J. Korean Math. Soc. **59** (2022), No. 1, pp. 35–51
https://doi.org/10.4134/JKMS.j200650
pISSN: 0304-9914 / eISSN: 2234-3008

PRACTICAL FHE PARAMETERS AGAINST LATTICE ATTACKS

Jung Hee Cheon, Yongha Son, and Donggeon Yhee

Abstract. We give secure parameter suggestions to use sparse secret vectors in LWE based encryption schemes. This should replace existing security parameters, because homomorphic encryption (HE) schemes use quite different variables from the existing parameters. In particular, HE schemes using sparse secrets should be supported by experimental analysis, here we summarize existing attacks to be considered and security levels for each attacks. Based on the analysis and experiments, we compute optimal scaling factors for CKKS.

1. Introduction

Homomorphic encryption (HE) is a cryptosystem that allows computations on encrypted data without decryptions. HE has an advantage in data science with privacy preserving. For example, data outsourcing services often handle confidential data so that they need a data securing scheme which enables operations in secure states [22,23,29]. Since the use cases are public services, a standardization for HE is required. HomomorphicEncryption.org is a consortium motivated by the needs, they summarize the reason for HE requirements [5] and make efforts for standard suggestions for API, secure parameters, etc. [8,10].

To ensure the security, HE uses computationally hard math problems. One of such problems is LWE, which is roughly a distinguishing problem asking whether a pair of a matrix and a vector is randomly given or is given by an approximately linear relation. An important assumption for LWE is that the distinguishing problem is computationally hard to solve [26]. RLWE, a special version of LWE using algebraic integers instead of a matrix and a vector, is also used but has not been proved yet as a hard problem.

As far as the authors know, currently major HE libraries are constructed based on LWE and RLWE, for example, HEaaN, HElib, SEAL, Paradise, and so on. Thus the security analysis are also based on the analysis for LWE and RLWE. LWE-estimator : `https://bitbucket.org/malb/lwe-estimator/`

Received December 1, 2020; Revised August 14, 2021; Accepted November 2, 2021.
2010 *Mathematics Subject Classification.* 94A60.
Key words and phrases. Fully homomorphic encryption, sparse secrets, hybrid attacks.

`src/master/` is used for experiments on parameters, and the above white papers [5, 8, 10] issued by HomomorphicEncryption.org are written based on the experiments. Unfortunately, the analysis is not enough because real schemes use different distributions for secret vectors.

Small secrets and bootstrappings. In original LWE and above standard suggestions, secret vectors are chosen by the uniform random distribution on a modulus vector space. On the other hands, the mentioned HE schemes often choose secret vectors in a sparse distribution, though it is naturally expected that the security will be harmed due to the reduction of possible choices of secret vectors.

There are two reasons why FHE choose sparse secrets despite being possibly threatened. The first reason is to enable encrypted multiplications. If a secret polynomial is large with respect to L^2-norm on its coefficient vector, then multiplication errors derived from encrypted multiplications is too large in the resulting ciphertext to preserve corresponding plaintext. The second reason is bootstrapping procedure. In asymptotic analysis of bootstrapping costs for each schemes, the expected costs are given according to the degree of the polynomials for the uniform ternary secret distribution and according to the hamming weight of the coefficients vectors for a sparse secret distribution [11, 12, 14].

1.1. Our contributions

We aim to suggest secure parameter choices for LWE based FHE against known attacks. For the attacks, their complexities are analyzed by reductions to SIS (shortest integer solution) problem and we estimate them by experiments using BKZ algorithm. These new suggestions have to replace existing suggestions [3] which are vulnerable to new attacks.[1] In particular, *sparsity* of secret keys causes a new kind of attacks so that existing parameters are not secure any more. Based on our experiments, we suggest secure parameters for RLWE against all known attacks.

In addition to the suggestions, we analyze how sparsity affects a maximal depth of circuits for CKKS scheme before bootstrapping. And then we compute available depth for given parameters. The computations consider 2 cases for reasonable error increases at each homomorphic operations.

Methodology. We combine two methods - brute force checks for small parameters and asymptotic estimations for large parameters.

To measure attack complexities, it is not effective to run the attack algorithms directly for huge parameters. For example, if one tries in brute force to check a parameter satisfying 128-bit security against an attack algorithm, it needs 2^{128} trials on average. This is not realistic.

[1]The existing parameters do not reflect usual distribution for secret keys, called *sparse distribution.*

Each attack is already given with its asymptotic complexity, so we take experiments for small parameters and compute real complexities for huge parameters according to the asymptotic complexity analysis. In this paper, we mainly use BKZ algorithm as the method. The major factor determining complexity is the dimension of a given vector space (or the rank of a given ring of integers). BKZ algorithm is a strategy that seperates the vector space into small dimensional subspaces (called blocks) and search certain kind of vectors[2] in each blocks. Then using the vectors, one can run LLL algorithm more effectively. There are various applications of BKZ algorithm, we mainly refer to [13] for using BKZ.

1.2. Related works

We mainly refer [2, 4, 24] for analysis of hard lattice problems and [15, 17] for most recent attacks on LWE instances. The works are purposed to analyze attack algorithms against given LWE samples, which become a basis for secure parameter selections.

In addition to the references, we refer a paper in similar purpose of ours: Curtis and Player have reported security analysis on sparse secret distribution with maximum lattice dimension 2^{17} [19]. It also suggests secure parameters taking hybrid methods into account. We will compare the analysis to ours in Section 3.3.

Acknowledgement. This work is done before the second author is employed by Samsung SDS. The first author is supported by the National Research Foundation of Korea(NRF) Grant funded by the Korean Government(MSIT) (No.2017R1A5A1015626) and Samsung Electronics Co., Ltd(IO201209-07883-01). The third author is supported by the National Research Foundation of Korea(NRF) Grant funded by the Korean Government(MSIT)(No.2017R1A5A 1015626).

2. Background

At first, notions should be setup. A bold lower case letter (e.g., $\mathbf{a}, \mathbf{b}, \ldots$) will denote a column vector and a bold upper case letter (e.g., $\mathbf{A}, \mathbf{B}, \ldots$) will be a matrix.

Definition 1 (Sparse secret distributions)**.** Let h be a positive integer. An n-dimensional vector is said to follow a *sparse secret distribution* of hamming weight h if exactly h components are nonzero. If the nonzero components are chosen in $\{-1, 1\}$ and -1 and 1 are equally chosen, it is said to follow a *ternary sparse secret distribution.*

Definition 2 (Distinguishing LWE problem)**.** Let ϕ be a non-uniform distribution on $\mathbb{Z}_q$. Let $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{\ell \times n} \times \mathbb{Z}_q^{\ell}$ be given with $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}$ for some $\mathbf{s} \in \mathbb{Z}_q^n$

[2]In our cases, the algorithm seeks after short vectors.

and $\mathbf{e} \xleftarrow{\phi^\ell} \mathbb{Z}_q^\ell$ or $\mathbf{b} \in \mathbb{Z}_q^\ell$ uniformly random. A *distinguishing LWE problem* is a question whether $\mathbf{b}$ is given by $\mathbf{As} + \mathbf{e}$ or given randomly. A *search LWE problem* is to find $\mathbf{s} \in \mathbb{Z}_q^n$.

In Regev's paper, a distinguishing LWE problem is given with uniform randomly chosen $\mathbf{s}$, while we consider a sparse $\mathbf{s}$ in this paper.

Definition 3 (RLWE)**.** Let $R_q := \mathbb{Z}_q[x]/\langle \Phi(x) \rangle$ be a polynomial ring with a modulus q and an irreducible polynomial $\Phi(x)$ of degree n. Let ϕ be a non-uniform distribution on R_q. Let $(a(x), b(x)) \in R_q \times R_q$ be given with $b(x) = a(x)s(x) + e(x)$ for some $s(x) \in R_q$ and $e(x) \xleftarrow{\phi} R_q$ or $b(x) \in R_q$ uniformly random. A *distinguishing RLWE problem* is a question whether $b(x)$ is given $a(x)s(x) + e(x)$ or given randomly. A *search RLWE problem* is to find $s(x) \in R_q$.

2.1. Homomorphic encryption

Homomorphic encryption is a form of an encryption system which enables computations on encrypted data without decrypting the data. We briefly review notions for homomorphic encryption to be used in this paper.

Definition 4 (Leveled homomorphic encryption)**.** For given parameters, a *leveled homomorphic encryption scheme* consists of 4 algorithms;

- Enc, which turns a plaintext into a ciphertext of a given level L.
- Dec, which turns a ciphertext of any level into a plaintext so that
$$\mathsf{Dec}(\mathsf{Enc}(m)) = m$$
for any plaintext m.
- Add, which turns two ciphertexts of same level into a ciphertext of the level so that
$$\mathsf{Dec}(\mathsf{Add}(\mathsf{Enc}(m_1), \mathsf{Enc}(m_2))) = m_1 + m_2.$$
- Mult, which turns two ciphertexts of same level L' into a ciphertext of level $L' - 1$ so that
$$\mathsf{Dec}(\mathsf{Mult}(\mathsf{Enc}(m_1), \mathsf{Enc}(m_2))) = m_1 m_2,$$
where $0 < L' \leq L$.

Definition 5 (Bootstrapping)**.** Bootstrapping is an algorithm that refreshes the level of a given ciphertext.

2.2. Dual lattice attack

This attack strategy solves LWE by converting it to a short integer solution (SIS) problem [1, 21]. The purpose is, for given $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$, to distinguish whether $(\mathbf{A}, \mathbf{b})$ follows an LWE distribution or uniformly random distribution [2, 25].

Applying the attack, one finds a short vector $\vec{y}$ in a *dual lattice* defined by

$$L_q^{\perp} := \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{x}^T\mathbf{A} \equiv 0 \mod q\}.$$

If $(\mathbf{A}, \vec{b})$ is sampled from LWE distribution, it holds that $\langle \mathbf{y}, \mathbf{b} \rangle \equiv_q \langle \mathbf{y}, \mathbf{e} \rangle$ and hence is likely to be small. Otherwise, $\langle \mathbf{y}, \mathbf{b} \rangle$ is uniformly distributed over $\mathbb{Z}_q$. So the value of the pairing determines whether $\mathbf{b}$ follows LWE in high probability (if the value is small) or clearly not (if the value is not small).

2.3. Primal uSVP attack

This attack strategy aims to find the secret vector $\vec{s}$ directly from given sample $(\mathbf{A}, \vec{b})$. For that, search version of LWE can be solved by finding a unique shortest vector in a lattice generated by column vectors of

$$\mathbf{B} = \begin{pmatrix} q\mathbf{I_m} & \mathbf{A} & \mathbf{b} \\ 0 & q\mathbf{I_n} & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

If the samples are given by an LWE distribution, then the lattice contains a vector $\mathbf{v}$; $\mathbf{v}^{\mathrm{T}} = (\mathbf{e}^{\mathrm{T}}, \mathbf{s}^{\mathrm{T}}, -\mathbf{1}) \in \mathbb{Z}^m \times \mathbb{Z}^n \times \mathbb{Z}$. BKZ algorithm with some blocksize β experimentally succeed to find such $\mathbf{v}$ [4], though it is not guaranteed in general.

3. Security level according to hamming weights

In this section, we suggest new parameters for the rank N of base ring and ciphertext modulus q. The parameters are chosen against new attacks that use the sparsity of secret keys.

During a multiplication using RLWE-based HE schemes, noises in ciphertexts are amplified by the size of coefficients of the secret polynomial. As the secret polynomial has larger coefficients, after fewer multiplications the noise overflows the ciphertext modulus [7,16] or spoils messages [16,20]. The increase of noises gives a reason to use only small coefficients for a secret polynomial.

All known FHE schemes are realized with bootstrapping technique. In general, bootstrapping is the most time-consuming part in FHE. Since the running time of bootstrapping is highly sensitive to the size of the secret polynomial [11, 14], not only a small bound on the size of coefficients, but also almost all coefficients of a secret polynomial are chosen to be 0.

These two reasons - multiplication in encrypted state and bootstrapping - justify the use of a small and sparse distribution on coefficients of a secret polynomial. However, the narrow distribution on secret polynomials may give an advantage to adversarial attackers because the secret polynomial is easier to be found than a uniformly chosen polynomial.

3.1. An attack against sparsity: Hybrid method

Hybrid method is a combination of a deterministic attack and meet-in-the-middle attack (MITM) to LWE samples. MITM is a guessing strategy that seperates a secret $\mathbf{s} \in \mathbb{Z}_q^n$ into two part $(\mathbf{s_g}, \mathbf{s}') \in \mathbb{Z}_q^g \times \mathbb{Z}_q^{n-g}$. The cost of this attack is proportional to the square root of the number of candidate secret vectors. The method is less sensitive to the absolute size of error when the ratio of error and modulus is sufficiently small. The strategy is a reduction not of lattices but of the dimension, so primal attack and dual attack can be combined with.

We briefly describe hybrid algorithm of MITM and primal attack [9, 17, 28] and of MITM and dual attack [15].

Hybrid-Primal attack. The strategy is finding a short vector

$$\mathbf{v} = \begin{pmatrix} \mathbf{v}' \\ \mathbf{v}_g \end{pmatrix} = \mathbf{B} \begin{pmatrix} \mathbf{x} \\ \mathbf{v}_g \end{pmatrix}$$

for some $g \in \mathbb{N}$ and $\mathbf{v}_g \in \mathbb{Z}_q^g$. $\mathbf{B}$ is written in a form $\left(\begin{smallmatrix} \mathbf{T} & \mathbf{C} \\ 0 & \mathbf{I}_g \end{smallmatrix}\right)$ and $\mathbf{v}' = \mathbf{Tx} + \mathbf{Cv}_g$. Then finding short $\mathbf{v}_g$ is as same as finding near point in the space generated by $\mathbf{T}$ to a point $\mathbf{Cv}_g$. A lower dimensional vector $\mathbf{v}_g$ is now in guessing, say $\mathbf{v}_g = \mathbf{v_1} + \mathbf{v_2}$. Since $\mathbf{Cv_1} = \mathbf{Cv}_g - \mathbf{Cv_2} = \mathbf{v}' - \mathbf{Tx} - \mathbf{Cv_2}$ so that finding a nearest point in T to $\mathbf{Cv_1}$ is as hard as finding one to $\mathbf{v}' - \mathbf{Cv_2}$.

Assuming that $\mathbf{T}$ is well-reduced so that v' is turned out to be the closest point to 0 in $v' + T$ by the Babai's nearest plane algorithms, it is expected

$$\mathbf{Cv_1} - [\mathbf{Cv_1}]_\mathbf{T} = -\mathbf{Cv_2} - [-\mathbf{Cv_2}]_\mathbf{T} + \mathbf{v}' \approx -\mathbf{Cv_2} + [\mathbf{Cv_2}]_\mathbf{T},$$

where $[\cdot]_T$ denotes the closest point in T to the given point. The hybrid strategy is trying to detect a collision between $\mathbf{Cv_1}$ and $\mathbf{Cv_2}$.

Hybrid-Dual attack. This strategy begins with parsing $\mathbf{A} = (\mathbf{A_1}|\mathbf{A_2})$ into two matrices with $\mathbf{A_1} \in \mathbb{Z}_q^{m\times(n-k)}$, $\mathbf{A_2} \in \mathbb{Z}_q^{m\times k}$, where k is a choice for lowering dimension. $\mathbf{s}$ is also considered as a joint of two vectors $(\mathbf{v_1}, \mathbf{v_2}) \in \mathbb{Z}_\mathbf{q}^{\mathrm{n-k}} \times \mathbb{Z}_\mathbf{q}^{\mathrm{k}}$. And then for a short

$$\mathbf{y} = (\mathbf{y_1}, \mathbf{y_2}) \in \{(\mathbf{v_1}, \mathbf{v_2}) \in \mathbb{Z}_\mathbf{q}^{\mathrm{m}} \times \mathrm{c}^{-1}\mathbb{Z}_\mathbf{q}^{\mathrm{n-k}} \mid \mathbf{v_1}^{\mathrm{T}}\mathbf{A_1} \equiv \mathrm{c}\mathbf{v_2} \bmod \mathbf{q}\},$$

$(\mathbf{y_1}^{\mathrm{T}}\mathbf{A_2}, \langle \mathbf{y}, \mathbf{b}\rangle)$ is a kind of LWE sample with a new secret vector $\mathbf{s_2} \in \mathbb{Z}_q^k$. In other words, given samples are reduced into LWE (candidate) samples of lower dimension with secret $\mathbf{s_2}$. If the new instance with k dimensional have small b_i's, then the original sample is given as an LWE instance. Otherwise, we apply MITM on the new instance and determine whether the original sample is chosen from LWE (Algorithm 3, [15]).

3.2. Attack complexity estimations

Our estimator[3] uses BKZ.sieve [6] in Sage for lattice reductions and for estimations of the costs of the reductions. The estimator returns feasible $\log Q$ value which is chosen to achieve security level λ against attacks above.

The experiments follow assumptions as in [17]. In particular, BKZ algorithm is considered with geometric series assumption and, during meet-in-the-middle step, the success probabilities are computed by Lemmas 4.1 and 4.2 in [17] to determine the number of guessing.

In our experiments, $\log Q$ is almost determined by the attack complexity of Hybrid-Primal attack which is written in **bold** style. There is one exception at $h = 64, \lambda = 256, \log N = 14$ which $\log Q$ is chosen against Hybrid-Dual attack.

Given two inputs $\log N, \lambda$, we search a candidate of maximal $\log Q$ that yields attack complexities less than 2^{λ}. It starts from a proper initial value of $\log Q$ (e.g. the twice of $\log Q$ for $\log N - 1$), and perform a sort of binary search. The beginning $\log Q$ can be chosen before running the estimator. In particular, we set an initial step value (e.g. quarter of the initial value), and if the initial $\log Q$ gives the minimal attack complexity larger (smaller, resp.) than 2^{λ}, then we add (subtract, resp.) the step value to $\log Q$ until the minimal attack complexity becomes smaller (larger, resp.) than 2^{λ}, and we continue this while halving the step value until the step value becomes 1.

Tables 1 and 2 in the following page are upper bounds for $\log Q$ and attack complexity of 4 algorithms. The experiments are established by algorithms given in a previous paper [15]. The columns of tables are given $\log N$, recommending $\log Q$ according to $\{\lambda, \log N, h\}$, and attack complexities.

Example. Security level λ means that attack complexity of all (known) attack must be at least 2^{λ}. This is often said λ bit complexity. If we choose $\log N = 17$ and $\log Q = 2022$, then the attack complexity of primal attack, dual lattice attack, Hybrid-Primal attack, and Hybrid-Dual attack are 173.0 bit, 147.6 bit, 128.9 bit, and 129.6 bit, respectively. In other words, $\log Q$ below 2022 is a secure parameter against the hybrid attacks for $\log N = 17$ and $\lambda = 128$.

3.3. Comparison to previous results

In this paper the authors compare their parameter suggestions to an existing parameter suggestion from the white paper of homomorphic encryption standardization consortium [3] and a previous discussion on the standardizing sparse LWE secrets [19].

Comparison to the white paper. Hybrid attacks are not considered in the white paper of homomorphic encryption standardization consortium [3]. For small hamming weight, the modulus Q should be chosen smaller than modulus given in the white paper.

[3]It can be accessed at https://github.com/Yongyongha/SparseLWE-estimator.

TABLE 1. An upper bound of $\log Q$ and attack complexity for $h = 64$

λ	$\log N$	$\log Q$	Primal	Dual	**Hybrid-Primal**	Hybrid-Dual
128	11	25	192.0	201.3	**132.1**	169.6
	12	52	186.8	199.6	**128.7**	147.6
	13	99	191.5	208.4	**132.5**	144.0
	14	219	183.0	180.6	**128.5**	133.7
	15	431	185.6	163.8	**132.8**	136.7
	16	930	179.6	152.8	**131.5**	133.4
	17	2022	173.0	147.6	**128.9**	129.6
192	12	20	278.5	395.8	**192.4**	259.7
	13	38	280.8	354.4	**192.8**	220.2
	14	79	276.5	384.9	**197.8**	209.9
	15	166	272.4	302.8	**192.0**	209.3
	16	352	268.2	247.5	**192.2**	207.6
	17	721	266.9	368.2	**201.7**	205.6
256	13	14	379.5	371.0	**256.9**	-
	14	47	325.9	623.7	284.3	**278.5**
	15	64	361.0	333.3	**258.8**	328.4
	16	122	366.9	338.7	**260.4**	314.5
	17	296	347.7	346.1	**256.7**	271.8

As Tables 1 and 2 show, the hybrid methods give an advantage to find a useful short vector in comparison with previous attacks. The advantage grows as smaller hamming weight is applied to the secret distribution.

In the other hands, in non-sparse distribution cases, the new attacks don't work in fact. Since the new attacks are initiated by a sparse distribution for choices of secrets, they have no advantage to uniformly chosen secrets. If one use a small uniform distribution, for example a binary uniform distribution as TFHE [18] or a ternary uniform distribution as SEAL [27], it is enough to follow existing parameters [3]. For example, our estimator with $h = \frac{N}{2}$ and $\frac{2N}{3}$, respectively, returns same parameter suggestion to a binary uniform distribution case and a ternary uniform distribution case, respectively.

Comparison to [19]. Curtis and Player present a conservative analysis of the Hybrid-Primal and Hybrid-Dual attacks for parameter sets of varying sparsity. The main difference to ours is that they assume that any probabilities associated to the meet-in-the-middle phase are set to 1, but the difference has only few effects on the choice of parameters. In Table 4, we compare the suggestions of Q in both papers. Our result has two advantages on secure parameter choices for sparse secrets. One is the practical suggestions for $\log N = 16, 17$, while [19] shows only estimations. Second one is a verification of the analysis on

TABLE 2. An upper bound of $\log Q$ and attack complexity for $h = 128$

λ	$\log N$	$\log Q$	Primal	Dual	**Hybrid-Primal**	Hybrid-Dual
128	11	42	163.3	153.8	**129.1**	160.8
	12	82	170.2	154.2	**129.2**	149.6
	13	165	171.4	150.7	**129.6**	139.3
	14	337	169.3	146.5	**129.0**	134.5
	15	700	164.0	142.9	**128.1**	131.8
	16	1450	159.0	140.9	**128.0**	128.5
	17	2900	160.2	141.8	**129.9**	130.4
192	12	44	285.1	259.0	**195.7**	242.7
	13	89	284.6	293.5	**195.0**	215.2
	14	178	286.3	245.3	**198.7**	209.8
	15	387	272.4	225.1	**194.6**	197.4
	16	804	266.8	222.1	**192.5**	196.1
	17	1650	263.4	220.2	**192.4**	194.0
256	13	43	419.1	464.9	**281.3**	347.3
	14	106	381.2	376.8	**258.2**	277.7
	15	209	385.2	369.4	**265.3**	282.8
	16	418	386.6	390.2	**271.1**	280.0
	17	942	366.0	311.2	**261.5**	265.9

TABLE 3. This table lists comparisons of $\log Q$ suggested by the white paper [3] and ours along overlapped $\log N$ and λ.

$\log N$	λ	white paper	ours($h = 128$)	ours($h = 64$)
13	128	218	165	99
	192	152	89	38
	256	118	43	14
14	128	438	337	219
	192	305	178	80
	256	237	106	47
15	128	881	700	431
	192	611	387	166
	256	476	209	64

success probabilities in meet-in-the-middle phase given by [17]: for a parameter set chosen tightly to the target security, e.g. $\log N = 15, h = 128, \lambda = 128$ with only 0.1 bit margin, our choice of Q has an advantage. Unfortunately, we don't find yet N, λ, h triples where the MITM success probabilities given by [17] has significant advantages.

TABLE 4. This table is the comparison table 4 in [19] and our associating choice of Q. The last column means a security margin from target security level λ.

h	target λ	$\log N$	$\log Q$ ([19])	$\log Q$ (ours)	(security - target)
		11	27	25	4.1
		12	55	52	0.7
64	128	13	111	99	4.5
		14	223	219	0.5
		15	496	431	4.8
		11	41	42	1.1
		12	83	82	1.2
128	128	13	171	165	1.6
		14	342	337	1.0
		15	**699**	**700**	**0.1**
		13	60	43	25.3
128	256	14	115	106	2.2
		15	263	209	9.3

4. Available depth for CKKS

Based on experimental observations and theoretical estimations, we will consider a more efficient set of parameters for CKKS scheme.

We give two tables according to certain assumptions on accumulation of errors. First assumption is assuming that errors are accumulated as large as possible, which may occur in squaring, i.e., a multiplication of same ciphtertext. This can be applied to any circuit. Second assumption is an expectation that errors grow on average, i.e., each HE operations take different input ciphertexts so that errors are not amplified so much. This is in fact an assumption on circuits using HE.

Remark 1. We may assume a third assumption that errors cancel each others in HE operations. In other words, error growth is bounded by a constant so that the number of operations is completely proportional to the modulus of ciphertexts. The third assumption is, however, an extreme one and it could be an unsubtential assumption for CKKS. In other hands, the assumption is realistic if the errors are seperated from plaintexts. In HElib a library using sparse secret distributions, the third assumption admits and the parameter choice can be applied to HElib.

4.1. CKKS overview

In CKKS scheme, a fresh ciphertext is given in modulus q which is said to be *top level*. To multiply ciphertexts, CKKS scheme publishes *evaluation key* in modulus Pq. The total security depends on Pq so that Pq is chosen as Q in previous section.

Scaling factor. Since a ciphertext space is discrete, a real valued data should be quantized before being encrypted. The unit of quantization is called scaling factor Δ. In a plaintext, a numerical data r is converted into a form $\lceil r \times \Delta \rfloor$ where rounding denotes the nearest integer. In other words, plaintexts or ciphertexts remember a scaled data $r \times \Delta$, not the plain data r.

Hamming weight and bootstrapping. We review why sparse secrets enable bootstrapping of CKKS scheme and an implement HEaaN.

Instead of uniform choice of a secret polynomial, sparse secret has an advantage in enabling bootstrapping process. The sparsity is given with ternary coefficients and is measured by the hamming weight. Under the condition that security level is satisfied, larger hamming weight implies heavier time-cost in bootstrapping and more bit-length $\log q$ of the modulus of ciphertexts. Since large modulus reduces the number of required bootstrappings, varying hamming weights is in fact a trade-off between the required number and running time of total bootstrappings in one circuit.

Briefly, we review the bootstrapping for CKKS [11, 14]. The bootstrapping procedure of CKKS scheme consists of 4 steps, saying $\mathsf{ModRaising}$, $\mathsf{CoeffToSlot}$, $\mathsf{SinEval}$, $\mathsf{SlotToCoeff}$. q denotes the modulus of input ciphertext.

(1) $\mathsf{ct} \to \mathsf{ct}'$ so that $\mathsf{Dec}(\mathsf{ct}') = m(x) + qI(x)$, where $m(x) = \mathsf{Dec}(\mathsf{ct})$. ct' has larger modulus than q.
(2) $\mathsf{Enc}(m(x) + qI(x)) \to \mathsf{Enc}(\{m(\zeta^{5^j}) + QI(\zeta^{5^j})\}_j)$.
(3) $\mathsf{Enc}(\{m(\zeta^{5^j}) + QI(\zeta^{5^j})\}_j) \xrightarrow{\text{encrypted mod } q} \mathsf{Enc}(\{m(\zeta^{5^j})\}_j)$.
(4) $\mathsf{Enc}(\{m(\zeta^{5^j})\}_j) \to \mathsf{Enc}(m(x))$.

Among them, $\mathsf{SinEval}$ is the most sensitive step to a variation of h. $\mathsf{SinEval}$ is in fact an alternative of an encrypted 'modulus q' - function, which is not implemented exactly yet. Instead of exact implementation of an encryption of modulus q function, we use an encryption of a polynomial which approximately become 'mod q' over certain region of $\mathbb{C}^{N/2}$. In particular, the range of $I(x)$ directly determines how high degree polynomial approximation is needed to preserve a certain precision of messages (Section 3 in [14]). Heuristically, the size $||I||_\infty$ is bounded as $O(\sqrt{h})$ (e.g. Section 5.3 in [14]).

4.2. Efficient scaling for depth of CKKS scheme.

We are assuming that the size of plaintext $||m(x)||_\infty \approx \Delta$, i.e., it is an encoding of scaling data which belongs to $\frac{1}{\Delta}\mathbb{Z}$ before scaling. Given n and hamming weight, upper bound of $\log q$ is determined. The bits length $\log q$ is the sum of the length of ModUp and the length of ciphertexts. The bits of the ciphertexts is again a sum of margin bits for rescaling, the precision size and the length of floating errors. In practical, we assume the length for ModUp is a quarter of the length of ciphertexts, i.e., we replace $\log q$ by $\frac{4}{5}\log q$. In previous section, estimated $\log q$ is now replaced by $\log Pq$ with ratio $\log P : \log q = 1 : 4$. The ratio 4 is the number of modulus switching keys.

We give Table 5 listing lower bound of $\log \Delta$ to ensure maximal level for three precision sizes $\log p = 10, 20, 30$. In the following table, small N is not listed because they don't admit enough large $\log q$ for a single multiplication preserving MSB at least $\log p$. Computations for the table refers to Appendix A.

$\log Pq$ is increased as h increases. In addition, maximum depth of multiplications without bootstrapping also increases.

Example. Let $h = 64$ be chosen and assume $\log p \geq 30$ is required. If $\log N = 17$, then available modulus q is about $1618 \approx \frac{2022}{5/4}$ bits. If $\log \Delta = 66$ is chosen, then after 21 steps of multiplications, 30 bits of MSB is ensured in the data. If $\log \Delta < 66$ and 21 levels are all consumed, then the MSB could be less then 30 bits in worst case. If $\log \Delta$ is too large, there is not enough modulus to be consumed 21 times. The bound is determined by $\log D \times (L+1) < \log q$.

TABLE 5. Maximum level of multiplications (worst case)

$\lambda = 128$, 4 keys for ModUp.

h	$\log p$	$\log N$	$\log Pq$	$\log \Delta$	max. level
64	10	14	219	29	5
		15	431	33	9
		16	930	42	17
		17	2022	54	29
	20	15	431	41	7
		16	930	49	14
		17	2022	60	25
	30	16	930	57	12
		17	2022	67	22
128	10	14	337	31	7
		15	700	38	13
		16	1450	48	23
		17	2900	62	36
	20	14	337	39	5
		15	700	46	11
		16	1450	55	20
		17	2900	68	32
	30	15	700	54	9
		16	1450	62	17
		17	2900	75	29
256	10	13	195	28	4
		14	393	33	8
		15	821	40	15
		16	1623	50	24
		17	3300	65	39
	20	14	393	41	6
		15	821	47	12
		16	1623	57	21
		17	3300	71	35
	30	14	393	50	5
		15	821	55	10
		16	1623	65	19
		17	3300	78	32

4.3. Under an assumption on error behavior

The worst case can occur when, for example, all the level is consumed for squarings of a ciphertext. It is the case that errors in each ciphertexts are same and each multiplication amplifies the error twice. In the other hands, circuits could be constructed to avoid such worst case as possible.

If homomorphic circuits are assumed that the error amplification is controlled, then more multiplications are available. This is exactly a question on circuit optimizations. The following Table 6 is a choice under an assumption that all errors are independent. In the case, m_1e_2 and m_2e_1 follow a random distribution independently and are bounded by ΔB_{enc}, so that $||m_1e_2+m_2e_1|| < \sqrt{2}\Delta B_{\mathsf{enc}}$. All the remaining computations are similar as the worst case, except that L is replaced by $L/2$ at the quadratic inequality (1) in Appendix A.

Table 6. Maximum level of multiplications (average case)

$\lambda = 128$, 4 keys for ModUp.

h	$\log p$	$\log N$	$\log Pq$	$\log \Delta$	max. level
64	10	14	219	26	5
		15	431	29	10
		16	930	35	20
		17	2000	43	36
	20	15	431	38	8
		16	930	43	16
		17	2000	51	31
	30	15	431	48	6
		16	930	51	13
		17	2000	58	26
128	10	13	165	26	4
		14	337	28	8
		15	700	33	16
		16	1450	39	28
		17	2900	49	46
	20	14	337	37	6
		15	700	41	12
		16	1450	47	23
		17	2900	56	40
	30	15	700	50	10
		16	1450	55	20
		17	2900	63	35
256	10	13	195	26	4
		14	393	29	9
		15	821	34	18
		16	1623	41	30
		17	3300	51	50
	20	14	393	38	7
		15	821	42	14
		16	1623	48	25
		17	3300	58	44
	30	14	393	47	5
		15	821	51	11
		16	1623	57	22
		17	3300	66	39

Appendix A. CKKS parameter estimating

As operations works in a circuit, in particular as multiplications runs, the null bits are consumed by scale factor Δ so that the possible number L of multiplication is limited by $\frac{\log q}{\log \Delta} - 1$. In particular, the length $\log q$ consists of $\log q = \log q_0 + L \log \Delta$ where $\log \Delta$ is the length consumed at each rescaling procedure and $\Delta < q_0 < \Delta^2$.

In CKKS, at each multiplication, error is estimated as follows:

(1) For two ciphertexts $\mathsf{ct}_1, \mathsf{ct}_2$ of level ℓ such that $\langle \mathsf{ct}_i, \mathsf{sk}\rangle = m_i + e_i$, $\mathsf{ct}_{\text{mult}}$ is a ciphertext of level ℓ such that

$$\langle \mathsf{ct}_{\text{mult}}, \mathsf{sk}\rangle = m_1 m_2 + (m_1 e_2 + m_2 e_1 + e_1 e_2 + e_{\text{mult}}),$$

where the latter term is new error. e_{mult} is the error occurring due to key-switching procedure.

(2) By rescaling $\mathsf{ct}_{\text{mult}}$, new ciphertext ct' of level $\ell - 1$ is obtained and

$$\langle \mathsf{ct}', \mathsf{sk}\rangle = \frac{1}{\Delta}(m_1 m_2 + (m_1 e_2 + m_2 e_1 + e_1 e_2 + e_{\text{mult}})) + e_{\mathsf{scale}}.$$

The scaling error e_{scale} arises due to a rounding operation in the rescaling.

In particular, as $e_1 e_2 + e_{\mathsf{ks}} < \Delta$, the scaling error contains those two terms.

The total error of a multiplication of two ciphertexts in a same level is bounded by

$$||e_1|| + ||e_2|| + e_{\mathsf{scale}} \leq 2 \times B_{\mathsf{enc}} + (1 + \frac{1}{\Delta})B_{\mathsf{scale}}$$

and this bound becomes new B_{enc} for the output ciphertext ct'.

Note that $(1 + \frac{1}{\Delta})B_{\mathsf{scale}}$ is independent to the level of ciphertexts. Let B_ℓ be an error bound for level ℓ ciphertexts. The final modulus q_0 is of bit length $\log q_0$ where the bit-length $\log \Delta$ is at least sum of the length of most significant bits(MSB) and the bit length of errors B_0. The MSB is in fact the precision digits of data.

Let $\log p$ be the length of MSB and B_{enc} be the error bound for fresh ciphertexts. $B_{\mathsf{enc}} = 8\sqrt{2}\sigma N + 6\sigma\sqrt{N} + 16\sigma\sqrt{hN}$ is determined by the dimension N of ciphertexts, hamming weight h, and the standard deviation σ of discrete Gaussian distribution used for LWE error (Lemma 1, [16]).

We have an equation $B_{\ell-1} = 2B_\ell + (1 + \frac{1}{\Delta})B_{\mathsf{scale}}$, where B_0 should be bounded by $B_0 + \frac{N}{2} < \frac{\Delta}{2^{p+1}}$ to be correctly decoded (Lemma 1, [16]). Therefore, Δ is chosen to be

$$\Delta > 2^{p+1}B_0 + 2^p N = 2^{L+p+1}B_{\mathsf{enc}} + (2^{L+p+1} - 2^{p+1})(1 + \frac{1}{\Delta})B_{\mathsf{scale}} + 2^p N$$

or

$$\Delta^2 - (2^{L+p+1}B_{\mathsf{enc}} + 2^p N + (2^{L+p+1} - 2^{p+1})B_{\mathsf{scale}})\Delta - (2^{L+p+1} - 2^{p+1})B_{\mathsf{scale}} > 0.$$

The latter quadratic inequality is equivalent to

$$\begin{aligned}\Delta > &(2^{L+p}B_{\mathsf{enc}} + 2^{p-1}N + (2^{L+p} - 2^p)B_{\mathsf{scale}}) \\ &+ \sqrt{(2^{L+p}B_{\mathsf{enc}} + 2^{p-1}N + (2^{L+p} - 2^p)B_{\mathsf{scale}})^2 + (2^{L+p+1} - 2^{p+1})B_{\mathsf{scale}}} \end{aligned} \tag{1}$$

due to $\Delta > 0$.

References

[1] M. Ajtai, *Generating hard instances of lattice problems (extended abstract)*, in Proceedings of the Twenty-eighth Annual ACM Symposium on the Theory of Computing (Philadelphia, PA, 1996), 99–108, ACM, New York, 1996. https://doi.org/10.1145/237814.237838

[2] M. R. Albrecht, *On dual lattice attacks against small-secret LWE and parameter choices in HElib and SEAL*, in Advances in cryptology—EUROCRYPT 2017. Part II, 103–129, Lecture Notes in Comput. Sci., 10211, Springer, Cham, 2017. https://link.springer.com/chapter/10.1007/978-3-319-56614-6_4

[3] M. Albrecht, M. Chase, H. Chen, J. Ding, S. Goldwasser, S. Gorbunov, S. Halevi, J. Hoffstein, K. Lauter, S. Lokam, et al., *Homomorphic encryption standard*, 2018.

[4] M. R. Albrecht, F. Göpfert, F. Virdia, and T. Wunderer, *Revisiting the expected cost of solving uSVP and applications to LWE*, in Advances in cryptology—ASIACRYPT 2017. Part I, 297–322, Lecture Notes in Comput. Sci., 10624, Springer, Cham, 2017. https://doi.org/10.1007/978-3-319-70694-8_11

[5] D. Archer, L. Chen, J.-H. Cheon, R. Gilad-Bachrach, R. A. Hallman, Z. Huang, X. Jiang, R. Kumaresan, B. A. Malin, H. Sofia, et al., *Applications of homomorphic encryption*, Technical report, https://homomorphicencryption.org, Redmond WA, USA, 2017.

[6] A. Becker, L. Ducas, N. Gama, and T. Laarhoven, *New directions in nearest neighbor searching with applications to lattice sieving*, in Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, 10–24, ACM, New York, 2016. https://doi.org/10.1137/1.9781611974331.ch2

[7] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, (*Leveled*) *fully homomorphic encryption without bootstrapping*, ACM Trans. Comput. Theory **6** (2014), no. 3, Art. 13, 36 pp. https://doi.org/10.1145/2633600

[8] M. Brenner, W. Dai, S. Halevi, K. Han, A. Jalali, M. Kim, K. Laine, A. Malozemoff, P. Paillier, Y. Polyakov, et al., *A standard API for RLWE-based homomorphic encryption*, Technical report, https://homomorphicencryption.org, Redmond WA, USA, 2017.

[9] J. Buchmann, F. Göpfert, R. Player, and T. Wunderer, *On the hardness of LWE with binary error: revisiting the hybrid lattice-reduction and meet-in-the-middle attack*, in Progress in cryptology—AFRICACRYPT 2016, 24–43, Lecture Notes in Comput. Sci., 9646, Springer, 2016. https://doi.org/10.1007/978-3-319-31517-1_2

[10] M. Chase, H. Chen, J. Ding, S. Goldwasser, S. Gorbunov, J. Hoffstein, K. Lauter, S. Lokam, D. Moody, T. Morrison, et al., *Security of homomorphic encryption*, Technical report, https://homomorphicencryption.org, Redmond WA, USA, 2017.

[11] H. Chen, I. Chillotti, and Y. Song, *Improved bootstrapping for approximate homomorphic encryption*, in Advances in cryptology—EUROCRYPT 2019. Part II, 34–54, Lecture Notes in Comput. Sci., 11477, Springer, Cham, 2019. https://doi.org/10.1007/978-3-030-17656-3_2

[12] H. Chen and K. Han, *Homomorphic lower digits removal an improved FHE bootstrapping*, in Advances in cryptology—EUROCRYPT 2018. Part I, 315–337, Lecture Notes in Comput. Sci., 10820, Springer, Cham, 2018. https://doi.org/10.1007/978-3-319-78381-9_12

[13] Y. Chen and P. Q. Nguyen, *BKZ 2.0: better lattice security estimates*, in Advances in cryptology—ASIACRYPT 2011, 1–20, Lecture Notes in Comput. Sci., 7073, Springer, Heidelberg, 2011. `https://doi.org/10.1007/978-3-642-25385-0_1`

[14] J. H. Cheon, K. Han, A. Kim, M. Kim, and Y. Song, *Bootstrapping for approximate homomorphic encryption*, in Advances in cryptology—EUROCRYPT 2018. Part I, 360–384, Lecture Notes in Comput. Sci., 10820, Springer, Cham, 2018. `https://doi.org/10.1007/978-3-319-78381-9_14`

[15] J. H. Cheon, M. Hhan, S. Hong, and Y. Son, *A hybrid of dual and meet-in-the-middle attack on sparse and ternary secret LWE*, IEEE Access **7** (2019), 89497–89506.

[16] J. H. Cheon, A. Kim, M. Kim, and Y. Song, *Homomorphic encryption for arithmetic of approximate numbers*, in Advances in cryptology—ASIACRYPT 2017. Part I, 409–437, Lecture Notes in Comput. Sci., 10624, Springer, Cham, 2017. `https://doi.org/10.1007/978-3-319-70694-8_15`

[17] J. H. Cheon and Y. Son, *Revisiting the hybrid attack on sparse and ternary secret LWE*, IACR Cryptol. ePrint Arch. **2019** (2019), 1019.

[18] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène, *TFHE: fast fully homomorphic encryption library*, August 2016.

[19] B. R. Curtis and R. Player, *On the feasibility and impact of standardising sparse-secret LWE parmeter sets for homomorphic encryption*, In roceedings of the 7th ACM Workshop on Encrypted Computing & Applied Homomorphic Cryptography, pages 1–10, 2019.

[20] J. Fan and F. Vercauteren, *Somewhat practical fully homomorphic encryption*, IACR Cryptol. ePrint Arch. **2012** (2012), 144.

[21] C. Gentry, *Fully homomorphic encryption using ideal lattices*, in STOC'09—Proceedings of the 2009 ACM International Symposium on Theory of Computing, 169–178, ACM, New York, 2009.

[22] M. Ibtihal, N. Hassan, et al., *Homomorphic encryption as a service for outsourced images in mobile cloud computing environment*, In Cryptography: Breakthroughs in Research and Practice, pages 316–330. IGI Global, 2020.

[23] P. Li, J. Li, Z. Huang, C.-Z. Gao, W.-B. Chen, and K. Chen, *Privacy-preserving outsourced classification in cloud computing*, Cluster Computing **21** (2018), no. 1, 277–286.

[24] R. Lindner and C. Peikert, *Better key sizes (and attacks) for LWE-based encryption*, in Topics in cryptology—CT-RSA 2011, 319–339, Lecture Notes in Comput. Sci., 6558, Springer, Heidelberg, 2011. `https://doi.org/10.1007/978-3-642-19074-2_21`

[25] D. Micciancio and O. Regev, *Lattice-based cryptography*, in Post-quantum cryptography, 147–191, Springer, Berlin. `https://doi.org/10.1007/978-3-540-88702-7_5`

[26] O. Regev, *On lattices, learning with errors, random linear codes, and cryptography*, J. ACM **56** (2009), no. 6, Art. 34, 40 pp. `https://doi.org/10.1145/1568318.1568324`

[27] *Microsoft SEAL (release 3.6)*, `https://github.com/Microsoft/SEAL`, Microsoft Research, Redmond, WA, 2020.

[28] T. Wunderer, *Revisiting the hybrid attack: Improved analysis and refined security estimates*, IACR Cryptol. ePrint Arch. **2016** (2016), 733.

[29] Y. Zhang, W. Dai, X. Jiang, H. Xiong, and S. Wang, *FORESEE: Fully Outsourced secuRe gEnome Study basEd on homomorphic Encryption*, In BMC medical informatics and decision making, volume 15, page S5. Springer, 2015.

Jung Hee Cheon
Department of Mathematics
Seoul National University
Seoul 08826, Korea
Email address: `jhcheon@gmail.com`

Yongha Son
Security Reseach Center
Samsung SDS
Seoul 06765, Korea
Email address: yongyonghaa@gmail.com

Donggeon Yhee
Industrial and Mathematical Data Analytics Research Center
Seoul National University
Seoul 08826, Korea
Email address: dgyhee@gmail.com