16384	230	232
32768	462	464
65536	929	931
131072	1866	1868

Table 5.2: 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 security level categories 128, 192 and 256.

n	$\log_2(q)$	$\log_2(\sigma)$		
		Binary	Ternary	Gaussian
		$\lambda = 128$		
630	32	18.5	17.2	14.6
1024		8.3	7.1	4.6
≥ 2048		2.0	2.0	2.0
630	64	50.5	49.2	46.6
750		47.4	46.2	43.5
870		44.3	43.1	40.3
1024		40.3	39.1	36.4
2048		13.7	12.4	10.0
≥ 4096		2.0	2.0	2.0
		$\lambda = 192$		
750	32	22.1	20.8	17.9
1024		17.2	15.9	13.0
≥ 2048		2.0	2.0	2.0
750	64	54.1	52.8	49.9
870		52.0	50.6	47.7
1024		49.2	47.9	45.0
2048		30.9	29.5	26.5
≥ 4096		2.0	2.0	2.0
		$\lambda = 256$		
1024	32	21.8	20.5	17.4
2048		7.6	6.1	3.2
≥ 4096		2.0	2.0	2.0
1024	64	53.8	52.5	49.4
2048		39.6	38.1	35.0
4096		10.9	9.3	6.4
≥ 8192		2.0	2.0	2.0

Table 5.3: 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 security level categories 128, 192 and 256. Since DM and CGGI consider 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, the value of $\log_2(\sigma)$ should be adapted accordingly.

5.2 Functional parameter sets

In this section, we give examples of SHE and FHE parameters sets that could be used for BGV, BFV, CKKS, DM, or CGGI 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. They may not necessarily represent optimal configurations for the individual libraries, and they are not intended for comparison among libraries.

Functional parameters for BGV and BFV. Table 5.4 and 5.5 provide examples of parameter sets for (RNS variants of) BGV/BFV in an SHE setting, i.e., without bootstrapping. The parameters were estimated to illustrate the Category 128, 192, or 256 security levels. The notation used in both tables is described in Table 5.1. The parameters in Table 5.4 were generated[23] using Microsoft SEAL [SEA23]. The high-level procedure for generating Table 5.4 is to set the modulus q to the maximum value supported for a given ring dimension, and then find the maximum multiplicative depth that can be achieved by examining the noise budget after decryption. The parameters in Table 5.5 were generated using the cryptographic context generation API in OpenFHE v1.2.0[24]. The high-level idea is to allow the user to enter the main application specifications, such as multiplicative depth, plaintext modulus, and security level, and let the library estimator find appropriate lattice parameters. Note that the purpose of both tables is to illustrate the main considerations when selecting parameters, rather than providing optimized parameters for a given application. Table 5.5 also lists the values of d_{num}, the number of digits for hybrid key switching, which affects both the size of the maximal modulus $q = PQ$ and size of evaluation keys for multiplication and key switching. A higher value of d_{num} allows the user to reduce P, hence achieving the largest depth for a given ring dimension, but it also increases the evaluation key size and key switching runtime[25]. Hence, d_{num} is a configurable parameter that may be tailored to application needs.

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.

[23] Table 5.4 can be reproduced using a script available at https://github.com/WeiDaiWD/SEAL-Depth-Estimator.

[24] The OpenFHE cryptographic context generation capability finds parameters using the multiplicative depth, plaintext modulus, number of digits used for hybrid key switching (d_{num}), security level, desired scaling modulus size for BFV, and other parameters. These parameter sets can be reproduced using the scripts available at https://github.com/gong-cr/FHE-Security-Guidelines/.

[25] One evaluation key in this case has the size of d_{num} ciphertexts with modulus PQ and the key switching runtime is proportional to d_{num}; see [HK20] for more details.

[26] The depth L is conservatively chosen for both BGV and BFV to achieve negligible practical (via subgaussian analysis) decryption probability of failure by using the expansion factor of $2\sqrt{n}$; (see [KPZ21] for more details on parameter estimation for BGV and BFV in OpenFHE).

[27] For BGV, up to 5 additions and 3 key switching operations were allowed per level. The `FLEXIBLEAUTOEXT` scaling mode was used.

λ	128	192	256
$\log_2(n)$	14	15	16
$\log_2(q)$	424	585	920
$\log_2(t)$	20	20	20
χ_s	Ternary	Ternary	Ternary
σ (χ_e)	3.2	3.2	3.2
L (BFV)	10	14	23
L (BGV)	8	12	19

Table 5.4: Sample SEAL parameters for BFV/BGV without bootstrapping.

λ	128	192	256
χ_s	Ternary	Ternary	Ternary
σ (χ_e)	3.19	3.19	3.19
t	65537	65537	786433
$\log_2(n)$	14	15	16
BFV parameters			
L[26]	10	15	18
$\log_2(Q)$	360	531	720
$\log_2(P)$	60	60	180
$\log_2(PQ)$	420	591	900
d_{num}	6	9	4
BGV parameters			
L[27]	8	13	16
$\log_2(Q)$	337	532	686
$\log_2(P)$	60	60	240
$\log_2(PQ)$	397	592	926
d_{num}	10	15	4

Table 5.5: Sample OpenFHE parameters for BFV/BGV without bootstrapping.

Sample parameters for CGGI and DM. In Table 5.6 we present examples of parameters for CGGI and DM that are estimated to meet the Category 128 security level. Note that for DM we refer to the parameters for its optimized variant proposed in [LMK+23] and implemented in OpenFHE. The notation used in Table 5.6 is as defined in Table 5.1, 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 digit size and number of digits used in key-switching keys; and $(\beta_\mathsf{pbs}, \ell_\mathsf{pbs})$ denote the digit size and number of digits used in the bootstrapping keys. Finally, p_error denotes the error probability for a single bootstrapping operation. The TFHE-rs parameters in Table 5.6 were generated using the optimization techniques found in Concrete [BBB+23]. The OpenFHE parameters in Table 5.6 were found using the OpenFHE estimation tool for DM and CGGI variants[28].

[28] The OpenFHE parameters can be regenerated using the OpenFHE lattice estimator tool at https://github.com/openfheorg/openfhe-lattice-estimator (commit 4f9e143), which uses the Lattice Estimator for finding secure LWE parameters.

λ	128	128	128	128	128	128	128	128
Scheme	CGGI	CGGI	CGGI	CGGI	CGGI	CGGI	DM	DM
Library	TFHE-rs	TFHE-rs	Concrete	Concrete	OpenFHE	OpenFHE	OpenFHE	OpenFHE
n	841	785	805	687	503	556	447	556
$\log_2(N)$	11	9	11	9	10	10	10	10
k	1	4	1	3	1	1	1	1
q	2^{64}	2^{64}	2^{64}	2^{64}	$\approx 2^{27}$	$\approx 2^{27}$	$\approx 2^{28}$	$\approx 2^{27}$
q_{ks}	2^{64}	2^{64}	2^{64}	2^{64}	$\approx 2^{14}$	$\approx 2^{15}$	$\approx 2^{14}$	$\approx 2^{15}$
t	2^4	2	2^4	2	2	2	2	2
χ_{LWE}	Binary	Binary	Binary	Binary	Ternary	Ternary	Gaussian	Ternary
χ_{GLWE}	Binary	Binary	Binary	Binary	Ternary	Ternary	Gaussian	Ternary
β_{ks}	2^3	2^4	2^3	2^4	2^5	2^5	2^5	2^5
ℓ_{ks}	5	3	5	3	3	3	3	3
β_{pbs}	2^{22}	2^{23}	2^{15}	2^{18}	2^9	2^7	2^{10}	2^9
ℓ_{pbs}	1	1	2	1	3	4	3	3
σ_{LWE}	$2^{45.72}$	$2^{47.22}$	$2^{15.68}$	$2^{45.99}$	3.19	3.19	3.19	3.19
σ_{GLWE}	$2^{15.68}$	$2^{14.05}$	$2^{14.05}$	$2^{49.02}$	3.19	3.19	3.19	3.19
p_{error}	2^{-64}	2^{-64}	2^{-64}	2^{-64}	2^{-40}	2^{-220}	2^{-55}	2^{-120}

Table 5.6: Sample parameters for CGGI and DM. The first two parameter sets for CGGI (with $n = 742$ and 777) are taken from the TFHE-rs library[29]. The third and fourth parameter sets (with $n = 805$ and 687) are from the Concrete compiler. The fourth (with $n = 503$) and fifth (with $n = 556$) parameter sets are taken from the parameters recommended for the CGGI implementation in OpenFHE v1.2.0 [MP21, AAB+22]. Finally, the sixth (with $n = 447$) and seventh (with $n = 593$) correspond to the parameters recommended for the DM implementation in OpenFHE v1.2.0 [LMK+23, AAB+22]. Note that the failure probabilities p_{error} are computed using different techniques (see Appendix B for details). The parameter t, plaintext modulus, is sometimes also referred to as p in the literature.

Sample parameters for RNS-CKKS. In Table 5.7, respectively Table 5.8, we present example parameter sets for (an RNS variant) of CKKS without, respectively with, bootstrapping. The parameters in Table 5.7 are estimated to meet the Category 128, 192, or 256 levels of security. The parameters in Table 5.8 are estimated to meet the Category 128 level of security.

The parameters in Table 5.7 were selected using OpenFHE v1.2.0 [AAB+22]. The parameters in Table 5.8 are selected[30] using Lattigo v5.0.2 [Tun23][31] for Set I and using OpenFHE v1.2.0 [AAB+22] for Set II. The rescaling method for all OpenFHE parameter sets was set to `FLEXIBLEAUTO` and d_{num} was set to 3. Both libraries contain implementation of several bootstrapping algorithms, including [CHK+18, CCS19, HK20, BMTPH21, BCC+22].

[29] We note that the TFHE-rs parameter sets presented in Table 5.6 are not associated to a public script for reproducibility.

[30] Tables 5.7 and 5.8 can be reproduced using scripts available at https://github.com/gong-cr/FHE-Security-Guidelines/.

[31] Lattigo also provides support by default for the sparse secret encapsulation technique [BTPH22], but this feature was disabled to instead use a dense secret.

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 5.8 provides the number of consumed levels for the execution of each of these blocks.

λ	128	192	256
$\log_2(N)$	14	15	15
$\chi_{\mathbf{s}}$	Ternary	Ternary	Ternary
σ ($\chi_{\mathbf{e}}$)	3.19	3.19	3.19
Base Prime Size	40	43	40
L	7	9	7
$\log_2(PQ)$	427	592	434
$\log_2(Q)$	307	412	314
$\log_2(P)$	120	180	120
$\log_2$ (Scaling Factor)	38	41	39
Precision Bit	22.3	24.0	22.2

Table 5.7: Sample parameters for RNS-CKKS without bootstrapping

	Set I[32]	Set II[33]
λ	128	128
$\log_2(N)$	16	16
Number of Slots[34]	32768	32768
$\chi_{\mathbf{s}}$	Ternary	Ternary
σ ($\chi_{\mathbf{e}}$)	3.19	3.19
Base Prime Size	45	60
L (after bootstrapping)	10	6
$\log_2$(Scaling Factor)	35	58
$\log_2(PQ)$	1734	1691
$\log_2(Q)$	1464	1511
$\log_2(P)$	305	180
Level cost of SlotsToCoeffs	4	3
Level cost of EvalMod	12	13
$\log_2(\mathsf{Pr}[\lvert\lvert I(X)\rvert\rvert > K])$[35]	-37.65	-37.65
K	512	512
Level cost of CoeffsToSlots	3	3
Iterations[36]	1	1
Precision Bits[37]	15.9	12.0

Table 5.8: Sample parameters for RNS-CKKS with bootstrapping.

5.3 Parameter selection in 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 5.9 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 and compilers.

OpenFHE. OpenFHE [AAB+22] supports the schemes BFV, BGV, CGGI, CKKS and DM. 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[38] 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[39] 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[40] 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[41] to find appropriate parameters for a given FHE computation. It makes use of the Lattice Estimator to find the security level of the parameters.

[32] The scaling factor in this parameter set does not affect bootstrapping as Lattigo uses different independent internal scaling factors for each step of the bootstrapping circuit.

[33] OpenFHE automatically adds "small" flooding noise on top of existing approximation error as a mitigation for the case when the decryption result may be accidentally shared; this flooding noise slightly reduces the output precision.

[34] Number of Slots refers to the number of complex numbers that are encrypted in each separate ciphertext.

[35] Detailed explanation on this bootstrapping failure probability and the parameter K can be found in Appendix A.

[36] 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.

[37] Precision Bits are evaluated as the negative base 2 logarithm of the average L1 norm between results from standard (cleartext) calculation and those computed homomorphically.

[38] The relevant code can be found in files `bfvrns-parametergeneration.cpp`, `bgvrns-parametergeneration.cpp`, and `ckksrns-parametergeneration.cpp` (Retrieved from OpenFHE v1.2.0).

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

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

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

Library	Link	BFV	BGV	CKKS	CGGI/DM	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 5.9: Open-source homomorphic encryption libraries and the algorithms they support.

HECATE and ELASM. Besides EVA, there have been other efforts proposing automatic scale management schemes for CKKS 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. A survey of earlier FHE compiler works can be found in [VJH21].

6 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 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 noise 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 Martin Albrecht and Léo Ducas for helpful discussions on cost models, and Martin Albrecht for feedback on an earlier version of this document. The authors are also grateful to Andreea Alexandru, Ahmad Al Badawi, and Daniele Micciancio for the discussions related to security models and the implications of correctness for security in FHE schemes. The authors would like to thank Alexander Viand for the valuable discussion on automatic parameter management using compilers.

The authors would like to thank Jung Hee Cheon for his numerous contributions to this work including initiating the project and giving helpful suggestions and feedback throughout its development. The authors would also like to thank Bahattin Yildiz for his contributions to an earlier version of this work during a previous employment at Intel Labs.

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

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

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.

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

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.

AAB+22. Ahmad Al Badawi, Andreea Alexandru, Jack Bates, Flavio Bergamaschi, David Bruce Cousins, Saroja Erabelli, Nicholas Genise, Shai Halevi, Hamish Hunt, Andrey Kim, Yongwoo Lee, Zeyu Liu, Daniele Micciancio, Carlo Pascoe, Yuriy Polyakov, Ian Quah, Saraswathy R.V., Kurt Rohloff, Jonathan Saylor, Dmitriy Suponitsky, Matthew Triplett, Vinod Vaikuntanathan, and Vincent Zucca. OpenFHE: Open-source fully homomorphic encryption library. Cryptology ePrint Archive, Paper 2022/915, 2022.

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).

ABMP24. Andreea Alexandru, Ahmad Al Badawi, Daniele Micciancio, and Yuriy Polyakov. Application-aware approximate homomorphic encryption: Configuring fhe for practical use. Cryptology ePrint Archive, Paper 2024/203, 2024. https://eprint.iacr.org/2024/203.

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.

ACC+21. 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. In Kristin Lauter, Wei Dai, and Kim Laine, editors, *Protecting Privacy through Homomorphic Encryption*, pages 31–62. Springer International Publishing, Cham, 2021.

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.

AGHV22. Adi Akavia, Craig Gentry, Shai Halevi, and Margarita Vald. Achievable cca2 relaxation for homomorphic encryption. In Eike Kiltz and Vinod Vaikuntanathan, editors, *Theory of Cryptography*, pages 70–99, Cham, 2022. Springer Nature Switzerland.

AGPS20. Martin R. Albrecht, Vlad Gheorghiu, Eamonn W. Postlethwaite, and John M. Schanck. Estimating quantum speedups for lattice sieves. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part II*, volume 12492 of *Lecture Notes in Computer Science*, pages 583–613. Springer, 2020.

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.

APS15a. 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. https://github.com/malb/lattice-estimator.

APS15b. 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. https://bitbucket.org/malb/lwe-estimator/src/master/.

AS22. Martin R. Albrecht and Yixin Shen. Quantum augmented dual attack. *CoRR*, abs/2205.13983, 2022.

Bar20. Elaine Barker. Recommendation for key management: Part 1 – general. Technical Report NIST Special Publication 800-57 Part 1, Revision 5, U.S. Department of Commerce, Washington, D.C., 2020.

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.

BCM+24. Jean-Philippe Bossuat, Anamaria Costache, Christian Mouchet, Lea Nürnberger, and Juan Ramón Troncoso-Pastoriza. Practical q-IND-CPA-d-secure approximate homomorphic encryption. Cryptology ePrint Archive, Paper 2024/853, 2024.

BDF18. Guillaume Bonnoron, Léo Ducas, and Max Fillinger. Large fhe gates from tensored homomorphic accumulator. In Antoine Joux, Abderrahmane Nitaj, and Tajjeeddine Rachidi, editors, *Progress in Cryptology – AFRICACRYPT 2018*, pages 217–251, Cham, 2018. Springer International Publishing.

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.

BDPS14. Alexandra Boldyreva, Jean Paul Degabriele, Kenneth G. Paterson, and Martijn Stam. On symmetric encryption with distinguishable decryption failures. In Shiho Moriai, editor, *Fast Software Encryption*, pages 367–390, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

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.

BR15. Jean-François Biasse and Luis Ruiz. Fhew with efficient multibit bootstrapping. In Kristin Lauter and Francisco Rodríguez-Henríquez, editors, *Progress in Cryptology – LATINCRYPT 2015*, pages 119–135, Cham, 2015. Springer International Publishing.

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.

BSW12. Dan Boneh, Gil Segev, and Brent Waters. Targeted malleability: homomorphic encryption for restricted computations. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, page 350–366, New York, NY, USA, 2012. Association for Computing Machinery.

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. To appear at ACM CCS 2024.

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.

CHK20. Jung Hee Cheon, Seungwan Hong, and Duhyeong Kim. Remark on the security of CKKS scheme in practice. Cryptology ePrint Archive, Paper 2020/1581, 2020.

CHP23. José Cabrero-Holgueras and Sergio Pastrana. 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.

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.

CL21. André Chailloux and Johanna Loyer. Lattice sieving via quantum random walks. 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 63–91. Springer, 2021.

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 cpa^{d} security of "exact" and threshold FHE schemes and libraries. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2024, Proceedings, Part III*, volume 14922 of *Lecture Notes in Computer Science*, pages 3–33. Springer, 2024.

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.

CSY22. Jung Hee Cheon, Yongha Son, and Donggeon Yhee. Practical fhe parameters against lattice attacks. *J. Korean Math. Soc*, 59(1):35–51, 2022.

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.

DGJ+19. Jan-Pieter D'Anvers, Qian Guo, Thomas Johansson, Alexander Nilsson, Frederik Vercauteren, and Ingrid Verbauwhede. Decryption failure attacks on ind-cca secure lattice-based schemes. In Dongdai Lin and Kazue Sako, editors, *Public-Key Cryptography – PKC 2019*, pages 565–598, Cham, 2019. Springer International Publishing.

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.

DM15. Léo Ducas and Daniele Micciancio. Fhew: Bootstrapping homomorphic encryption in less than a second. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015*, pages 617–640, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

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.

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.

FHR22. Prastudy Fauzi, Martha Norberg Hovd, and Håvard Raddum. On the ind-cca1 security of fhe schemes. *Cryptography*, 6(1), 2022.

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. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 169–178. ACM, 2009.

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.

GKPV10. Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert, and Vinod Vaikuntanathan. Robustness of the learning with errors assumption. In Andrew Chi-Chih Yao, editor, *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings*, pages 230–240. Tsinghua University Press, 2010.

GNSJ24. Qian Guo, Denis Nabokov, Elias Suvanto, and Thomas Johansson. Key recovery attacks on approximate homomorphic encryption with Non-Worst-Case noise flooding countermeasures. In *33rd USENIX Security Symposium (USENIX Security 24)*, Philadelphia, PA, aug 2024. USENIX Association.

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

HGS99. Chris Hall, Ian Goldberg, and Bruce Schneier. Reaction attacks against several public-key cryptosystem. In Vijay Varadharajan and Yi Mu, editors, *Information and Communication Security*, pages 2–12, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

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.

JR23. Samuel Jaques and Arthur G. Rattew. Qram: A survey and critique, 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.

KMR24. Elena Kirshanova, Chiara Marcolla, and Sergi Rovira. Guidance for efficient selection of secure parameters for fully homomorphic encryption. In Serge Vaudenay and Christophe Petit, editors, *Progress in Cryptology - AFRICACRYPT 2024 - 15th International Conference on Cryptology in Africa, Douala, Cameroon, July 10-12, 2024, Proceedings*, volume 14861 of *Lecture Notes in Computer Science*, pages 376–400. Springer, 2024.

KPZ21. Andrey Kim, Yuriy Polyakov, and Vincent Zucca. Revisiting homomorphic encryption schemes for finite fields. In Mehdi Tibouchi and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2021*, pages 608–639, Cham, 2021. Springer International Publishing.

KS22. Kamil Kluczniak and Leonard Schild. Fdfb: Full domain functional bootstrapping towards practical fully homomorphic encryption. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2023(1):501–537, Nov. 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. To appear at ACM CCS 2024.

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.

LLW24. Eunmin Lee, Joohee Lee, and Yuntao Wang. Improved meet-LWE attack via ternary trees. Cryptology ePrint Archive, Paper 2024/824, 2024.

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.

LMK+23. Yongwoo Lee, Daniele Micciancio, Andrey Kim, Rakyong Choi, Maxim Deryabin, Jieun Eom, and Donghoon Yoo. Efficient FHEW bootstrapping with small evaluation keys, and applications to threshold homomorphic encryption. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023*, pages 227–256, Cham, 2023. Springer Nature Switzerland.

LMSS22. Baiyu Li, Daniele Micciancio, Mark Schultz, and Jessica Sorrell. Securing approximate homomorphic encryption using differential privacy. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology – CRYPTO 2022*, pages 560–589, Cham, 2022. Springer Nature Switzerland.

LMSV12. Jake Loftus, Alexander May, Nigel P. Smart, and Frederik Vercauteren. On cca-secure somewhat homomorphic encryption. In Ali Miri and Serge Vaudenay, editors, *Selected Areas in Cryptography*, pages 55–72, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

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. Accelerating BGV bootstrapping for large p using null polynomials over $\mathbb{Z}_{p^e}$. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology - EUROCRYPT 2024 - 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26-30, 2024, Proceedings, Part II*, volume 14652 of *Lecture Notes in Computer Science*, pages 403–432. Springer, 2024.

ML24. Guangsheng Ma and Hongbo Li. On the security of homomorphic encryption schemes with restricted decryption oracles. *Journal of Systems Science and Complexity*, 37:2240 – 2261, 2024.

MML+23. Johannes Mono, Chiara Marcolla, Georg Land, Tim Güneysu, and Najwa Aaraj. Finding and evaluating parameters for BGV. In Nadia El Mrabet, Luca De Feo, and Sylvain Duquesne, editors, *Progress in Cryptology - AFRICACRYPT 2023*, pages 370–394, Cham, 2023. Springer Nature Switzerland.

MN24. Mark Manulis and Jérôme Nguyen. Fully homomorphic encryption beyond ind-cca1 security: Integrity through verifiability. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology – EUROCRYPT 2024*, pages 63–93, Cham, 2024. Springer Nature Switzerland.

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, François Charton, and Kristin E. Lauter. The cool and the cruel: Separating hard parts of LWE secrets. In Serge Vaudenay and Christophe Petit, editors, *Progress in Cryptology - AFRICACRYPT 2024 - 15th International Conference on Cryptology in Africa, Douala, Cameroon, July 10-12, 2024, Proceedings*, volume 14861 of *Lecture Notes in Computer Science*, pages 428–453. Springer, 2024.

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.

oST24. National Institute of Standards and Technology. Module-lattice-based key-encapsulation mechanism standard. Technical Report Federal Information Processing Standards Publications (FIPS PUBS) 203 August 13, 2024, U.S. Department of Commerce, Washington, D.C., 2024.

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.

PHS19. Alice Pellet-Mary, Guillaume Hanrot, and Damien Stehlé. Approx-svp in ideal lattices with preprocessing. 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.

PS24. Amaury Pouly and Yixin Shen. Provable dual attacks on learning with errors. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology - EUROCRYPT 2024 - 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26-30, 2024, Proceedings, Part VI*, volume 14656 of *Lecture Notes in Computer Science*, pages 256–285. Springer, 2024.

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.

VJH21. Alexander Viand, Patrick Jattke, and Anwar Hithnawi. Sok: Fully homomorphic encryption compilers. In *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, May 2021.

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. In Qiang Tang and Vanessa Teague, editors, *Public-Key Cryptography - PKC 2024 - 27th IACR International Conference on Practice and Theory of Public-Key Cryptography, Sydney, NSW, Australia, April 15-17, 2024, Proceedings, Part III*, volume 14603 of *Lecture Notes in Computer Science*, pages 3–35. Springer, 2024.

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 5.8. 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 $d\frac{\kappa\sigma_h}{\sqrt{\mathsf{E}[h]+1}} \approx d\kappa\sqrt{1-p}$ if h deviates by $d\sigma_h$ of $\mathsf{E}[h]$[42]. Therefore
$$\mathsf{Pr}\big[h \leq \mathsf{E}[h] + d\sigma_h\big] = \mathsf{erf}\left(\frac{d\sigma_h}{\sqrt{2}\sigma_h}\right) = \mathsf{erf}\left(\frac{d}{\sqrt{2}}\right).$$

[42] 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 $K \coloneqq K + K'$.

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 $\delta = \log_2(\mathsf{Pr}[\|I(Y)\| > K])$.
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 $\mathsf{Pr}[\|I(Y)\| > K]) \leq 2^{\delta}$.

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.

B CGGI/DM bootstrapping failure probability

In this Appendix we give more details about the failure probability in CGGI/DM bootstrapping, as mentioned in Table 5.6.

The OpenFHE bootstrapping failure probability estimation method is taken from [MP21]. The correctness of OpenFHE parameters was checked using numerical experiments. The fresh ciphertexts were pre-bootstrapped before performing any Boolean operations to estimate the error for the case of independently refreshed ciphertexts. For each parameter set, we recorded the actual values of the error/noise for a relatively large sample (1,000 bootstrapping runs), and then estimated the standard deviation of the error β_{exp}. Assuming the normal distribution of the error, we estimated the decryption failure probability, i.e., the probability of the error exceeding $q/8$, for both DM/LMK+ and CGGI cryptosystems. Since we need to support one homomorphic addition for Boolean gates, we estimated the probability of decryption failure as $1 - \text{erf}(\frac{q/8}{2\beta_{\text{exp}}})$.

In TFHE-rs and Concrete, the approach is similar, except that, due to the increased precision considered, $\frac{q}{8}$ is replaced by $\frac{q}{2^{\log_2(t)+2}}$. Here, t is the size of the plaintext space. This can be seen to match the above by setting $t = 2$.