Improved Security for a Ring-Based Fully Homomorphic Encryption Scheme

Joppe W. Bos[1], Kristin Lauter[1], Jake Loftus[2⋆], and Michael Naehrig[1]

[1] Microsoft Research,
{jbos,klauter,mnaehrig}@microsoft.com
[2] University of Bristol,
loftus@cs.bris.ac.uk

Abstract. In 1996, Hoffstein, Pipher and Silverman introduced an efficient lattice based encryption scheme dubbed NTRUEncrypt. Unfortunately, this scheme lacks a proof of security. However, in 2011, Stehlé and Steinfeld showed how to modify NTRUEncrypt to reduce security to standard problems in ideal lattices. At STOC 2012, López-Alt, Tromer and Vaikuntanathan proposed a fully homomorphic scheme based on this modified system. However, to allow homomorphic operations and prove security, a non-standard assumption is required in their scheme. In this paper, we show how to remove this non-standard assumption via techniques introduced by Brakerski at CRYPTO 2012 and construct a new fully homomorphic encryption scheme from the Stehlé and Steinfeld version based on standard lattice assumptions and a circular security assumption. The scheme is scale-invariant and therefore avoids modulus switching, it eliminates ciphertext expansion in homomorphic multiplication, and the size of ciphertexts is one ring element. Moreover, we present a practical variant of our scheme, which is secure under stronger assumptions, along with parameter recommendations and promising implementation results. Finally, we present a novel approach for encrypting larger input sizes by applying a CRT approach on the input space.

Keywords: Leveled homomorphic encryption, fully homomorphic encryption, ring learning with errors.

1 Introduction

Fully homomorphic encryption (FHE) is a powerful form of encryption which allows an untrusted server to carry out arbitrary computation on encrypted data on behalf of a client. Introduced in [20] by Adleman, Dertouzos and Rivest, the problem of constructing a scheme which could evaluate any function on encrypted data remained open until 2009, when Gentry constructed an FHE scheme based on ideal lattices [9]. Gentry's scheme effectively laid down a blueprint for constructing FHE schemes and paved the way for many further constructions [25,2,3,5,4,22,19,10,8]. The main focus of the cryptologic research community has been on improving the efficiency of FHE and basing its security on standard assumptions.

Recently, López-Alt et al. [15] proposed an FHE scheme based on the work by Stehlé and Steinfeld [23] in which a provably secure version of NTRUEncrypt [12] is presented with security based on standard problems in ideal lattices. Unfortunately, the FHE scheme from [15] needs to make an additional assumption relating to the uniformity of the public key, the so-called decisional small polynomial ratio (DSPR) assumption, to allow homomorphic operations and remain semantically secure. We show how to avoid this additional assumption and transform the results from [23] into a fully homomorphic encryption scheme based on standard lattice assumptions only. This is achieved by limiting noise growth during homomorphic operations

⋆ Most of this work was done while the third author was an intern in the Cryptography Research group at Microsoft Research.

via a tensoring technique recently introduced by Brakerski [2]. Besides this theoretical advantage, our scheme has other attractive properties. Firstly, this new scheme is *scale-invariant* in the sense of [2], i.e. it avoids the modulus-switching technique of Brakerski, Gentry and Vaikuntanathan [3]. Secondly, we cut the ciphertext size in half, for comparable parameters, since a ciphertext consists of only a single ring element as opposed to the two or more ring elements for schemes based purely on the (ring) learning with errors (RLWE) assumption [16]. Finally, we present a technique to reduce the overall complexity incurred in homomorphic computation by working with separate, small plaintext moduli which are later combined (via the Chinese remainder theorem) into a larger plaintext modulus. For some applications, when floating point operations on encrypted data are necessary, our technique could prove especially useful.

This paper is organized as follows. In Section 2 we recall basic mathematical techniques used throughout the paper, the RLWE and DSPR assumptions, as well as the relevant result by Stehlé and Steinfeld. Section 3 states the public-key encryption scheme that is the foundation for the new leveled homomorphic scheme introduced in Section 4. This section also discusses correctness and security. Section 5 shows that the leveled homomorphic scheme from Section 4 can be bootstrapped to a fully homomorphic scheme. A more practical variant of the leveled scheme is introduced in Section 6 together with its security analysis, recommendations for secure parameters, and implementation performance numbers. We also present some optimizations including the CRT approach. Section 7 concludes the paper.

2 Preliminaries

In this section, we define all basic notation that is needed in the paper. The most important structure is the ring R. Let d be a positive integer and define $R = \mathbb{Z}[X]/\Phi_d(X)$ as the ring of polynomials with integer coefficients modulo the d-th cyclotomic polynomial $\Phi_d(X) \in \mathbb{Z}[X]$. The degree of Φ_d is $n = \varphi(d)$, where φ is Euler's totient function. The elements of R can be uniquely represented by all polynomials in $\mathbb{Z}[X]$ of degree less than n. Arithmetic in R is arithmetic modulo $\Phi_d(X)$, which is implicit whenever we write down terms or equalities involving elements in R. An arbitrary element $a \in R$ can be written as $a = \sum_{i=0}^{n-1} a_i X^i$ with $a_i \in \mathbb{Z}$ and we identify a with its vector of coefficients $(a_0, a_1, \ldots, a_{n-1})$. In particular, a can be viewed as an element of the $\mathbb{R}$-vector space $\mathbb{R}^n$. We choose the maximum norm on $\mathbb{R}^n$ to measure the size of elements in R. The maximum norm of a is defined as $\|a\|_\infty = \max_i\{|a_i|\}$. When multiplying two elements $g, h \in R$, the norm of their product gh expands with respect to the individual norms of g and h. The maximal norm expansion that can occur is captured in the ring constant $\delta = \sup\{\|g \cdot h\|_\infty/(\|g\|_\infty \|h\|_\infty) : g, h \in R\}$. When d is a power of 2 and thus $\Phi_d(X) = X^n + 1$, we have $\delta = n$ [9, Section 3.4]. To keep the exposition more general, we do not restrict to this special case and work with general δ in most of what follows.

Let χ be a probability distribution on R. We assume that we can efficiently sample elements from R according to χ, and we use the standard notation $a \leftarrow \chi$ to denote that $a \in R$ is sampled from χ. The distribution χ on R is called B-bounded for some $B > 0$ if for all $a \leftarrow \chi$ we have $\|a\|_\infty < B$, i.e. a is B-bounded (see [3, Def. 3] and [15, Def. 3.1 and 3.2]). Let us introduce a specific example of a distribution on R. First, define the discrete Gaussian distribution $D_{\mathbb{Z},\sigma}$ with mean 0 and standard deviation σ over the integers, which assigns a probability proportional to $\exp(-\pi|x|^2/\sigma^2)$ to each $x \in \mathbb{Z}$. When d is a power of 2 and $\Phi_d(X) = X^d + 1$, we can take χ to be the spherical discrete Gaussian $\chi = \mathcal{D}_{\mathbb{Z}^n,\sigma}$, where each coefficient of the polynomial is sampled according to the one-dimensional distribution $D_{\mathbb{Z},\sigma}$

(see [16] for more details and why $\chi = \mathcal{D}_{\mathbb{Z}^n,\sigma}$ is the right choice in that case). The distribution χ is used in many fully homomorphic encryption schemes based on RLWE to sample random error polynomials that have small coefficients with high probability. Such polynomials are a significant part of the noise terms used in the encryption process. To deduce meaningful bounds on noise size and noise growth during homomorphic operations, we assume that the distribution we are working with is B-bounded for some B. For the discrete Gaussian, this is a reasonable assumption since sampled elements tend to be small with high probability. By rejecting samples with norm larger than B, we can sample from a truncated Gaussian distribution that is statistically close to the true discrete Gaussian if B is chosen large enough. For example, if we take $B = 6\sigma$, all samples are B-bounded with very high probability [17, Lemma 4.4].

Although the principal object of interest for our scheme is the ring R, and all polynomials that we deal with are considered to be elements of R, we often reduce polynomial coefficients modulo an integer modulus q. We denote the map that reduces an integer x modulo q and uniquely represents the result by an element in the interval $(-q/2, q/2]$ by $[\cdot]_q$. We extend this map to polynomials in $\mathbb{Z}[X]$ and R by applying it to their coefficients separately, i.e. $[\cdot]_q : R \to R,\ a = \sum_{i=0}^{n-1} a_i X^i \mapsto \sum_{i=0}^{n-1} [a_i]_q X^i$. Furthermore, we extend this notation to vectors of polynomials by applying it to the entries of the vectors separately. Sometimes we use reduction modulo q and uniquely represent the result by an element in $[0, q)$. In this case, we write $r_q(x)$ to mean the reduction of x into $[0, q)$. A polynomial $f \in R$ is invertible modulo q if there exists a polynomial $f^{-1} \in R$ such that $ff^{-1} = \tilde{f}$, where $\tilde{f}(X) = \sum_i a_i X^i$ with $a_0 = 1 \mod q$ and $a_j = 0 \mod q$ for all $j \neq 0$. Our homomorphic encryption scheme uses two different moduli. In addition to a modulus q that is used to reduce the coefficients of the elements that represent ciphertexts, there is a second modulus $t < q$ that determines the message space R/tR, i.e. messages are polynomials in R modulo t. We make frequent use of the quantity $\Delta = \lfloor q/t \rfloor$ and it is readily verified that $q - r_t(q) = \Delta \cdot t$.

In [2], functions called BitDecomp and PowersOfTwo are used. We slightly generalize these to an arbitrary base and describe our notation next. Fix a positive integer $w > 1$ that is used to represent integers in a radix-w system. Let $\ell_{w,q} = \lfloor \log_w(q) \rfloor + 2$, then a non-negative integer $z < q$ can be written as $\sum_{i=0}^{\ell_{w,q}-2} z_i w^i$ where the z_i are integers such that $0 \le z_i < w$. If z is an integer in the interval $(-q/2, q/2]$, it can be written uniquely as $\sum_{i=0}^{\ell_{w,q}-1} z_i w^i$ with $z_i \in (-w/2, w/2]$. With this, an element $x \in R$ with coefficients in $(-q/2, q/2]$ can be written as $\sum_{i=0}^{\ell_{w,q}-1} x_i w^i$, where $x_i \in R$ with coefficients in $(-w/2, w/2]$. Since then $x_i = [x_i]_w$, we write $x = \sum_{i=0}^{\ell_{w,q}-1} [x_i]_w w^i$ to make clear that the norm of the coefficient polynomials x_i is at most $w/2$. With this notation, define

$$D_{q,w} : R \to R^{\ell_{w,q}}, \quad x \mapsto ([x_0]_w, [x_1]_w, \ldots, [x_{l_{q,w}-1}]_w) = ([x_i]_w)_{i=0}^{\ell_{w,q}-1},$$

this function for $w = 2$ is called BitDecomp in [2]. We define a second function

$$P_{q,w} : R \to R^{\ell_{w,q}}, \quad x \mapsto ([x]_q, [xw]_q, \ldots, [xw^{\ell_{w,q}-1}]_q) = ([xw^i]_q)_{i=0}^{\ell_{w,q}-1},$$

which is called PowersOfTwo in [2] for $w = 2$. For any two $x, y \in R$, we see that the scalar product of the vectors $D_{q,w}(x)$ and $P_{q,w}(y)$ is the same as the product xy modulo q, because

$$\langle D_{q,w}(x), P_{q,w}(y) \rangle = \sum_{i=0}^{\ell_{w,q}-1} [x_i]_w [yw^i]_q \equiv y \sum_{i=0}^{\ell_{w,q}-1} [x_i]_w w^i \equiv xy \pmod{q}.$$

Note that when $\|f\|_\infty < B$ for some $B < q$, then only the $\ell_{w,B} := \lfloor \log_w(B) \rfloor + 2$ least significant polynomials in $D_{q,w}(f)$ can be non-zero. We use the tensor product of two vectors in the usual way, i.e. for a positive integer ℓ and two vectors $a, b \in R^\ell$, the tensor $a \otimes b \in R^{\ell^2}$ is the concatenation of the $a_i b$ for $i \in \{1, 2, \ldots, \ell\}$. We extend the functions $D_{q,w}$ and $P_{q,w}$ to vectors. For $v = (v_1, v_2, \ldots, v_\ell) \in R^\ell$ denote the vector $(D_{q,w}(v_1), \ldots, D_{q,w}(v_\ell)) \in R^{\ell \cdot \ell_{w,q}}$ by $D_{q,w}(v)$, likewise we extend $P_{q,w}$.

Several operations in the scheme require scaling by rational numbers such that the resulting polynomials do not necessarily belong to R but instead have rational coefficients. In that case, a rounding procedure is applied to get back to integer coefficients. The usual rounding of a rational number a to the nearest integer is denoted by $\lfloor a \rceil$.

The Ring Learning With Errors (RLWE) Problem. Our scheme relies on the hardness of the (decisional) ring learning with errors problem, which was first introduced by Lyubashevsky, Peikert and Regev [16].

Definition 1 (Decision-RLWE). *Given a security parameter λ, let d and q be integers depending on λ, let $R = \mathbb{Z}[X]/\Phi_d(X)$ and let $R_q = R/qR$. Given a distribution χ over R_q that depends on λ, the Decision-$\mathsf{RLWE}_{d,q,\chi}$ problem is to distinguish the following two distributions. The first distribution consists of pairs (a, u), where $a, u \leftarrow R_q$ are drawn uniformly at random from R_q. The second distribution consists of pairs of the form $(a, a \cdot s + e)$. The element $s \leftarrow R_q$ is drawn uniformly at random and is fixed for all samples. For each sample, $a \leftarrow R_q$ is drawn uniformly at random, and $e \leftarrow \chi$. The Decision-$\mathsf{RLWE}_{d,q,\chi}$ assumption is that the Decision-$\mathsf{RLWE}_{d,q,\chi}$ problem is hard.*

In [16], it was shown that the hardness of RLWE can be established by a quantum reduction to worst-case shortest vector problems in ideal lattices over the ring R, see also [3, Thm. 2]. It is known that the *search* variant of $\mathsf{RLWE}_{d,q,\chi}$, in which we are required to explicitly find the secret s given an $\mathsf{RLWE}_{d,q,\chi}$ instance, is equivalent to the decision problem [16]. There are a number of variants of RLWE which are as hard as RLWE, for example we can restrict the sampling of a and e to invertible elements only [23]. And we can also choose s from χ without incurring any loss of security [1].

The Decisional Small Polynomial Ratio (DSPR) Problem. In [15], López-Alt, Tromer and Vaikuntanathan introduced the decisional small polynomial ratio problem. They describe a multikey fully homomorphic encryption scheme with security based on the assumption that the DSPR problem is hard in the ring R_q for $R = \mathbb{Z}[x]/(x^n + 1)$ for n a power of 2 and $t = 2$. We state a more general form of the problem for any cyclotomic ring $R = \mathbb{Z}[x]/(\Phi_d(x))$ and general $1 < t < q$. Let $h = tg/f \pmod{q}$ where $f = 1 + tf'$ and $f', g \leftarrow \chi$ where χ is a truncated Gaussian distribution. In [15], the problem of distinguishing such an element h from a uniformly random element of $R_q = R/qR$ was formalized as the DSPR problem. Assuming the hardness of DSPR and RLWE, the scheme in [15] is secure. To state the problem, define the following: for a distribution χ on R_q and $z \in R_q$ we define $\chi_z = \chi + z$ to be the distribution shifted by z. Also, let $R_q^\times$ be the set of all invertible elements in R_q.

Definition 2 (DSPR). *For security parameter λ, let d and q be integers, let $R = \mathbb{Z}[X]/\Phi_d(X)$ and $R_q = R/qR$ and let χ be a distribution over R_q, all depending on λ. Let $t \in R_q^\times$ be invertible in R_q, $y_i \in R_q$ and $z_i = -y_i t^{-1} \pmod{q}$ for $i \in \{1, 2\}$. The $\mathsf{DSPR}_{d,q,\chi}$ problem is to distinguish elements of the form $h = a/b$ where $a \leftarrow y_1 + t \cdot \chi_{z_1}, b \leftarrow y_2 + t \cdot \chi_{z_2}$ from uniformly random elements of R_q. The $\mathsf{DSPR}_{d,q,\chi}$ assumption is that the $\mathsf{DSPR}_{d,q,\chi}$ problem is hard.*

The following result in the full version of [23] shows that $\mathsf{DSPR}_{d,q,\chi}$ is hard when the χ_{z_i} are shifted versions of a discrete Gaussian distributions χ which is $\mathcal{D}_{\mathbb{Z}^n,\sigma}$ restricted to $R_q^\times$ for a large enough deviation σ. A discrete Gaussian on $R_q^\times$ can be obtained from a discrete Gaussian on R_q by rejecting non-invertible elements. Let $U(R_q^\times)$ be the uniform distribution on $R_q^\times$.

Theorem 1 (Stehlé and Steinfeld [24]). *Let $d \geq 8$ be a power of 2 such that $\Phi_d(X) = X^{d/2} + 1$ splits into k_q irreducible factors modulo a prime $q \geq 5$. Let $0 < \epsilon < 1/3$, $t \in R_q^\times$, χ and σ as above, $y_i \in R_q$ and $z_i = -y_i t^{-1} \pmod q$ for $i \in \{1, 2\}$. Then the statistical distance D between distributions $\frac{y_1 + t\cdot\chi_{z_1}}{y_2 + t\cdot\chi_{z_2}} \pmod q$ and $U(R_q^\times)$ is bounded by*

$$D \leq \begin{cases} 2^{10d} \cdot q^{-\frac{\lfloor \epsilon k_q \rfloor}{k_q}\cdot d} & \text{if } \sigma \geq d \cdot \sqrt{\log(8dq)} \cdot q^{\frac{1}{2}+\epsilon}, \\ 2^{10d} \cdot q^{-\epsilon d} & \text{if } \sigma \geq \sqrt{d \log(8dq)} \cdot q^{\frac{1+k_q\epsilon}{2}} \text{ and } q \geq d^{\frac{k_q}{1-2k_q\epsilon}}. \end{cases}$$

3 Basic Scheme

In this section, we describe the basic public key encryption scheme that is the foundation for the leveled schemes of the next sections. The scheme is parameterized by a modulus q and a plaintext modulus $1 < t < q$. Ciphertexts are elements of $R = \mathbb{Z}[X]/\Phi_d(X)$ and plaintexts are elements of R/tR (see Section 2). Secret keys and errors are generated from different distributions, for example Gaussian distributions of different width. The secret key is derived from the distribution χ_{key}, and errors are sampled from the distribution χ_{err}. We use "Regev-style" encryption as in [2] and [8]. The scheme consists of the following algorithms.

- $\mathsf{Basic.ParamsGen}(\lambda)$: Given the security parameter λ, fix a positive integer d that determines R, moduli q and t with $1 < t < q$, and distributions $\chi_{\text{key}}, \chi_{\text{err}}$ on R. Output $(d, q, t, \chi_{key}, \chi_{err})$.
- $\mathsf{Basic.KeyGen}(d, q, t, \chi_{key}, \chi_{err})$: Sample polynomials $f', g \leftarrow \chi_{key}$ and let $f = [tf' + 1]_q$. If f is not invertible modulo q, choose a new f'. Compute the inverse $f^{-1} \in R$ of f modulo q and set $h = [tgf^{-1}]_q$. Output the public and private key pair $(\mathsf{pk}, \mathsf{sk}) = (h, f) \in R^2$.
- $\mathsf{Basic.Encrypt}(h, m)$: The message space is R/tR. For a message $m + tR$, choose $[m]_t$ as its representative. Sample $s, e \leftarrow \chi_{\text{err}}$, and output the ciphertext

$$c = [\lfloor q/t \rfloor [m]_t + e + hs]_q \in R.$$

- $\mathsf{Basic.Decrypt}(f, c)$: To decrypt a ciphertext c, compute

$$m = \left[\left\lfloor \frac{t}{q} \cdot [fc]_q \right\rceil \right]_t \in R.$$

In the following, we often refer to a message as an element m in the ring R although the message space is R/tR, keeping in mind that encryption always takes place on the representative $[m]_t$ and that by decrypting, all that can be recovered is m modulo t.

Correctness. The following lemma provides conditions for a ciphertext c such that the decryption algorithm outputs the message m that was originally encrypted.

Lemma 1. *Let q, t, and $\Delta = \lfloor q/t \rfloor$ be as above and let $c, f, m \in R$. If there exists $v \in R$ such that*

$$fc = \Delta[m]_t + v \pmod q \text{ and } \|v\|_\infty < (\Delta - r_t(q))/2,$$

then $\mathsf{Basic.Decrypt}(f, c) = [m]_t$*, i.e.* c decrypts correctly under the secret key f.

Proof. The proof can be found in Appendix A. □

Of course, for any given c, f and m, there always exists a $v \in R$ such that $fc = \Delta[m]_t + v \pmod q$. But only a v of small norm allows one to recover $[m]_t$ from c. Since we are always free to vary v modulo q, i.e. to add any multiple of q to it, we choose v to be the canonical element $[v]_q$. This means that we choose v with the smallest possible norm among all polynomials that satisfy the equation. We call this specific v the *inherent noise in c with respect to m and f*. The previous lemma says that if the inherent noise in a ciphertext is small enough, then decryption works correctly.

Inherent noise in initial ciphertexts. The following lemma derives a bound on the inherent noise in a freshly encrypted ciphertext output by $\mathsf{Basic.Encrypt}$, assuming bounds B_{key} on the key and B_{err} on the error distributions. Note that since $f', g \leftarrow \chi_{\text{key}}$, we have $\|f'\|_\infty, \|g\|_\infty < B_{\text{key}}$ and it follows that $\|tg\|_\infty < tB_{\text{key}}$ and $\|f\|_\infty = \|1 + tf'\|_\infty < tB_{\text{key}}$ since $t \geq 2$.

Lemma 2. *Let the key and error distributions be B_{key}-bounded and B_{err}-bounded, respectively. Given $m \in R$, a public key $h = [tgf^{-1}]_q \in R$ with secret key $f = 1 + tf'$, $g, f' \leftarrow \chi_{key}$, and let $c = \mathsf{Basic.Encrypt}(h, m)$. There exists $v \in R$ such that $fc = \Delta[m]_t + v \pmod q$ and*

$$\|v\|_\infty < \delta t B_{\text{key}} \left(2B_{\text{err}} + \frac{1}{2} r_t(q) \right).$$

In particular, by Lemma 1, decryption works correctly if $2\delta t B_{\text{key}}(2B_{\text{err}} + \frac{1}{2} r_t(q)) + r_t(q) < \Delta$.

Proof. The proof is given in Appendix B. □

4 Leveled Homomorphic Scheme

In this section, we state our leveled homomorphic encryption scheme LHE based on the Basic scheme from the previous section. We then analyze the homomorphic operations and deduce bounds on the noise growth that occurs during these operations.

- $\mathsf{LHE.ParamsGen}(\lambda)$: Given the security parameter λ, output parameters $(d, q, t, \chi_{\text{key}}, \chi_{\text{err}}, w)$, where $(d, q, t, \chi_{\text{key}}, \chi_{\text{err}}) \leftarrow \mathsf{BasicParamsGen}(\lambda)$ and $w > 1$ is an integer.
- $\mathsf{LHE.KeyGen}(d, q, t, \chi_{\text{key}}, \chi_{\text{err}}, w)$: Compute $h, f \leftarrow \mathsf{Basic.KeyGen}(d, q, t, \chi_{\text{key}}, \chi_{\text{err}})$. Sample $\boldsymbol{e}, \boldsymbol{s} \leftarrow \chi_{\text{err}}^{\ell_{w,q}^3}$, compute

$$\boldsymbol{\gamma} = [f^{-1} P_{q,w}(D_{q,w}(f) \otimes D_{q,w}(f)) + \boldsymbol{e} + h \cdot \boldsymbol{s}]_q \in R^{\ell_{w,q}^3},$$

  and output $(\mathsf{pk}, \mathsf{sk}, \mathsf{evk}) = (h, f, \boldsymbol{\gamma})$.
- $\mathsf{LHE.Encrypt}(\mathsf{pk}, m)$: To encrypt $m \in R$ output $c \leftarrow \mathsf{Basic.Encrypt}(\mathsf{pk}, m) \in R$.
- $\mathsf{LHE.Decrypt}(\mathsf{sk}, c)$: Output the message $m \leftarrow \mathsf{Basic.Decrypt}(\mathsf{sk}, c) \in R$.
- $\mathsf{LHE.KeySwitch}(\tilde{c}_{\text{mult}}, \mathsf{evk})$: Output $[\langle D_{q,w}(\tilde{c}_{\text{mult}}), \mathsf{evk} \rangle]_q \in R$.
- $\mathsf{LHE.Add}(c_1, c_2)$: Compute the addition of the input ciphertexts $c_{\text{add}} = [c_1 + c_2]_q$.

- $\mathsf{LHE.Mult}(c_1, c_2, \mathsf{evk})$: Compute

$$\tilde{c}_{\text{mult}} = \left\lfloor \frac{t}{q} P_{q,w}(c_1) \otimes P_{q,w}(c_2) \right\rceil \in R^{\ell_{w,q}^2},$$

  and output

$$c_{\text{mult}} = \mathsf{LHE.KeySwitch}(\tilde{c}_{\text{mult}}, \mathsf{evk}).$$

Since encryption and decryption are the same as in the Basic scheme from Section 3, the correctness bound does not change and Lemmas 1 and 2 hold for LHE as well. Next, we analyze the homomorphic operations LHE.Add and LHE.Mult.

Homomorphic Addition. Given two ciphertexts $c_1, c_2 \in R$, which encrypt two messages m_1, m_2 with inherent noise terms v_1, v_2, their sum *modulo* q, $c_{\text{add}} = [c_1 + c_2]_q$, encrypts the sum of the messages *modulo* t, $[m_1 + m_2]_t$. Indeed, we can write $[m_1]_t + [m_2]_t = [m_1 + m_2]_t + t r_{\text{add}}$ for some $r_{\text{add}} \in R$ with $\|r_{\text{add}}\|_\infty \leq 1$. Since

$$f[c_1 + c_2]_q = fc_1 + fc_2 = \Delta([m_1]_t + [m_2]_t) + (v_1 + v_2) = \Delta([m_1 + m_2]_t + t r_{\text{add}}) + (v_1 + v_2) \pmod q,$$

we obtain $f[c_1 + c_2]_q = \Delta[m_1 + m_2]_t + (v_1 + v_2 - r_t(q) r_{\text{add}}) \pmod q$. This means that the size of the inherent noise v_{add} of c_{add} is bounded by

$$\|v_{\text{add}}\|_\infty \leq \|v_1\|_\infty + \|v_2\|_\infty + r_t(q). \tag{1}$$

Up to the term $r_t(q) < t$, the inherent noise terms are added during homomorphic addition.

Homomorphic Multiplication. The homomorphic multiplication operation is divided into two parts. The first part describes a basic procedure to obtain an intermediate ciphertext that encrypts the product $[m_1 m_2]_t$ modulo t of two messages m_1 and m_2. However, the intermediate ciphertext can not be decrypted with Basic.Decrypt using the secret key f. The second part performs a procedure which allows a public transformation of this intermediate ciphertext to a ciphertext that can be decrypted with f. This latter procedure was introduced in [5] in the form of relinearization and was later expanded in [3] into a method called key switching, that transforms a ciphertext decryptable under one secret key to one decryptable under any other secret key. For our analysis, we assume that χ_{key} and χ_{err} are B_{key}- and B_{err}-bounded, respectively. Even if we work with unbounded Gaussian distributions, this is a valid assumption since elements drawn from either distribution have bounded norm for suitable bounds with high probability. The deduction of noise bounds mostly follows the basic multiplication section of [8], since ciphertexts and the decryption algorithm in LHE have a very similar structure to those in [8].

First Step. Let $c_1, c_2 \in R$ be ciphertexts that encrypt messages $m_1, m_2 \in R$. In the first step of the homomorphic multiplication operation, we compute

$$\tilde{c}_{mult} = \left\lfloor \frac{t}{q} P_{q,w}(c_1) \otimes P_{q,w}(c_2) \right\rceil.$$

The following theorem shows that $\langle \tilde{c}_{\text{mult}}, D_{q,w}(f) \otimes D_{q,w}(f) \rangle = \Delta[m_1 m_2]_t + \tilde{v}_{\text{mult}} \pmod q$, and it provides a bound for the size of $\tilde{v}_{\text{mult}}$. Thus, $\tilde{c}_{\text{mult}}$ can be viewed as an encryption of $[m_1 m_2]_t$ under $D_{q,w}(f) \otimes D_{q,w}(f)$, if the inherent noise term $\tilde{v}_{\text{mult}}$ is small enough.

Theorem 2 (Multiplication Noise). *Let $c_1, c_2 \in R$ be ciphertexts encrypting $m_1, m_2 \in R$, decryptable with the secret key f. Let $v_1, v_2 \in R$ be the inherent noise terms in c_1, c_2 and let $V > 0$ such that $\|v_i\|_\infty \le V < \Delta/2$, $i \in \{1,2\}$. Let $\tilde{c}_{\mathrm{mult}}$ be the intermediate ciphertext in* $\mathsf{LHE.Mult}$*, and let $\ell_{w,tB_{\mathrm{key}}} = \lfloor \log_w(tB_{\mathrm{key}}) \rfloor + 2$. Then $\langle \tilde{c}_{\mathrm{mult}}, D_{q,w}(f) \otimes D_{q,w}(f) \rangle = \Delta[m_1 m_2]_t + \tilde{v}_{\mathrm{mult}} \pmod q$ where*

$$\|\tilde{v}_{\mathrm{mult}}\|_\infty < \delta t(2 + \delta \ell_{w,tB_{\mathrm{key}}} w)V + \frac{1}{2}\delta^2 \ell_{w,tB_{\mathrm{key}}} w(t^2 + \ell_{w,tB_{\mathrm{key}}} w).$$

Proof. See Appendix D. □

When starting with two ciphertexts at a given inherent noise level, the first step of the homomorphic multiplication increases the inherent noise level by a multiplicative factor of roughly $\delta^2 t \ell_{w,tB_{\mathrm{key}}} w$ and an additive term of $\frac{1}{2}\delta^2 \ell_{w,tB_{\mathrm{key}}} w(t^2 + \ell_{w,tB_{\mathrm{key}}} w)$.

Key Switching. The second part in the homomorphic multiplication procedure is a key switching step, which transforms the ciphertext $\tilde{c}_{\mathrm{mult}}$ into a ciphertext c_{mult} that is decryptable under the original secret key f. We define an evaluation key

$$\mathsf{evk} = [f^{-1} P_{q,w}(D_{q,w}(f) \otimes D_{q,w}(f)) + \boldsymbol{e} + h \cdot \boldsymbol{s}]_q,$$

where $\boldsymbol{e}, \boldsymbol{s} \leftarrow \chi_{\mathrm{err}}^{\ell_{w,q}^3}$ are vectors of polynomials sampled from the error distribution χ_{err} and $[\cdot]_q$ is applied to each coefficient of the vector. Note that this key is a vector of quasi-encryptions of $f^{-1} P_{q,w}(D_{q,w}(f) \otimes D_{q,w}(f))$, which depends on the secret key f, under its corresponding public key and that it is made public because it is needed for the homomorphic multiplication operation. Therefore, we need to make a circular security assumption, namely that the scheme is still secure even given that evk is publicly known (see Section 4.1). The following lemma deduces a bound on the noise caused by the key switching procedure and states an overall bound on the noise growth during a single homomorphic multiplication operation.

Lemma 3. *Let notation be as in Theorem 2 and as introduced above. In particular, let $\tilde{c}_{\mathrm{mult}}$ be the intermediate ciphertext in* $\mathsf{LHE.Mult}$*. Its inherent noise term is denoted by $\tilde{v}_{\mathrm{mult}}$. Let* evk *be the evaluation key and let $c_{\mathrm{mult}} = \mathsf{LHE.KeySwitch}(\tilde{c}_{\mathrm{mult}}, \mathsf{evk})$. Then $f c_{\mathrm{mult}} = \Delta[m_1 m_2]_t + v_{\mathrm{mult}} \pmod q$, where*

$$\|v_{\mathrm{mult}}\|_\infty < \|\tilde{v}_{\mathrm{mult}}\|_\infty + \delta^2 t \ell_{w,q}^3 w B_{\mathrm{err}} B_{\mathrm{key}}.$$

Proof. See Appendix E. □

Theorem 2 and Lemma 3 give an overall upper bound on the noise growth during a homomorphic multiplication. This clearly dominates the noise growth for homomorphic addition.[3]

4.1 Correctness & Security of LHE

This section discusses correctness and security of LHE. We state correctness by giving an asymptotic bound on the number of multiplicative levels in an arithmetic circuit that can be correctly evaluated. For this, we concretely focus on a parameter setting such that the assumptions of Theorem 1 hold. This means that the DSPR problem is hard in R_q. We

[3] As noted in [2] the number of elements in $D_{q,w}(f) \otimes D_{q,w}(f)$ can be reduced from $\ell_{w,q}^2$ to $\binom{\ell_{w,q}}{2}$ which correspondingly reduces the number of ring elements in evk.

therefore fix the following parameters: let d be a power of 2, $n = \varphi(d)$, $\epsilon \in (0, 1)$, $k \in (1/2, 1)$ and let $q = 2^{d^\epsilon}$ be a prime such that $\Phi_d(X) = X^n + 1$ splits into n irreducible factors modulo q. Let χ_{key} be a discrete Gaussian distribution on R_q with deviation $\sigma_{\text{key}} \geq d\sqrt{\log(8dq)} \cdot q^k$, and let χ_{err} be a $\omega(\sqrt{d\log(d)})$-bounded Gaussian distribution on R. Finally, we fix $w = 2$ and $t = 2$, but note that similar results hold for general w, t – this restriction is merely for the purpose of exposition.

Theorem 3 (Correctness of LHE). *For the parameter choices above,* LHE *can evaluate any circuit of depth*

$$L = \mathcal{O}\left(\frac{(1-k)\log(q)}{\log(\log(q)) + \log(d)}\right)$$

Proof. See Appendix F. □

To prove security of LHE, we need to assume that IND-CPA security can be maintained even when an adversary has access to elements of the evaluation key evk. Due to the way we construct evk it is not sufficient to simply replace f by L distinct secret keys f_i, as has been done in previous works – a specific assumption is still required. This is a form of key dependent message security, for the family of functions defining the evaluation key. Under this "circular security" assumption, the IND-CPA security of LHE follows from the IND-CPA security of the scheme Basic described in Section 3 and the RLWE assumption.

Theorem 4 (Security of LHE). *The scheme* LHE *is IND-CPA secure under the* $\mathsf{RLWE}_{d,q,\chi_{\text{err}}}$ *assumption and the assumption that the scheme remains IND-CPA secure, even when an adversary has access to* evk *output by* $\mathsf{LHE.KeyGen}(d, q, 2, \chi_{\text{key}}, \chi_{\text{err}}, 2)$.

Proof. Since $\sigma_{\text{key}} \geq d\sqrt{\log(8dq)} \cdot q^k$ for some $k > 1/2 + \nu$ with $\nu > 0$, the conditions of Theorem 1 are satisfied. Hence the public key is indistinguishable from a uniform element of $R_q^\times$. It follows from Lemma 13 in [23] that the scheme Basic is IND-CPA secure under the $\mathsf{RLWE}_{d,q,\chi_{\text{err}}}$ assumption in R_q. Under the circular security assumption outlined above, the IND-CPA security of LHE follows. □

For the proof of Theorem 4, we only need parameters that satisfy the assumptions in Theorem 1. For the parameters outlined at the beginning of this subsection, the RLWE assumption is believed to be hard based on standard worst-case lattice problems.

5 From Leveled to Fully Homomorphic Encryption

In [9], Gentry showed how a fully homomorphic scheme can be obtained from a leveled homomorphic scheme supporting computation of circuits of sufficient depth. If a scheme can evaluate its own decryption circuit and one additional multiplication, then that scheme can be converted to a fully homomorphic scheme. The only caveat is that we have to make an additional assumption: to execute the bootstrapping procedure, it is necessary to augment the public key with encryptions $\mathsf{LHE.Encrypt}(\mathsf{pk}, \mathsf{sk}[j])$ of the bits of the secret key, under its corresponding public key. Similarly to the assumption on the evaluation key, we need to make an additional assumption that including encryptions of bits of the secret key does not affect security. The following lemma estimates the depth of the decryption circuit for LHE.

Lemma 4. *The decryption circuit for* LHE *can be implemented as a polynomial size circuit of depth* $\mathcal{O}(\log(\log(q)) + \log(d))$ *over* $\mathbb{F}_2$.

Proof. The first stage of decryption in LHE consists of the multiplication of two elements of R_q. In [4, Lemma 4.5], it was shown that this can be computed using a circuit of depth $\mathcal{O}(\log(\log(q)) + \log(d))$ over $\mathbb{F}_2$ (see also [15, Lemma 4.4]). Note that the scaling and rounding operation can be done at a cost of less than the above multiplication with integer multiplications and simple bit shift operations following techniques in [8]. Finally, the reduction modulo 2 does not increase the depth since this simply corresponds to outputting the least significant bit. □

To achieve a fully homomorphic scheme, we simply view the decryption circuit as a circuit computed on the bits of the secret key at a ciphertext c we wish to refresh. The noise in the resulting *fresh* ciphertext will be of fixed size depending on the noise in the encryptions of the bits of the secret key.

Theorem 5 (Fully Homomorphic Encryption). *Under the same assumptions as in Theorem 4 and the additional assumption that* LHE *remains IND-CPA secure even when an adversary is given encryptions* $\mathsf{LHE.Encrypt}(\mathsf{pk}, \mathsf{sk}[j])$ *of the bits of the secret key output by* $\mathsf{LHE.KeyGen}(d, q, 2, \chi_{\text{key}}, \chi_{\text{err}}, 2)$*, and for the same parameter choices as in Section 4.1,* LHE *can be made into a IND-CPA fully homomorphic encryption scheme.*

Proof. From Theorem 3 we know that LHE can correctly compute any circuit of depth

$$\mathcal{O}\left(\frac{(1-k)\log(q)}{\log(\log(q)) + \log(d)}\right) = \mathcal{O}\left(\frac{(1-k)d^{\epsilon}}{\log(d)}\right)$$

for our parameter choices. Since this is greater than the depth of the decryption circuit (for k, ϵ close to 1/2, say) it follows from Gentry's Bootstrapping Theorem [9] that LHE can be converted into a fully homomorphic scheme. □

6 A More Practical Variant of the Scheme

In this section, we propose a more practical variant LHE′ of LHE. The difference to LHE lies in the homomorphic multiplication procedure. In LHE′, an intermediate ciphertext is simply a single polynomial while it is a vector of polynomials in LHE. This results in an evaluation key that consists of only $\ell_{w,q}$ polynomials instead of $\ell_{w,q}^3$ for LHE and thus in a simpler key switching procedure. We now state the scheme and discuss the noise growth during the simplified homomorphic multiplication operation LHE′.Mult.

- $\mathsf{LHE'.ParamsGen}(\lambda)$: Output $(d, q, t, \chi_{key}, \chi_{err}) \leftarrow \mathsf{BasicParamsGen}(\lambda)$.
- $\mathsf{LHE'.KeyGen}(d, q, t, \chi_{key}, \chi_{err}, w)$: Compute $h, f \leftarrow \mathsf{Basic.KeyGen}(d, q, t, \chi_{\text{key}}, \chi_{\text{err}})$. Sample $\boldsymbol{e}, \boldsymbol{s} \leftarrow \chi_{err}^{\ell_{w,q}}$, compute $\boldsymbol{\gamma} = [P_{q,w}(f) + \boldsymbol{e} + h \cdot \boldsymbol{s}]_q \in R^{\ell_{w,q}}$. and output $(\mathsf{pk}, \mathsf{sk}, \mathsf{evk}) = (h, f, \boldsymbol{\gamma})$.
- $\mathsf{LHE'.Encrypt}(\mathsf{pk}, m)$: To encrypt $m \in R$ output $c \leftarrow \mathsf{Basic.Encrypt}(\mathsf{pk}, m) \in R$.
- $\mathsf{LHE'.Decrypt}(\mathsf{sk}, c)$: Output the message $m \leftarrow \mathsf{Basic.Decrypt}(\mathsf{sk}, c) \in R$.
- $\mathsf{LHE'.KeySwitch}(\tilde{c}_{\text{mult}}, \mathsf{evk})$: Output the ciphertext $[\langle D_{q,w}(\tilde{c}_{\text{mult}}), \mathsf{evk}\rangle]_q$.
- $\mathsf{LHE'.Add}(c_1, c_2)$: Output the ciphertext $c_{add} \leftarrow \mathsf{LHE.Add}(c_1, c_2) = [c_1 + c_2]_q$.
- $\mathsf{LHE'.Mult}(c_1, c_2, \mathsf{evk})$: Output the ciphertext

$$c_{\text{mult}} = \mathsf{LHE'.KeySwitch}(\tilde{c}_{\text{mult}}, \mathsf{evk}), \text{ where } \tilde{c}_{\text{mult}} = \left[\left\lfloor \frac{t}{q} c_1 c_2 \right\rceil\right]_q .$$

For two ciphertexts $c_1, c_2 \in R$ that encrypt $m_1, m_2 \in R$, the intermediate ciphertext $\tilde{c}_{\text{mult}}$ during homomorphic multiplication $\mathsf{LHE}'.\mathsf{Mult}$ satisfies $f^2\tilde{c}_{\text{mult}} = \Delta[m_1m_2]_t + \tilde{v}_{\text{mult}} \pmod q$ as shown in the following theorem. This means that $\tilde{c}_{\text{mult}}$ is an encryption of $[m_1m_2]_t$ under f^2. The theorem also provides an upper bound on the inherent noise term in the intermediate ciphertext. We assume that the error distribution χ_{err} is B_{err}-bounded and that the key distribution χ_{key} is B_{key}-bounded.

Theorem 6 (Multiplication Noise). *Let $c_1, c_2 \in R$ be ciphertexts encrypting $m_1, m_2 \in R$, which are decryptable with the secret key f. Let $v_1, v_2 \in R$ be the inherent noise terms in c_1, c_2 and let $V > 0$ such that $\|v_i\|_\infty \le V < \Delta/2$, $i \in \{1, 2\}$. Let $\tilde{c}_{\text{mult}}$ be the intermediate ciphertext in $\mathsf{LHE}'.\mathsf{Mult}$. Then $f^2c_{\text{mult}} = \Delta[m_1m_2]_t + \tilde{v}_{\text{mult}} \pmod q$ where $\|\tilde{v}_{\text{mult}}\|_\infty < \delta t(3 + \delta t B_{\text{key}})V + \frac{1}{2}\delta^2t^2B_{\text{key}}(B_{\text{key}} + t)$.*

Proof. See Appendix G. □

Key Switching. The key switching algorithm now transforms such an intermediate encryption into a ciphertext that can be decrypted with f itself. The evaluation key is $\mathsf{evk} = [P_{q,w}(f) + \boldsymbol{e} + h \cdot \boldsymbol{s}]_q$, where $\boldsymbol{e}, \boldsymbol{s} \leftarrow \chi_{\text{err}}^{\ell_{w,q}}$ are vectors of polynomials sampled from the error distribution χ_{err}. Again, this key is a vector of quasi-encryptions of the secret key f under its corresponding public key. It is required for the homomorphic multiplication operation and is therefore made public. This means, we need to make a circular security assumption as for LHE, namely that the scheme is still secure even given that evk is publicly known. The following lemma gives a bound on the key switching noise.

Lemma 5. *Let $\tilde{c}_{\text{mult}}$ be the intermediate ciphertext in $\mathsf{LHE}'.\mathsf{Mult}$. Its inherent noise term is denoted by $\tilde{v}_{\text{mult}}$. Let $\boldsymbol{\gamma}$ be the evaluation key from above and $c_{\text{mult}} = \mathsf{LHE}'.\mathsf{KeySwitch}(\tilde{c}_{\text{mult}}, \boldsymbol{\gamma})$. Then $fc_{\text{mult}} = \Delta[m_1m_2]_t + v_{\text{mult}} \pmod q$, where*

$$\|v_{\text{mult}}\|_\infty < \|\tilde{v}_{\text{mult}}\|_\infty + \delta^2t\ell_{w,q}wB_{\text{err}}B_{\text{key}}.$$

Proof. The proof is given in Appendix H. □

6.1 Correctness & Security of LHE'

In the following theorem, we give an explicit bound for correctness of a homomorphic evaluation of an arithmetic circuit in R/tR of multiplicative depth L that is organized in a leveled tree structure of multiplications without any additions. At each level all ciphertexts are assumed to have inherent noise terms of roughly the same size. The bounds that we obtain might be too large and could be significantly reduced for computations that involve more additions and less multiplications as well as multiplications of ciphertexts with imbalanced inherent noise terms. In favor of simplicity, we restrict to the above setting.

Theorem 7 (Correctness of LHE'). *The scheme LHE' can correctly evaluate an arithmetic circuit consisting of L-levels of multiplications in R/tR on ciphertexts with inherent noise of size at most V that are arranged in a binary tree of L levels of multiplications if*

$$2(1+\epsilon_1)^{L-1}\delta^{2L}t^{2L-1}B_{\text{key}}^L\left((1+\epsilon_1)tV + L\cdot\left(\frac{1}{2}t(B_{\text{key}}+t) + \ell_{w,q}wB_{\text{err}}\right)\right) + t - 1 < \Delta.$$

Proof. See Appendix J. □

Appendix I gives detailed bounds on the increase of the inherent noise terms in ciphertexts during homomorphic addition and multiplication. One can take these bounds to deduce overall bounds for the exact computation that is supposed to be carried out on encrypted data. The obtained bounds can then be used to deduce tailored parameters for the scheme to ensure correctness and security for that particular setting, possibly resulting in more efficient parameters for the specific computation.

The security of LHE' is based on the RLWE assumption and a circular security assumption similar to the one for LHE. The price we pay for a simpler homomorphic multiplication operation lies in an additional security assumption. Since LHE' only works for a much narrower key distribution that does not satisfy the requirements for applying the Stehlé and Steinfeld result [23, Thm. 4.1], security also relies on the Decisional Small Polynomial Ratio (DSPR) assumption, as stated in Section 2. In LHE, this assumption could be avoided by making the scheme work with a key distribution as demanded by [23]. Following the same hybrid argument as in [15], one can prove that the scheme described in this section is secure under the DSPR assumption and the RLWE assumption (see [15, Section 3.3]). If a, b are two elements sampled from a Gaussian with very small standard deviation or from a different distribution that yields polynomials with very small coefficients only, the ratio $h = a/b$ can clearly not be uniform because the number of elements for a and b is too small and produces only a small number of values for h when compared to all elements in R_q. Still, a computationally bounded adversary might not be able to distinguish such a case from uniform randomly chosen h.

Theorem 8 (Security of LHE'). *Let d be a positive integer, q and $t < q$ be two moduli, w be a fixed positive integer, and let χ_{key} and χ_{err} be distributions on R. The scheme LHE' is IND-CPA secure under the $\mathsf{RLWE}_{d,q,\chi_{\text{err}}}$ assumption, the $\mathsf{DSPR}_{d,q,\chi_{\text{key}}}$ assumption, and the assumption that the scheme remains IND-CPA secure even when the evaluation key evk output by $\mathsf{LHE}'.\mathsf{KeyGen}(d, q, t, \chi_{\text{key}}, \chi_{\text{err}})$ is known to the adversary.*

6.2 Parameters

In this section, we give suggestions for choosing concrete parameters which can be used as a guideline to instantiate practical schemes with varying complexity. There are multiple parameters one can adjust, so we restrict ourselves to a subset of choices which we think are most relevant. We consider two settings. In the first, we fix a specific size for the modulus q. This is interesting for instance when a fast modular multiplication implementation (in either hard- or software) is already available, and one prefers to use this to boost the scheme's performance. We fix different sizes for the modulus q starting from 64 bits up to 1024 bits. The other setting focuses on special-purpose polynomial arithmetic. Here, we fix the degree $n = \varphi(d)$ to be a power of 2 between 2^{11} and 2^{16}.

The parameters presented in Table 1 are obtained by following the security analysis of Lindner and Peikert [14] under the assumption that the results from [14] in the LWE setting carry over to the RLWE setting, and assuming that the assumptions in Section 6.1 hold. This analysis is similar to the ones from [11,8,13] and we refer to [11] for a more complete discussion of assumptions made in deriving parameters. A more detailed discussion of Table 1 can be found in Appendix K.

Table 1. Parameters that guarantee security of $\lambda = 80$ bits against the distinguishing attack with advantage $\epsilon = 2^{-80}$. We fix $w = 2^{32}$, the key distribution is assumed to be bounded by $B_{\text{key}} = 1$, and we use $\sigma_{\text{err}} = 8$ and $B_{\text{err}} = 6\sigma_{\text{err}}$. Either for fixed sizes of q, we give the minimal degree $n_{\min}$ (left part), or for fixed dimension n, we give the maximal size $\log(q_{\max})$ (right part). For each pair (q, n) according to the given sizes, and different values of t, correctness is guaranteed for at most $L_{\max}$ multiplicative levels.

$\lceil\log(q)\rceil$	$n_{\min}$	t	$L_{\max}$
		2	1
64	1641	256	0
		1024	0
		2	3
128	3329	256	2
		1024	1
		2	5
192	5018	256	3
		1024	3
		2	7
256	6707	256	5
		1024	4
		2	15
512	13463	256	10
		1024	9
		2	31
1024	26974	256	21
		1024	19

n	$\log(q_{\max})$	t	$L_{\max}$
		2	2
2^{11}	79	256	1
		1024	0
		2	5
2^{12}	157	256	3
		1024	2
		2	10
2^{13}	312	256	6
		1024	5
		2	19
2^{14}	622	256	13
		1024	12
		2	38
2^{15}	1243	256	26
		1024	23
		2	72
2^{16}	2485	256	50
		1024	46

6.3 Implementation

Currently there are not many known implementation results for FHE schemes. Some of those which have been published demonstrate that the current state-of-the-art is *totally impractical* as illustrated by the implementations which are capable of computing AES homomorphically [11,7]. Other people have focused on implementing relatively simple schemes that require only a few levels of multiplications [13]. When using the ring $R = \mathbb{Z}[X]/(X^{4096} + 1)$, $t = 2^{10}$ and a 130-bit prime q, the authors of [13] present implementation results on an Intel Core 2 Duo running at 2.1 GHz. Encryption takes 756 ms, addition of ciphertexts 4 ms, multiplication of ciphertexts 1590 ms (this includes the degree reduction) and decryption 57 ms.

We have implemented the LHE' variant proposed in Section 6 in a C-library. All the arithmetic has been built from scratch and we do not depend on any external number theory library. Using almost the same parameters (we use a 127-bit prime q) with $w = 2^{32}$ we obtained the following results on an Intel Core i7-3520M at 2893.484 MHz with hyperthreading turned off and over-clocking ("turbo boost") disabled. Encryption runs in 79.2 million cycles (27 ms), addition of ciphertexts in 70 thousand cycles (0.024 ms), multiplication of ciphertexts (including the key-switching) in 90.7 million cycles (31 ms) and decryption in 14.1 million cycles (5 ms).

This performance increase by at least one order of magnitude (for the decryption) to two orders of magnitude (for the addition of ciphertexts) can be partially explained by the fact that we are running on a more recent processor and that we implemented the scheme directly in C (avoiding the overhead incurred by using a computer algebra system as in [13]). The remainder of the speed-up is due to our newly proposed scheme, in particular due to a simpler multiplication operation on ciphertexts that uses a more compact evaluation key consisting of only 4 elements. These performance numbers highlight the fact that FHE is practical for schemes which do not require very deep circuits (like AES) but instead only need a few (around 2^2 to 2^5) multiplications.

6.4 Truncating Ciphertext Words

Brakerski [2, Section 4.2] first suggested for his scale-invariant LWE scheme to discard some least significant bits of the ciphertext. Based on this idea, we describe an optimization to our scheme which significantly reduces both the ciphertext length and the number of elements in the evaluation key. By aligning the number of bits we discard with a multiple of w used in $\mathsf{LHE.KeySwitch}$, the number of elements required to switch keys is reduced per multiplication.

Define $\mathsf{LHE.Discard_w}(c, i)$, which takes as input a ciphertext and the number $0 \leq i < \ell_{w,q}$ of w-words to be truncated and outputs $c' = \mathsf{LHE.Discard_w}(c, i) = \lfloor w^{-i} c \rfloor$. Then, $w^i c'$ is equal to c with the i least significant w-words of c being set to 0. If $cf = \Delta m + v \pmod q$, then $w^i \cdot (c' f) = \Delta m + v' \pmod q$ with $\|v'\|_\infty \leq \|v\|_\infty + \frac{1}{2} \delta w^i \|f\|_\infty$. For a constant $B > 0$ such that $2B > \delta \|f\|_\infty / 2$, if we discard $\log_w(2B) - \log_w(\delta \|f\|_\infty)$ words, we incur an additional noise term of size B, but the ciphertext can now be represented by $\log_w(q/B) + \log_w(\delta \|f\|_\infty / 2)$ words. This means that, with discarding, the length of ciphertexts does not depend on the absolute value of q but only on the ratio of q to the noise in the ciphertext. Perhaps more importantly, this means that when we consider $D_{q,w}(c)$ for a ciphertext c with coefficients represented by roughly $\log_w(q/B)$ words, all the lowest $\log_w(B)$ words are now zero. If c is a ciphertext decryptable under f^2, in the key switching step, we only need the top $\log_w(q/B)$ elements from the evaluation key to carry out the switch.

6.5 Encoding input data via the CRT method

For our leveled homomorphic encryption scheme, we have carefully given bounds on parameters and input data to ensure correctness and security. For applications such as outsourcing of storage and computation on private data to the cloud, it could be the case that the user requires a flexible system which allows for additional computation, more computation than was planned for when setting parameters for the system. Here we propose a way to extend the system to allow additional computation without resetting the parameters. By allowing encoding of larger integers using the Chinese Remainder approach, we can allow for either greater precision of computation or larger integer inputs, using the same underlying field size and lattice dimension but at the cost of increasing the number of ciphertexts to be operated on. Integers up to a bound B can be encoded as collections of integers, each up to a bound t_i, and operations can be done correctly on the collection, as long as the t_i are coprime and the product of the t_i is greater than B. Each integer modulo t_i in the collection can then be encrypted and those ciphertexts can be processed in parallel to return encrypted collections which are then decrypted and the CRT theorem is used to recover the output as an actual integer.

7 Conclusions

We have proposed a new fully homomorphic encryption scheme based on the scheme by Stehlé and Steinfeld which removes the non-standard decisional small polynomial ratio assumption needed in the homomorphic encryption scheme by López-Alt, Tromer and Vaikuntanathan.

Hence, the security is solely based on standard lattice assumptions and a circular security assumption. Our new scheme avoids modulus switching, eliminates ciphertext expansion in homomorphic multiplication and reduces the size of ciphertexts to a single ring element. Furthermore, we have presented a more practical variant of our scheme which does need the decisional small polynomial ratio assumption. For this latter scheme we presented parameters and implementation results.

Acknowledgments. The authors thank Nigel P. Smart for helpful suggestions and the anonymous reviewers for their constructive feedback.

References

1. B. Applebaum, D. Cash, C. Peikert, and A. Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference*, volume 5677 of *Lecture Notes in Computer Science*, pages 595–618. Springer, 2009.
2. Z. Brakerski. Fully homomorphic encryption without modulus switching from classical GapSVP. In *Advances in Cryptology - Crypto 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 868–886. Springer, 2012.
3. Z. Brakerski, C. Gentry, and V. Vaikuntanathan. Fully homomorphic encryption without bootstrapping. In *ITCS*, pages 309–325, 2012.
4. Z. Brakerski and V. Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In *FOCS*, pages 97–106, 2011.
5. Z. Brakerski and V. Vaikuntanathan. Fully homomorphic encryption from ring-LWE and security for key dependent messages. In *CRYPTO*, pages 505–524, 2011.
6. Y. Chen and P. Q. Nguyen. BKZ 2.0: Better lattice security estimates. In *ASIACRYPT*, volume 7073 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2011.
7. J.-S. Coron, T. Lepoint, and M. Tibouchi. Batch fully homomorphic encryption over the integers. Cryptology ePrint Archive, Report 2013/036, 2013. `http://eprint.iacr.org/`.
8. J. Fan and F. Vercauteren. Somewhat practical fully homomorphic encryption. Cryptology ePrint Archive, Report 2012/144, 2012. `http://eprint.iacr.org/`.
9. C. Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, pages 169–178, 2009.
10. C. Gentry and S. Halevi. Implementing gentry's fully-homomorphic encryption scheme. In *Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, volume 6632 of *Lecture Notes in Computer Science*, page 129. Springer, 2011.
11. C. Gentry, S. Halevi, and N. P. Smart. Homomorphic evaluation of the AES circuit. In *Advances in Cryptology - Crypto 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 850–867. Springer, 2012.
12. J. Hoffstein, J. Pipher, and J. H. Silverman. NTRU: A ring-based public key cryptosystem. In J. Buhler, editor, *ANTS*, volume 1423 of *Lecture Notes in Computer Science*, pages 267–288. Springer, 1998.
13. K. Lauter, M. Naehrig, and V. Vaikuntanathan. Can homomorphic encryption be practical? In C. Cachin and T. Ristenpart, editors, *CCSW*, pages 113–124. ACM, 2011.
14. R. Lindner and C. Peikert. Better key sizes (and attacks) for LWE-based encryption. In *Proceedings of the 11th international conference on Topics in cryptology: CT-RSA 2011*, CT-RSA'11, pages 319–339, Berlin, Heidelberg, 2011. Springer-Verlag.
15. A. López-Alt, E. Tromer, and V. Vaikuntanathan. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In *STOC*, pages 1219–1234, 2012.
16. V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. In *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, volume 6110 of *Lecture Notes in Computer Science*, pages 1–23. Springer, 2010.
17. D. Micciancio and O. Regev. Worst-case to average-case reductions based on Gaussian measures. In *SIAM J. on Computing*, pages 372–381. IEEE Computer Society, 2004.
18. D. Micciancio and O. Regev. Lattice-based cryptography. In D. J. Bernstein, J. Buchmann, and E. Dahmen, editors, *Post-Quantum Cryptography*, pages 147–191. Springer Berlin Heidelberg, 2009.

19. D. Naccache, J.-S. Coron, A. Mandal, and M. Tibouchi. Fully homomorphic encryption over the integers with shorter public keys. In *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference*, volume 6841 of *Lecture Notes in Computer Science*, page 483. Springer, 2011.
20. R. L. Rivest, L. Adleman, and M. L. Dertouzos. On data banks and privacy homomorphisms. In *Foundations of secure computation*, volume 4, pages 169–180. New-York: Academic Press, 1978.
21. C.-P. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theor. Comput. Sci.*, 53:201–224, 1987.
22. N. P. Smart and F. Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. In *Public Key Cryptography - PKC 2010, 13th International Conference on Practice and Theory in Public Key Cryptography, Paris, France, May 26-28, 2010. Proceedings*, volume 6056 of *Lecture Notes in Computer Science*, pages 420–443. Springer, 2010.
23. D. Stehle and R. Steinfeld. Making NTRU as secure as worst-case problems over ideal lattices. In *Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, volume 6632 of *Lecture Notes in Computer Science*, page 27. Springer, 2011.
24. D. Stehle and R. Steinfeld. Making NTRU as secure as worst-case problems over ideal lattices. Cryptology ePrint Archive, Report 2013/004, 2012.
25. M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. Fully homomorphic encryption over the integers. In *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, volume 6110 of *Lecture Notes in Computer Science*, pages 24–43. Springer, 2010.

A Proof of Lemma 1

Since $fc = \Delta[m]_t + v \pmod q$, we can write $[fc]_q = \Delta[m]_t + v + qa$ for some $a \in R$ and compute

$$t/q \cdot [fc]_q = (t/q)\Delta[m]_t + v \cdot t/q + ta.$$

Using $q = \Delta t + r_t(q)$ and setting $\epsilon = r_t(q)/t$, this yields $t/q \cdot [fc]_q = [m]_t + t/q(v - (q/t - \Delta)[m]_t) + ta = [m]_t + t/q(v - \epsilon[m]_t) + ta$. The assumption on the norm of v implies $\|v - \epsilon[m]_t\|_\infty < (\Delta - r_t(q))/2 + r_t(q)/2 = \Delta/2 \leq q/(2t)$, i.e. $\|t/q(v - \epsilon[m]_t)\|_\infty < 1/2$. Therefore, rounding of $[fc]_q$ gives $[m]_t + ta$ and thus c decrypts correctly. □

B Proof of Lemma 2

We have $c = \Delta[m]_t + e + hs$ for $e, s \leftarrow \chi_{\text{err}}$. Using $\Delta t = -r_t(q) \pmod q$, it follows that

$$fc = \Delta f[m]_t + fe + tgs = \Delta[m]_t - r_t(q)f'[m]_t + fe + tgs \pmod q.$$

Define $v = fe + tgs - r_t(q)f'[m]_t$. Since $f', g \leftarrow \chi_{\text{key}}$ and $s, e \leftarrow \chi_{\text{err}}$, we obtain $\|fe\|_\infty, \|tgs\|_\infty < \delta t B_{\text{key}} B_{\text{err}}$ and $\|f'[m]_t\|_\infty < \delta B_{\text{key}} t/2$. Altogether, $\|v\|_\infty < 2\delta t B_{\text{key}} B_{\text{err}} + r_t(q)\delta t B_{\text{key}}/2$. □

C A Useful Tool

Lemma 6. *Let $c_1, c_2 \in R$ be ciphertexts that encrypt messages $m_1, m_2 \in R$, that are decryptable with the secret key f, and that have inherent noise terms $v_1, v_2 \in R$ where $\|v_i\|_\infty < \Delta/2$. Let $fc_i = \Delta[m_i]_t + v_i + qr_i$ for polynomials $r_i \in R$ as above. Let $[m_1]_t[m_2]_t = [m_1 m_2]_t + tr_m$ and $v_1 v_2 = [v_1 v_2]_\Delta + \Delta r_v$ where $r_m, r_v \in R$. Then $\|r_m\|_\infty < \frac{1}{2}\delta t$, $\|r_v\|_\infty \leq \frac{1}{2}\delta \min_i \|v_i\|_\infty$ and*

$$\begin{aligned}\frac{t}{q} f^2 c_1 c_2 =& \Delta[m_1 m_2]_t + [m_1]_t v_2 + [m_2]_t v_1 + t(v_1 r_2 + r_1 v_2) - \\ & r_t(q)([m_1]_t r_2 + [m_2]_t r_1 + r_m) + r_v + r_r + qs\end{aligned}$$

where $s \in R$ *and* $r_r = \frac{t}{q}[v_1 v_2]_\Delta - \frac{r_t(q)}{q}(\Delta[m_1]_t[m_2]_t + [m_1]_t v_2 + [m_2]_t v_1 + r_v) \in \frac{1}{q}R$. *Note that all terms on the right hand side of the equation are elements of* R, *except possibly* r_r, *which is bounded by* $\|r_r\|_\infty < \frac{1}{2}(1 + r_t(q)\delta(1 + \frac{t}{2} + \min_i \|v_i\|_\infty))$.

Proof. The bounds on r_m and r_v can be derived as follows. First, we have $\|r_m\|_\infty = \frac{1}{t}\|[m_1]_t[m_2]_t - [m_1 m_2]_t\|_\infty \leq \frac{1}{t}(\delta\frac{t^2}{4} + \frac{t}{2}) = \delta\frac{t}{4} + \frac{1}{2} < \frac{1}{2}\delta t$. Similarly, $\|r_v\|_\infty \leq \frac{1}{\Delta}\delta\|v_1\|_\infty\|v_2\|_\infty + \frac{1}{2} < \frac{1}{2}\delta \min_i \|v_i\|_\infty + \frac{1}{2}$, i.e. $2\|r_v\|_\infty < \delta \min_i \|v_i\|_\infty + 1$. Since $r_v \in R$, we get $2\|r_v\|_\infty \leq \delta \min_i \|v_i\|_\infty$. Multiplying out and making use of the equality $q - r_t(q) = \Delta t$, we obtain

$$\begin{aligned}
\frac{t}{q} \cdot f^2 c_1 c_2 &= \frac{\Delta t}{q}\Delta[m_1]_t[m_2]_t + \frac{\Delta t}{q}([m_1]_t v_2 + [m_2]_t v_1) + t(v_1 r_2 + r_1 v_2) + \frac{t}{q} v_1 v_2 \\
&\quad + \Delta t([m_1]_t r_2 + r_1[m_2]_t) + qt r_1 r_2 \\
&= \Delta[m_1]_t[m_2]_t - \frac{r_t(q)}{q}\Delta[m_1]_t[m_2]_t + [m_1]_t v_2 + [m_2]_t v_1 - \frac{r_t(q)}{q}([m_1]_t v_2 + [m_2]_t v_1) \\
&\quad + t(v_1 r_2 + v_2 r_1) + \frac{t}{q} v_1 v_2 + q([m_1]_t r_2 + r_1[m_2]_t) - r_t(q)([m_1]_t r_2 + [m_2]_t r_1) + qt r_1 r_2 \\
&= \Delta[m_1]_t[m_2]_t + [m_1]_t v_2 + [m_2]_t v_1 + t(v_1 r_2 + r_1 v_2) - r_t(q)([m_1]_t r_2 + [m_2]_t r_1) \\
&\quad - \frac{r_t(q)}{q}(\Delta[m_1]_t[m_2]_t + [m_1]_t v_2 + [m_2]_t v_1) + \frac{t}{q} v_1 v_2 + q(t r_1 r_2 + [m_1]_t r_2 + [m_2]_t r_1) \\
&= \Delta[m_1 m_2]_t + [m_1]_t v_2 + [m_2]_t v_1 + t(v_1 r_2 + r_1 v_2) - r_t(q)([m_1]_t r_2 + [m_2]_t r_1 + r_m) + r_v \\
&\quad - \frac{r_t(q)}{q}(\Delta[m_1]_t[m_2]_t + [m_1]_t v_2 + [m_2]_t v_1 + r_v) + \frac{t}{q}[v_1 v_2]_\Delta \\
&\quad + q(t r_1 r_2 + [m_1]_t r_2 + [m_2]_t r_1 + r_m).
\end{aligned}$$

With $s = t r_1 r_2 + [m_1]_t r_2 + [m_2]_t r_1 + r_m \in R$, we obtain the equation for $\frac{t}{q} f^2 c_1 c_2$ with r_r as given above. The bound for $\|r_r\|_\infty$ follows from

$$\begin{aligned}
\|r_r\|_\infty &= \frac{1}{q}\|t[v_1 v_2]_\Delta - r_t(q)(\Delta[m_1]_t[m_2]_t + [m_1]_t v_2 + [m_2]_t v_1 + r_v)\|_\infty \\
&\leq \frac{1}{q}\left(\frac{\Delta t}{2} + r_t(q)\Big(\delta\frac{\Delta t^2}{4} + \delta\frac{t}{2}(\|v_1\|_\infty + \|v_2\|_\infty) + \|r_v\|_\infty\Big)\right) \\
&< \frac{1}{2} + r_t(q)\delta\Big(\frac{t}{4} + \frac{1}{2} + \frac{1}{2}\min_i \|v_i\|_\infty\Big),
\end{aligned}$$

where we have used the bound on $\|r_v\|_\infty$ and the fact that $\|v_i\|_\infty < \Delta/2$. □

D Proof of Theorem 2

To analyse how large $\tilde{v}_{\text{mult}}$ is, let $v_1, v_2 \in R$ be the inherent noise terms in c_1, c_2. Using $fc_i = \Delta[m_i]_t + v_i \pmod q$ and $\langle P_{q,w}(c_i), D_{q,w}(f)\rangle = fc_i \pmod q$ for $i \in \{1, 2\}$, this means we can write

$$\langle P_{q,w}(c_i), D_{q,w}(f)\rangle = \Delta[m_i]_t + v_i + q \cdot r_i \tag{2}$$

for polynomials $r_i \in R$. We assume that the assumptions in Lemma 1 are satisfied and that the v_i are chosen such that $\|v_i\|_\infty < (\Delta - r_t(q))/2$. In particular, c_1, c_2 are decryptable. Since the coefficients of $P_{q,w}(c_i)$ are bounded by $q/2$ in absolute value, those of $D_{q,w}(f)$ by $w/2$, and $D_{q,w}(f)$ has at most $\ell_{w,tB_{\text{key}}} = \lfloor \log_w(tB_{\text{key}}) \rfloor + 1$ non-zero entries, the polynomials r_i can be bounded by

$$\|r_i\|_\infty \leq \frac{1}{q}(\delta \ell_{w,tB_{\text{key}}} \frac{q}{2} \cdot \frac{w}{2} + \Delta \frac{t}{2} + \|v_i\|_\infty) \leq \frac{1}{4}\delta \ell_{w,tB_{\text{key}}} w + 1 < \frac{1}{2}\delta \ell_{w,tB_{\text{key}}} w. \quad (3)$$

Note that we have used $\Delta \leq q/t$, $\|v_i\|_\infty < \Delta/2$ and the definition of δ. Multiplying the scalar products $\langle P_{q,w}(c_1), D_{q,w}(f)\rangle$ and $\langle P_{q,w}(c_2), D_{q,w}(f)\rangle$, using $\langle P_{q,w}(c_1) \otimes P_{q,w}(c_2), D_{q,w}(f) \otimes D_{q,w}(f)\rangle = \langle P_{q,w}(c_1), D_{q,w}(f)\rangle \langle P_{q,w}(c_2), D_{q,w}(f)\rangle$ and substituting Equation (2) yields

$$\begin{aligned}\langle P_{q,w}(c_1) \otimes P_{q,w}(c_2), D_{q,w}(f) \otimes D_{q,w}(f)\rangle =& \Delta^2 [m_1]_t [m_2]_t + \Delta([m_1]_t v_2 + [m_2]_t v_1) \\ &+ q(v_1 r_2 + r_1 v_2) + v_1 v_2 \\ &+ q\Delta([m_1]_t r_2 + r_1 [m_2]_t) + q^2 r_1 r_2.\end{aligned}$$

Next, we incorporate $[m_1 m_2]_t$ into the above expression and scale by t/q. As observed in [8], simply scaling by Δ would give an additional error term caused by rounding of $q^2 r_1 r_2$. To make things more clear we expand the exposition of [8] for our scheme in Lemma 6, which is stated and proved in Appendix C. We obtain

$$\begin{aligned}\frac{t}{q}\langle P_{q,w}(c_1) \otimes P_{q,w}(c_2), D_{q,w}(f) \otimes D_{q,w}(f)\rangle = & \Delta[m_1 m_2]_t + [m_1]_t v_2 + [m_2]_t v_1 \quad (4) \\ & + t(v_1 r_2 + r_1 v_2) - r_t(q)([m_1]_t r_2 + [m_2]_t r_1 + r_m) \\ & + r_v + r_r + qs,\end{aligned}$$

where r_m, r_v, and r_r are as in Lemma 6 and we have $\|r_m\|_\infty < \frac{1}{2}\delta t$, $\|r_v\|_\infty \leq \frac{1}{2}\delta \min_i \|v_i\|_\infty$, and $\|r_r\|_\infty < \frac{1}{2}(1 + r_t(q)\delta(1 + \frac{t}{2} + \min_i \|v_i\|_\infty))$.

To bound the size of the inherent noise term in $\tilde{c}_{\text{mult}}$, we need to consider $\langle \tilde{c}_{\text{mult}}, D_{q,w}(f) \otimes D_{q,w}(f)\rangle$. Define

$$\begin{aligned} r_a &= \frac{t}{q}\langle P_{q,w}(c_1) \otimes P_{q,w}(c_2), D_{q,w}(f) \otimes D_{q,w}(f)\rangle - \langle \tilde{c}_{\text{mult}}, D_{q,w}(f) \otimes D_{q,w}(f)\rangle \\ &= \left\langle \left(\frac{t}{q} P_{q,w}(c_1) \otimes P_{q,w}(c_2)\right) - \left\lfloor \frac{t}{q} \cdot P_{q,w}(c_1) \otimes P_{q,w}(c_2) \right\rceil, D_{q,w}(f) \otimes D_{q,w}(f) \right\rangle. \quad (5)\end{aligned}$$

The coefficients of all polynomials in the vector in the left argument of the scalar product are bounded in absolute value by $1/2$, while those in the vector in the right argument are products of polynomials with coefficients bounded by $w/2$. Both vectors have length $\ell_{w,q}^2$, but at most $\ell_{w,tB_{\text{key}}}^2$ entries of $D_{q,w}(f) \otimes D_{q,w}(f)$ are non-zero, which means we get a bound on r_a as

$$\|r_a\|_\infty \leq \ell_{w,tB_{\text{key}}}^2 \delta \cdot \frac{1}{2} \cdot \delta \left(\frac{w}{2}\right)^2 = \frac{1}{8}(\delta \ell_{w,tB_{\text{key}}} w)^2.$$

We are now in a position to bound the inherent noise term in the intermediate ciphertext $\tilde{c}_{\text{mult}}$ after the first part of the homomorphic multiplication procedure. Again, this is very similar to [8].

We have $\langle \tilde{c}_{\text{mult}}, D_{q,w}(f) \otimes D_{q,w}(f)\rangle = \frac{t}{q}\langle P_{q,w}(c_1) \otimes P_{q,w}(c_2), D_{q,w}(f) \otimes D_{q,w}(f)\rangle - r_a = \Delta[m_1 m_2]_t + \tilde{v}_{\text{mult}} \pmod q$ from Equations (4) and (5), where we define

$$\tilde{v}_{\text{mult}} = [m_1]_t v_2 + [m_2]_t v_1 + t(v_1 r_2 + r_1 v_2) - r_t(q)([m_1]_t r_2 + [m_2]_t r_1 + r_m) + r_v + r_r - r_a.$$

It follows

$$\|\tilde{v}_{\text{mult}}\|_\infty \leq \|[m_1]_t v_2 + [m_2]_t v_1\|_\infty + t\|v_1 r_2 + r_1 v_2\|_\infty + r_t(q)\|[m_1]_t r_2 + [m_2]_t r_1\|_\infty$$
$$+ r_t(q)\|r_m\|_\infty + \|r_v\|_\infty + \|r_r\|_\infty + \|r_a\|_\infty.$$

We bound the summands separately and add together to obtain the overall bound. We have $\|[m_1]_t v_2 + [m_2]_t v_1\|_\infty \leq \delta \frac{t}{2}(\|v_1\|_\infty + \|v_2\|_\infty)$, since the coefficients of the $[m_i]_t$ are of absolute value at most $t/2$. Similarly, the bounds on $\|r_i\|_\infty$ lead to $\|v_1 r_2 + r_1 v_2\|_\infty \leq \delta \cdot \frac{1}{2}\delta \ell_{w,tB_{\text{key}}} w(\|v_1\|_\infty + \|v_2\|_\infty)$, and $\|[m_1]_t r_2 + [m_2]_t r_1\|_\infty \leq \delta t \frac{1}{2}\delta \ell_{w,tB_{\text{key}}} w$. We have already obtained bounds for $\|r_m\|_\infty$, $\|r_v\|_\infty$, $\|r_r\|_\infty$, and $\|r_a\|_\infty$ above. We summarize and get

$$\begin{aligned}\|\tilde{v}_{\text{mult}}\|_\infty < {} & \frac{1}{2}\delta t(\|v_1\|_\infty + \|v_2\|_\infty) + \frac{1}{2}\delta^2 t \ell_{w,tB_{\text{key}}} w(\|v_1\|_\infty + \|v_2\|_\infty) \\ & + \frac{1}{2} r_t(q)\delta^2 t \ell_{w,tB_{\text{key}}} w + \frac{1}{2} r_t(q)\delta t + \frac{1}{2}\delta \min_i \|v_i\|_\infty \\ & + \frac{1}{2}(1 + r_t(q)\delta(1 + \frac{t}{2} + \min_i \|v_i\|_\infty)) + \frac{1}{8}(\delta \ell_{w,tB_{\text{key}}} w)^2.\end{aligned} \tag{6}$$

We simplify the expression by replacing the $\|v_i\|_\infty$ by a common upper bound V, e.g. $V = \max\{\|v_1\|_\infty, \|v_2\|_\infty\}$. This makes sense if the inherent noise terms are known to be of roughly the same size. If they are of different magnitudes, one gets more precise bounds by using the more complicated formulas that keep these sizes as separate inputs. Using $r_t(q) < t$, we obtain the claimed bound. □

E Proof of Lemma 3

It is $fc = fc_{\text{mult}} = \langle \tilde{c}_{\text{mult}}, D_{q,w}(f) \otimes D_{q,w}(f)\rangle + f\langle D_{q,w}(\tilde{c}_{\text{mult}}), e\rangle + gt\langle D_{q,w}(\tilde{c}_{\text{mult}}), s\rangle \pmod q$. We substitute $\langle \tilde{c}_{\text{mult}}, D_{q,w}(f) \otimes D_{q,w}(f)\rangle$ from Theorem 2 and obtain $fc = \Delta[m_1 m_2]_t + v_{\text{mult}} \pmod q$ with

$$v_{\text{mult}} = \tilde{v}_{\text{mult}} + f\langle D_{q,w}(\tilde{c}_{\text{mult}}), \boldsymbol{e}\rangle + tg\langle D_{q,w}(\tilde{c}_{\text{mult}}), \boldsymbol{s}\rangle \pmod q.$$

We bound $\|\langle D_{q,w}(\tilde{c}_{\text{mult}}), \boldsymbol{e}\rangle\|_\infty \leq \delta \ell_{w,q}^3 \frac{w}{2} B_{\text{err}}$. We get the same bound when $\boldsymbol{e}$ is replaced by $\boldsymbol{s}$. With $\|f\|_\infty, \|tg\|_\infty < tB_{\text{key}}$ we deduce $\|v_{\text{mult}}\|_\infty < \|\tilde{v}_{\text{mult}}\|_\infty + 2\delta \cdot tB_{\text{key}} \cdot \delta \ell_{w,q}^3 \frac{w}{2} B_{\text{err}}$, which proves the lemma. □

F Proof of Theorem 3

Let $m_1, m_2 \in R$ and let $c_i = \mathsf{LHE.Encrypt}(\mathsf{pk}, m_i)$ for $i \in \{1, 2\}$. Denote by $v_i \in R$ the inherent noise term in c_i and suppose $\|v_i\|_\infty \leq V < \Delta/2$. Let $c_{\text{add}} = \mathsf{LHE.Add}(c_1, c_2)$ and $c_{\text{mult}} = \mathsf{LHE.Mult}(c_1, c_2, \boldsymbol{\gamma})$ such that $fc_{\text{add}} = \Delta[m_1 + m_2]_t + v_{\text{add}} \pmod q$ and $fc_{\text{mult}} = \Delta[m_1 m_2]_t + v_{\text{mult}} \pmod q$. Note since $w = 2$, $\ell_{w,q} = \lfloor \log_2(q) \rfloor + 2$. Then combining Lemma 3 and Theorem 2 we deduce that

$$\|v_{\text{mult}}\|_\infty < \delta t(2 + \delta \ell_{w,tB_{\text{key}}} w)V + \frac{1}{2}\delta^2 \ell_{w,tB_{\text{key}}} w(t^2 + \ell_{w,tB_{\text{key}}} w) + \delta^2 t \ell_{w,q}^3 w B_{\text{err}} B_{\text{key}}.$$

Next, we introduce a common bound $V > 0$ for the size of the inherent noise in fresh ciphertexts, and assume that the noise growth for homomorphic additions can be neglected when compared to that for multiplications. By iterating the previous bound for L levels of multiplications, we deduce that for a depth L circuit consisting of additions and multiplications the noise in the ciphertext is bounded by $C_1^L \cdot V + LC_1^{L-1}C_2 = C_1^L \cdot V(1 + LC_2/C_1V)$ where

$$C_1 = \delta t(2 + \delta \ell_{w,tB_{\text{key}}} w)$$
$$C_2 = \tfrac{1}{2}\delta^2 \ell_{w,tB_{\text{key}}} w(t^2 + \ell_{w,tB_{\text{key}}} w) + \delta^2 t \ell_{w,q}^3 w B_{\text{err}} B_{\text{key}}$$

Now observe that $C_1 = \text{poly}(d)\log(q)$ since $\delta = \phi(d) = d/2$. Moreover, $C_2/C_1V = \mathcal{O}(\log(q)^2)$. Hence overall, to guarantee correctness, we have that

$$q/V = \mathcal{O}(L \cdot \text{poly}(d)^L \cdot \log(q)^{L+2}).$$

Substituting $V = \text{poly}(d) \cdot q^k$ for some $k \in (1/2, 1)$, it can be seen that the above can be satisfied if

$$L = \mathcal{O}\left(\frac{(1-k)\log(q)}{\log(\log(q)) + \log(d)}\right).$$

G Proof of Theorem 6

Let $v_1, v_2 \in R$ be the inherent noise terms in c_1, c_2. This means we can write $fc_i = \Delta[m_i]_t + v_i + q \cdot r_i$ for polynomials $r_i \in R$. We assume that the assumptions in Lemma 1 are satisfied and that the v_i are chosen such that $\|v_i\|_\infty < (\Delta - r_t(q))/2$. In particular, c_1, c_2 are decryptable. The polynomials r_i can be bounded by

$$\|r_i\|_\infty \le \frac{1}{q}\|fc_i - \Delta[m_i]_t - v_i\|_\infty < \frac{1}{q}\left(\delta t B_{\text{key}}\frac{q}{2} + \Delta\frac{t}{2} + \|v_i\|_\infty\right).$$

Note that we have used the definition of δ and the bound for the secret key $\|f\|_\infty < tB_{\text{key}}$. Since $\Delta \le q/t$ and $\|v_i\|_\infty < \Delta/2$ we obtain a bound on the r_i as $\|r_i\|_\infty < \frac{1}{2}\delta t B_{\text{key}} + 1$. Multiplying fc_1 and fc_2 and substituting the above expression for the fc_i yields

$$\begin{aligned} f^2 c_1 c_2 =& \Delta^2[m_1]_t[m_2]_t + \Delta([m_1]_t v_2 + [m_2]_t v_1) + q(v_1 r_2 + r_1 v_2) + \\ & v_1 v_2 + q\Delta([m_1]_t r_2 + r_1[m_2]_t) + q^2 r_1 r_2. \end{aligned}$$

Lemma 6 in Appendix C incorporates $[m_1 m_2]_t$ and scales by t/q, i.e. it gives us an expression for $\frac{t}{q}f^2 c_1 c_2$ which contains $\Delta[m_1 m_2]_t$ as a summand. But to bound the size of the inherent noise term in $\tilde{c}_{\text{mult}}$, we need to consider $f^2\tilde{c}_{\text{mult}} = f^2\lfloor \frac{t}{q}c_1 c_2 \rceil$. The difference between both terms is

$$r_a = \frac{t}{q}f^2 c_1 c_2 - f^2 \tilde{c}_{\text{mult}} = \frac{t}{q}f^2 c_1 c_2 - f^2\left\lfloor \frac{t}{q}c_1 c_2 \right\rceil. \tag{7}$$

This difference can be bounded by

$$\|r_a\|_\infty = \|f^2\left(\frac{t}{q}c_1 c_2 - \left\lfloor \frac{t}{q}c_1 c_2 \right\rceil\right)\|_\infty \le \frac{1}{2}\delta^2\|f\|_\infty^2 \le \frac{1}{2}(\delta t B_{\text{key}})^2.$$

In the following lemma, we bound the inherent noise term in the intermediate ciphertext $\tilde{c}_{\text{mult}}$ after the first part of the homomorphic multiplication procedure. This is very similar to [8]. To prove Theorem 6, we simplify the expression for the bound obtained in Lemma 7 by replacing the $\|v_i\|_\infty$ by the common upper bound V. Using $r_t(q) < t$, we obtain the bound in Theorem 6.

Lemma 7. *Let $c_1, c_2 \in R$ be ciphertexts encrypting $m_1, m_2 \in R$, which are decryptable with the secret key f. Assume that the inherent noise terms $v_1, v_2 \in R$ in c_1, c_2 satisfy $\|v_i\|_\infty < \Delta/2$. Let $f = 1 + tf'$ with $f' \leftarrow \chi_{\text{key}}$ where $\|\chi_{\text{key}}\|_\infty < B_{\text{key}}$. Define*

$$\tilde{c}_{\text{mult}} = \left\lfloor \frac{t}{q} \cdot c_1 c_2 \right\rceil.$$

Then $f^2 \tilde{c}_{\text{mult}} = \Delta [m_1 m_2]_t + \tilde{v}_{\text{mult}} \pmod{q}$ where

$$\begin{aligned}\|\tilde{v}_{\text{mult}}\|_\infty < \frac{1}{2}\delta &\left(t(2 + \delta t B_{\text{key}})(\|v_1\|_\infty + \|v_2\|_\infty) + (1 + r_t(q)) \min_i \|v_i\|_\infty \right) \\ &+ \frac{1}{2}\left(1 + (\delta t B_{\text{key}})^2 + r_t(q)\delta t(3 + \delta t B_{\text{key}})\right).\end{aligned}$$

Proof. Equation (5) and Lemma 6 yield $f^2 \tilde{c}_{\text{mult}} = \frac{t}{q} f^2 c_1 c_2 - r_a = \Delta[m_1 m_2]_t + \tilde{v}_{\text{mult}} \pmod{q}$, where we define

$$\tilde{v}_{\text{mult}} = [m_1]_t v_2 + [m_2]_t v_1 + t(v_1 r_2 + r_1 v_2) - r_t(q)([m_1]_t r_2 + [m_2]_t r_1 + r_m) + r_v + r_r - r_a.$$

It follows

$$\begin{aligned}\|\tilde{v}_{\text{mult}}\|_\infty \le \|[m_1]_t v_2 + [m_2]_t v_1\|_\infty &+ t\|v_1 r_2 + r_1 v_2\|_\infty + r_t(q)\|[m_1]_t r_2 + [m_2]_t r_1\|_\infty \\ &+ r_t(q)\|r_m\|_\infty + \|r_v\|_\infty + \|r_r\|_\infty + \|r_a\|_\infty.\end{aligned}$$

We bound the summands separately and add together to obtain the overall bound. We have $\|[m_1]_t v_2 + [m_2]_t v_1\|_\infty \le \delta \frac{t}{2}(\|v_1\|_\infty + \|v_2\|_\infty)$, since the coefficients of the $[m_i]_t$ are of absolute value at most $t/2$. Similarly, the bounds on $\|r_i\|_\infty$ lead to $\|v_1 r_2 + r_1 v_2\|_\infty \le \delta \cdot \frac{1}{2}(\delta t B_{\text{key}} + 1)(\|v_1\|_\infty + \|v_2\|_\infty)$, and $\|[m_1]_t r_2 + [m_2]_t r_1\|_\infty \le \delta t \frac{1}{2}(\delta t B_{\text{key}} + 1)$. We have already obtained bounds for $\|r_m\|_\infty$, $\|r_v\|_\infty$, $\|r_r\|_\infty$, and $\|r_a\|_\infty$ above. We summarize and get

$$\begin{aligned}\|\tilde{v}_{\text{mult}}\|_\infty &< \frac{1}{2}\delta t(\|v_1\|_\infty + \|v_2\|_\infty) + \frac{1}{2}\delta t(\delta t B_{\text{key}} + 1)(\|v_1\|_\infty + \|v_2\|_\infty) \\ &+ \frac{1}{2} r_t(q)\delta t(\delta t B_{\text{key}} + 1) + \frac{1}{2} r_t(q)\delta t + \frac{1}{2}\delta \min_i \|v_i\|_\infty \\ &+ \frac{1}{2}(1 + r_t(q)\delta(1 + \frac{t}{2} + \min_i \|v_i\|_\infty)) + \frac{1}{2}(\delta t B_{\text{key}})^2.\end{aligned}$$

A rearrangement of this lengthy expression together with the fact that $1 + t/2 \le t$ for $t \ge 2$ yields the claimed bound for $\|\tilde{v}_{\text{mult}}\|_\infty$. ☐

H Proof of Lemma 5

It is $fc = f\langle D_{q,w}(\tilde{c}_{\text{mult}}), \boldsymbol{\gamma}\rangle = f\langle D_{q,w}(\tilde{c}_{\text{mult}}), P_{q,w}(f) + \boldsymbol{e} + h\boldsymbol{s}\rangle = f^2\tilde{c}_{\text{mult}} + f\langle D_{q,w}(\tilde{c}_{\text{mult}}), \boldsymbol{e}\rangle + tg\langle D_{q,w}(\tilde{c}_{\text{mult}}), \boldsymbol{s}\rangle \pmod{q}$. We use $f^2\tilde{c}_{\text{mult}} = \Delta[m_1 m_2]_t + \tilde{v}_{\text{mult}}$ and obtain $fc = \Delta[m_1 m_2]_t + v_{\text{mult}} \pmod{q}$ with

$$v_{\text{mult}} = \tilde{v}_{\text{mult}} + f\langle D_{q,w}(\tilde{c}_{\text{mult}}), \boldsymbol{e}\rangle + tg\langle D_{q,w}(\tilde{c}_{\text{mult}}), \boldsymbol{s}\rangle \pmod{q}.$$

We bound $\|\langle D_{q,w}(\tilde{c}_{\text{mult}}), \boldsymbol{e}\rangle\|_\infty = \|\sum_{i=0}^{\ell_{w,q}-1}[\tilde{c}_i]_w e_i\|_\infty \le \delta \ell_{w,q} \frac{w}{2} B_{\text{err}}$, where $\tilde{c}_{\text{mult}} = \sum[\tilde{c}_i]_w w^i$ and $\boldsymbol{e} = (e_i)$. We get the same bound when $\boldsymbol{e}$ is replaced by $\boldsymbol{s}$. With $\|f\|_\infty, \|tg\|_\infty < tB_{\text{key}}$ we deduce $\|v_{\text{mult}}\|_\infty < \|\tilde{v}_{\text{mult}}\|_\infty + 2\delta \cdot tB_{\text{key}} \cdot \delta\ell_{w,q}\frac{w}{2}B_{\text{err}}$, which proves the lemma.

I Detailed Noise Bounds for LHE'

The following lemma summarizes detailed bounds on the noise growth during homomorphic addition and multiplication operations. These bounds are more detailed than the ones given in Section 6.1 and depend on the individual inherent noise sizes $\|v_i\|_\infty$ in each of the ciphertexts. Depending on the exact computation that is to be done, these bounds might be more accurate and might lead to more efficient parameters than the more coarse bounds given in Section 6.1.

Lemma 8. *Let* $(\mathsf{pk}, \mathsf{sk}, \mathsf{evk}) = (h, f, \boldsymbol{\gamma}) \in R^2$ *be output by* $\mathsf{LHE.KeyGen}$*. Let* $m_1, m_2 \in R$ *and let* $c_i = \mathsf{LHE.Encrypt}(\mathsf{pk}, m_i)$ *for* $i \in \{1, 2\}$*. Denote by* $v_i \in R$ *the inherent noise term in* c_i *and suppose* $\|v_i\|_\infty < \Delta/2$*. Let* $c_{\mathrm{add}} = \mathsf{LHE.Add}(c_1, c_2)$ *and* $c_{\mathrm{mult}} = \mathsf{LHE.Mult}(c_1, c_2, \boldsymbol{\gamma})$ *be such that* $fc_{\mathrm{add}} = \Delta[m_1 + m_2]_t + v_{\mathrm{add}} \pmod q$ *and* $fc_{\mathrm{mult}} = \Delta[m_1 m_2]_t + v_{\mathrm{mult}} \pmod q$*. Let* $w > 1$ *be the word length and let* $\ell_{w,q} = \lceil \log_w(q) \rceil$*. Then*

$$\begin{aligned}\|v_{\mathrm{add}}\|_\infty &\le \|v_1\|_\infty + \|v_2\|_\infty + r_t(q),\\ \|v_{\mathrm{mult}}\|_\infty &< \frac{1}{2}\delta \left(t(2 + \delta t B_{\mathrm{key}})(\|v_1\|_\infty + \|v_2\|_\infty) + (1 + r_t(q)) \min_i \|v_i\|_\infty \right)\\ &\quad + \frac{1}{2}\left(1 + (\delta t B_{\mathrm{key}})^2 + r_t(q)\delta t(3 + \delta t B_{\mathrm{key}})\right) + \delta^2 t \ell_{w,q} w B_{\mathrm{err}} B_{\mathrm{key}}.\end{aligned}$$

Proof. This is a combination of Inequality (1) and Lemmas 7 and 5. □

J Proof of Theorem 7

The proof is subdivided into the following corollary and lemma. The corollary simply combines the noise growth bounds from the first step of the multiplication and from the key switching part based on a common upper bound V for the inherent noise of the original ciphertexts.

Corollary 1. *In addition to the assumptions made in Lemma 8, let* $V > 0$ *be such that* $\|v_i\|_\infty \le V < \Delta/2$ *for* $i \in \{1, 2\}$*. Then the norms of the inherent noise terms* v_{add} *and* v_{mult} *satisfy* $\|v_{\mathrm{add}}\|_\infty \le 2V + r_t(q) < 2V + t$ *and* $\|v_{\mathrm{mult}}\|_\infty < C_1 V + C_2$*, for*

$$C_1 = (1 + \epsilon_1)\delta^2 t^2 B_{\mathrm{key}}, \quad C_2 = \delta^2 t B_{\mathrm{key}} \left(\frac{1}{2} t (B_{\mathrm{key}} + t) + \ell_{w,q} w B_{\mathrm{err}} \right), \quad \epsilon_1 = 3(\delta t B_{\mathrm{key}})^{-1}.$$

Proof. Theorem 6 shows that we can take $C_1 = \delta t(3 + \delta t B_{\mathrm{key}})$, which is equal to the above expression. We get the constant C_2 from Theorem 6 and Lemma 5. □

The following lemma iterates L levels of multiplications and deduces an overall noise bound for this operation.

Lemma 9. *Let* $c \in R$ *be a ciphertext that is the homomorphic product of ciphertexts of inherent noise size at most* V *arranged in a tree of* L *levels of multiplications. Let* $v \in R$ *be the inherent noise term in* c*. Then the norm of* v *can be bounded by*

$$\|v\|_\infty < C_1^L V + L C_1^{L-1} C_2.$$

Using the bounds in Corollary 1, we obtain

$$\begin{aligned}\|v\|_\infty &< (1 + \epsilon_1)^L (\delta t)^{2L} B_{\mathrm{key}}^L V + L(1 + \epsilon_1)^{L-1} \delta^{2L} t^{2L-1} B_{\mathrm{key}}^L \left(\frac{1}{2} t (B_{\mathrm{key}} + t) + \ell_{w,q} w B_{\mathrm{err}} \right)\\ &= (1 + \epsilon_1)^{L-1} \delta^{2L} t^{2L-1} B_{\mathrm{key}}^L \left((1 + \epsilon_1) t V + L \left(\frac{1}{2} t (B_{\mathrm{key}} + t) + \ell_{w,q} w B_{\mathrm{err}} \right) \right).\end{aligned}$$

Proof. This follows by induction on L by repeatedly applying the bound in Corollary 1. □

Theorem 7 follows from Lemma 9 and Lemma 1.

K Details on Parameter Selection

This section discusses in more detail the parameter selection recommendations made in Table 1. We use $B_{\text{key}} = 1$, in other words we are assuming that even when the polynomials f', g have coefficients in $\{-1, 0, 1\}$, the public key $h = [tgf^{-1}]_q$ is indistinguishable from uniform. The standard deviation of the error distribution is fixed at $\sigma_{\text{err}} = 8$; this is consistent with [18]. The high probability bound on the size of the coefficients of errors drawn from Gaussian distributions is chosen as $6\sigma_{\text{err}}$.

To distinguish with an advantage of ϵ in the RLWE problem, an adversary is required to find vectors of length at most $\alpha \cdot (q/\sigma)$ where $\alpha = \sqrt{\ln(1/\epsilon)/\pi}$. In our specific parameter examples, we use $\epsilon = 2^{-80}$, which results in $\alpha \approx 4.201$. We refer to [14] for a more complete description of a distinguishing attack and the precise lattices we are required to find short vectors in. Running Schnorr-Euchner's BKZ [21], the best known lattice reduction algorithm known in practice, and its successor BKZ 2.0 [6] for security parameter λ (following [11] we use $\lambda = 80$) one expects to find vectors of length $2^{2\sqrt{n \log_2(q) \log_2(\delta_{\text{RHF}})}}$ in time $T_{\text{BKZ}} = 2^{\lambda}$ where δ_{RHF} is the so-called root Hermite factor. This latter quantity is the overwhelming factor determining the quality of the basis which can be achieved in a given time and is computed from $\log_2(T_{\text{BKZ}}) = 1.8/\log_2(\delta_{\text{RHF}}) - 110$ [14]. It is currently infeasible to achieve a target root Hermite factor $\delta_{\text{RHF}} < 1.005$ [6]. To guarantee security, we require that the shortest vector obtained through lattice reduction is longer than a vector which could give an adversary a non-negligible advantage ϵ in the Ring-LWE distinguishing problem. This means that for security we thus require $\alpha \cdot q/\sigma < 2^{2\sqrt{n \log_2(q) \log_2(\delta_{\text{RHF}})}}$. For fixed parameters α and δ_{RHF}, this inequality provides bounds on the remaining parameters q, σ_{err} and n. Fixing σ_{err} too ($\sigma_{\text{err}} = 8$ here), we get a dependency between q and n that is expressed in the two settings discussed above as follows. When we fix q, we obtain a lower bound $n_{\min}$ for the dimension n to guarantee security against the distinguishing attack. For the example values for the sizes of q given in the first column of the left part of Table 1, we list this minimal degree in the second column. We used the worst case bound for a modulus q of that size. Vice versa, first fixing the degree n means that we get an upper bound $q_{\max}$ for q. We display the relation between n and the size $\log(q_{\max})$ in the first two columns of the right part of Table 1.

For guaranteeing correctness, we use the noise bounds derived in the previous section. As mentioned in Section 2, when d is a power of 2 and thus $\Phi_d(x) = x^n + 1$, the expansion factor is $\delta = n$. Then, by Lemma 1 and Lemma 9 we know that our scheme can correctly evaluate a depth L circuit as long as $(1+\epsilon_1)^{L-1} n^{2L} t^{2L-1} B_{\text{key}}^{L} \left((1+\epsilon_1)tV + L\left(\frac{1}{2}t(B_{\text{key}} + t) + \ell_{w,q} w B_{\text{err}}\right)\right) < (\Delta - r_t(q))/2$ holds, where $\epsilon_1 = 3(ntB_{\text{key}})^{-1}$ and $V = ntB_{\text{key}}(2B_{\text{err}} + r_t(q)/2)$ is the inherent noise of fresh ciphertexts by Lemma 2. For each row in either the left or the right part of Table 1, we take the given values for q and n together with different values for t and check what is the maximum number of levels $L_{\max}$ for which the correctness bound still holds. Note that in the left part, we take the minimal degree $n_{\min}$. This means that when choosing a power of 2 for the degree, the values for $L_{\max}$ might change. In the right part, we take the largest possible value for q with the given maximal bit size.

It is important to ensure that the security bounds as well as the correctness bounds are both satisfied. Note that the authors of [8] failed to check their parameters presumably obtained from the correctness bound in the security bound, too, resulting in insecure parameters of $q = 2^{1358}$ and $n = 2^{10}$. This shows that obtaining parameters for homomorphic encryption schemes is non-trivial.