Faster Fully Homomorphic Encryption

Damien Stehlé[1,2] and Ron Steinfeld[2]

[1] CNRS/Department of Mathematics and Statistics (F07), University of Sydney NSW 2006, Australia.
damien.stehle@gmail.com – http://perso.ens-lyon.fr/damien.stehle
[2] Centre for Advanced Computing - Algorithms and Cryptography,
Department of Computing, Macquarie University, NSW 2109, Australia
rons@ics.mq.edu.au – http://www.ics.mq.edu.au/~rons/

Abstract. We describe two improvements to Gentry's fully homomorphic scheme based on ideal lattices and its analysis: we provide a refined analysis of one of the hardness assumptions (the one related to the Sparse Subset Sum Problem) and we introduce a probabilistic decryption algorithm that can be implemented with an algebraic circuit of low multiplicative degree. Combined together, these improvements lead to a faster fully homomorphic scheme, with a $\widetilde{O}(\lambda^3)$ bit complexity per elementary binary add/mult gate, where λ is the security parameter. These improvements also apply to the fully homomorphic schemes of Smart and Vercauteren [PKC'2010] and van Dijk et al. [Eurocrypt'2010].

Keywords: fully homomorphic encryption, ideal lattices, SSSP.

1 Introduction

A homomorphic encryption scheme allows any party to publicly transform a collection of ciphertexts for some plaintexts $\pi_1, \ldots, \pi_n$ into a ciphertext for some function/circuit $f(\pi_1, \ldots, \pi_n)$ of the plaintexts, without the party knowing the plaintexts themselves. Such schemes are well known to be useful for constructing privacy-preserving protocols, for example as required in 'cloud computing' applications: a user can store encrypted data on a server, and allow the server to process the encrypted data without revealing the data to the server. For over 30 years, all known homomorphic encryption schemes supported only a limited set of functions f, which restricted their applicability. The theoretical problem of constructing a *fully* homomorphic encryption scheme supporting arbitrary functions f, was only recently solved by the breakthrough work of Gentry [11]. More recently, two further fully homomorphic schemes were presented [27, 7], following Gentry's framework. The underlying tool behind all these schemes is the use of *Euclidean lattices*, which have previously proved powerful for devising many cryptographic primitives (see, e.g., [22] for a recent survey).

A central aspect of Gentry's fully homomorphic scheme (and the subsequent schemes) is the *ciphertext refreshing* (Recrypt) operation. The ciphertexts in Gentry's scheme contain a random 'noise' component that grows in size as the ciphertext is processed to homomorphically evaluate a function f on its plaintext. Once the noise size in the ciphertext exceeds a certain threshold, the ciphertext can no longer be decrypted correctly. This limits the number of homomorphic operations that can be performed. To get around this limitation, the Recrypt operation allows to 'refresh' a ciphertext, i.e., given a ciphertext ψ for some plaintext π, to compute a new ciphertext ψ' for π (possibly for a different key), but such that the size of the noise in ψ' is smaller than the size of the noise in ψ. By periodically refreshing the ciphertext (e.g., after computing each gate in f), one can then evaluate arbitrarily large circuits f.

The Recrypt operation is implemented by evaluating the decryption circuit of the encryption scheme homomorphically, given 'fresh' (low noise) ciphertexts for the bits of the ciphertext to be refreshed and the scheme's secret key. This homomorphic computation of the decryption circuit must of course be possible *without* any ciphertext refreshing, a condition referred to as *bootstrappability*. Thus, the complexity (in particular circuit depth, or multiplicative degree) of the scheme's decryption circuit is of fundamental importance to the feasibility and complexity of the fully homomorphic scheme. Unfortunately, the relatively high complexity of the decryption circuit in the schemes [11, 27, 7], together with the tension between the bootstrappability condition and the security of the underlying hard problems, implies the need for large parameters and leads to resulting encryption schemes of high bit-complexity.

Our Contributions. We present improvements to Gentry's fully homomorphic scheme [11] and its analysis, that reduce its complexity. Overall, letting λ be the security parameter (i.e., all known attacks against the scheme take time at least 2^λ), we obtain a $\widetilde{O}(\lambda^3)$ bit complexity for refreshing a ciphertext corresponding to a 1-bit plaintext. This is the cost per gate of the fully homomorphic scheme. To compare with, Gentry [10, Ch. 12] claims a $\widetilde{O}(\lambda^6)$ bit complexity for the same task, although the proof is incomplete.[3]

Our improved complexity stems from two sources. First, we give a more precise security analysis of the Sparse Subset Sum Problem (SSSP) against lattice attacks, compared to the analysis given in [11]. The SSSP, along with the Ideal lattice Bounded Distance Decoding (BDD) problem, are the two hard problems underlying the security of Gentry's fully homomorphic scheme. In his security analysis of BDD, Gentry uses the best known complexity bound for the approximate shortest vector problem (SVP) in lattices, but in analyzing SSSP, Gentry assumes the availability of an exact SVP oracle. Our new finer analysis of SSSP takes into account the complexity of approximate SVP, making it more consistent with the assumption underlying the analysis of the BDD problem, and leads to smaller parameter choices. Note that we actually use a vector variant of SSSP, which seems more resistant to lattice attacks, but looks somewhat less natural.[4] Second, we relax the definition of fully homomorphic encryption to allow for a negligible but non-zero probability of decryption error. We then show that, thanks to the randomness underlying Gentry's 'SplitKey' key generation for his squashed decryption algorithm (i.e., the decryption algorithm of the bootstrappable scheme), if one allows a negligible decryption error probability, then the rounding precision used in representing the ciphertext components can be approximately halved, compared to the precision in [11] which guarantees zero error probability. The reduced ciphertext precision allows us to decrease the degree of the decryption circuit. We mainly concentrate on Gentry's scheme [11], but our improvements apply equally well to the other related schemes [27, 7].

Road-map. In Section 2, we provide the background that is necessary to the understanding of our results. Section 3 contains a summary of Gentry's fully homomorphic encryption scheme. Section 4 contains our first contribution: an improved analysis of the hardness of the SSSP problem against lattice attacks. In Section 5, we present our second contribution: an improvement to Gentry's ciphertext refreshing ('recrypt') algorithm. Then, in Section 6, we analyze the implications of our improvements on the asymptotic efficiency of Gentry's scheme, and finally in Section 7 we discuss how our work can be adapted to other fully homomorphic schemes.

[3] This bound is claimed to hold for the scheme after Optimizations 1 and 2 of Section 12.3, but the analysis does not include the cost of the ciphertext expansion nor details which decryption circuit is applied homomorphically. For instance, evaluating the decryption circuit from [7, Le. 6.3] is too costly to derive the $\widetilde{O}(\lambda^6)$ bound. These gaps in the complexity analysis can be filled using the results of the present article, and the bound $\widetilde{O}(\lambda^6)$ indeed holds.

[4] Our finer analysis also gives an improved complexity assuming the hardness of the more natural integer SSSP problem used by Gentry, but in this case the resulting ciphertext refreshing complexity is higher, namely $\widetilde{O}(\lambda^{3.5})$ bit operations.

NOTATION. Vectors will be denoted in bold. If $\boldsymbol{x} \in \mathbb{R}^n$, then $\|\boldsymbol{x}\|$ denotes the Euclidean norm of $\boldsymbol{x}$. We make use of the Landau notations $O(\cdot), \widetilde{O}(\cdot), o(\cdot), \omega(\cdot), \Omega(\cdot), \widetilde{\Omega}(\cdot), \Theta(\cdot), \widetilde{\Theta}(\cdot)$. If n grows to infinity, we say that a function $f(n)$ is negligible if it is asymptotically $\leq n^{-c}$ for any $c > 0$. If X is a random variable, $E[X]$ denotes its mean and $\Pr[X = x]$ denotes the probability of the event $X = x$. We say that a sequence of events E_n holds with overwhelming probability if $\Pr[\neg E_n] \leq f(n)$ for a negligible function f. If D_1 and D_2 are two probability distributions over a discrete domain E, their statistical distance is $\frac{1}{2}\sum_{x \in E}|D_1(x) - D_2(x)|$. We will use the following variant of the well-known Hoeffding bound [13, Th. 2].

Lemma 1.1. *Let $X_1, \ldots, X_t$ denote independent random variables with mean μ, where $X_i \in [a_i, b_i]$ for some real vectors $\boldsymbol{a}, \boldsymbol{b}$. Let $X = \sum_i X_i$. Then, for any $k \geq 0$, the following bound holds:*

$$\Pr[|X - t\mu| \geq k] \leq 2 \cdot \exp(-2k^2/\|\boldsymbol{b} - \boldsymbol{a}\|^2).$$

2 Reminders

For a detailed introduction to the computational aspects of lattices, we refer to [21]. The article [12] provides an intuitive description of Gentry's fully homomorphic scheme.

2.1 Euclidean lattices

An n-dimensional lattice L is the set of all integer linear combinations of some linearly independent vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n \in \mathbb{Z}^n$, i.e., $L = \sum \mathbb{Z}\boldsymbol{b}_i$. The $\boldsymbol{b}_i$'s are called a basis of L. A basis $B = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n) \in \mathbb{Z}^{n \times n}$ is said to be in Hermite Normal Form (HNF) if $b_{i,j} = 0$ for $i > j$ and $0 \leq b_{i,j} < b_{i,i}$ otherwise. The HNF of a lattice is unique and can be computed in polynomial time given any basis, which arguably makes it a worst-case basis [20]. To a basis $B = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n) \in \mathbb{Z}^{n \times n}$ for lattice L, we associate the fundamental parallelepiped $\mathcal{P}(B) = \{\boldsymbol{v} = \sum_i y_i \cdot \boldsymbol{b}_i : y_i \in (-1/2, 1/2]\}$. For a vector $\boldsymbol{v} \in \mathbb{R}^n$, we denote by $\boldsymbol{v} \bmod B$ the unique vector $\boldsymbol{v}' \in \mathcal{P}(B)$ such that $\boldsymbol{v} - \boldsymbol{v}' \in L$. Note that $\boldsymbol{v}' = \boldsymbol{v} - B\lfloor B^{-1}\boldsymbol{v}\rceil$, where $\lfloor \cdot \rceil$ rounds the coefficients to the nearest integers (upwards in case of a real that is equally distant to two consecutive integers).

The minimum $\lambda_1(L)$ is the norm of any shortest non-zero vector in L. We now define two parametrized families of algorithmic problems that are central for euclidean lattices. Let $\gamma \geq 1$ be a function of the dimension. The γ-SVP (for Shortest Vector Problem) computational problem consists in finding a vector $\boldsymbol{b} \in L$ such that $0 < \|\boldsymbol{b}\| \leq \gamma\lambda_1(L)$, given as input an arbitrary basis for L. The γ-BDD (for Bounded Distance Decoding) computational problem consists in finding a vector $\boldsymbol{b} \in L$ closest to $\boldsymbol{t}$ given as inputs an arbitrary basis for L and a target vector $\boldsymbol{t}$ whose distance to L is $\leq \gamma\lambda_1(L)$. Solving γ-SVP and γ-BDD are computationally hard problem. The best algorithms for solving them for $\gamma = 1$ ([14, 2, 3]) run in time exponential with respect to the dimension. Oppositely, the smallest γ one can achieve in polynomial time is exponential, up to poly-logarithmic factors in the exponent ([18, 25, 4]). For intermediate γ, the best strategy is the hierarchical reductions of [25], and leads to the following conjecture.

Lattice 'Rule of Thumb' Conjecture. There exists an absolute constant c such that for any λ and any dimension n, one cannot solve γ-SVP (resp. γ-BDD) in time smaller 2^λ, with $\gamma = c^{n/\lambda}$.

There have been many improvements since the inventions of the algorithms above (see, e.g., [8, 23, 16]), but so far they have only lead to improved constants, without changing the overall framework.

The conjecture above also seems to hold even if one considers quantum computations [19]. In the present work, we will consider this conjecture for several different families of lattices: no algorithm is known to perform non-negligibly better for these than for more general lattices.

For a lattice L, we define $\det(L)$ as the magnitude of the determinant of any of its bases. Minkowski's theorem provides a link between the minimum and the volume of a given lattice.

Theorem 2.1 ([6, III.2.2]). *Let L be an n-dimensional lattice and V be a compact convex set that is symmetric about the origin. Let $m \geq 1$ be an integer. If $\mathrm{vol}(V) \geq m 2^n \det(L)^{1/n}$, then V contains at least m non-zero pairs of points $\pm\mathbf{b}$ of L.*

2.2 Ideal lattices

Let $f \in \mathbb{Z}[x]$ a monic degree n irreducible polynomial. Let R denote the polynomial ring $\mathbb{Z}[x]/f$. Let I be an (integral) ideal of R, i.e., a subset of R that is closed under addition, and multiplication by arbitrary elements of R. By mapping polynomials to the vectors of their coefficients, we see that the ideal I corresponds to a sublattice of $\mathbb{Z}^n$: we can thus view I as both a lattice and an ideal. An *ideal lattice* for f is a sublattice of $\mathbb{Z}^n$ that corresponds to an ideal $I \subseteq \mathbb{Z}[x]/f$. In the following, an ideal lattice will implicitly refer to an f-ideal lattice. For $v \in R$ we denote by $\|v\|$ its Euclidean norm (as a vector). We define a multiplicative expansion factor $\gamma_\times(R)$ for the ring R by $\gamma_\times(R) = \max_{u,v \in R} \frac{\|u \times v\|}{\|u\| \cdot \|v\|}$. A typical choice is $f = x^n + 1$ with n a power of 2, for which $\gamma_\times(R) = \sqrt{n}$ (see [11, Th. 9]).

We say that two ideals I and J of R are coprime if $I + J = R$, where $I + J = \{i + j : i \in I, j \in J\}$. An ideal I is said prime of degree 1 if $\det(I)$ is prime. For an ideal J of R, we define $J^{-1} = \{\boldsymbol{v} \in \mathbb{Q}[x]/f : \forall \boldsymbol{u} \in J, \boldsymbol{u} \times \boldsymbol{v} \in R\}$. This is an ideal of the fraction field $\mathbb{Q}[x]/f$ of R, and it is included in $\frac{1}{\det J} R$ (since $(\det J) \cdot R \subseteq J$). If $f = x^n + 1$ with n a power of 2, then R is the ring of integers of the $(2n)$th cyclotomic field and $J^{-1} \times J = R$ for any integral ideal J (the product of two ideals being defined similarly to the sum). An ideal I is said principal if it is generated by a single element $r \in I$, and then we write $I = (r)$. We define $\mathrm{rot}_f(r) \in \mathbb{Q}^{n \times n}$ as the basis of I consisting of the $x^k r(x) \bmod f$'s, for $k \in [0, n-1]$.

If I is an ideal lattice for $f = x^n + 1$, then we have $\lambda_1(I) \geq \det(I)^{1/n}$: an easy way to prove it is to notice that the rotations $x^k v$ of any shortest non-zero vector v form a basis of a full-rank sublattice of I, and to use the inequalities $\lambda_1(I)^n = \prod_k \|x^k v\| \geq \det((v)) \geq \det I$.

2.3 Homomorphic encryption

In this section, we review definitions related to homomorphic encryption. Our definitions are based on [11, 10], but we slightly relax the definition of decryption correctness, to allow a negligible probability of error. This is essential for our probabilistic improvement to Gentry's `Recrypt` algorithm.

Definition 2.1 (Homomorphic Encryption). *A homomorphic encryption scheme* Hom *consists of four algorithms:*

- KeyGen*: Given security parameter λ, outputs a secret key sk and public key pk.*
- Enc*: Given plaintext $\pi \in \{0, 1\}$ and public key pk, returns ciphertext ψ.*
- Dec*: Given ciphertext ψ and secret key sk, returns plaintext π.*
- Eval*: Given public key pk, a t-input circuit C (consisting of addition and multiplication gates modulo 2), and a tuple of ciphertexts $(\psi_1, \ldots, \psi_t)$ (corresponding to the t input bits of C), returns a ciphertext ψ (corresponding to the output bit of C).*

The scheme Hom *is* correct *for a family of circuits $\mathcal{C}$ taking at most $t = \mathcal{P}oly(\lambda)$ input bits if for any $C \in \mathcal{C}$ and for any input bits $\pi_1, \ldots, \pi_t$, the following holds with overwhelming probability over the randomness of* KeyGen *and* Enc*:*

$$\mathsf{Dec}(sk, \mathsf{Eval}(pk, C, (\psi_1, \ldots, \psi_t))) = C(\pi_1, \ldots, \pi_t),$$

where $(sk, pk) = \mathsf{KeyGen}(\lambda)$ and $\psi_i = \mathsf{Enc}(pk, \pi_i)$ for $i = 1, \ldots, t$.

The scheme Hom *is* compact *if for any circuit C with at most $t = \mathcal{P}oly(\lambda)$ input bits, the size of the ciphertext $\mathsf{Eval}(pk, C, (\psi_1, \ldots, \psi_t)))$ is bounded by a fixed polynomial $b(\lambda)$.*

Gentry [11] defined the powerful notion of a *bootstrappable* homomorphic encryption scheme: one that can homomorphically evaluate a decryption of two ciphertexts followed by one gate applied to the decrypted values. We again relax this notion to allow decryption errors.

Definition 2.2 (Bootstrappable Homomorphic Encryption). *Let* $\mathsf{Hom} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *denote a homomorphic encryption scheme. We define two circuits:*

- $\mathsf{Dec}-\mathsf{Add}$*: Takes as inputs a secret key sk and two ciphertexts ψ_1, ψ_2, and computes* $\mathsf{Dec}(sk, \psi_1) + \mathsf{Dec}(sk, \psi_2) \bmod 2$.
- $\mathsf{Dec}-\mathsf{Mult}$*: Takes as inputs a secret key sk and two ciphertexts ψ_1, ψ_2, and computes* $\mathsf{Dec}(sk, \psi_1) \times \mathsf{Dec}(sk, \psi_2) \bmod 2$.

We say that Hom *is bootstrappable if it is correct for $\mathcal{C} = \{\mathsf{Dec}-\mathsf{Add}, \mathsf{Dec}-\mathsf{Mult}\}$.*

Gentry discovered that a bootstrappable homomorphic encryption can be used to homomorphically evaluate arbitrary circuits. More precisely, he proved the following result (adapted to allow for decryption error).

Theorem 2.2 ([11, Se. 2]). *Given a bootstrappable homomorphic encryption scheme* Hom*, and parameters $d = \mathcal{P}oly(\lambda)$, it is possible to construct another homomorphic encryption scheme $\mathsf{Hom}^{(d)}$ that is compact and correct for all circuits of size $\mathcal{P}oly(\lambda)$. Furthermore, if the scheme* Hom *is semantically secure, then so is the scheme $\mathsf{Hom}^{(d)}$.*

The main idea of the transformation of Theorem 2.2 is as follows. The scheme $\mathsf{Hom}^{(d)}$ associates independent key pairs (sk_i, pk_i) (for $i \leq d$) of scheme Hom, one for each of the d levels of circuit C. The secret key for Hom^d is $(sk_1, \ldots, sk_d)$ and the public key is $(pk_1, \ldots, pk_d)$ along with $(\overline{sk}_{1,2}, \ldots, \overline{sk}_{d-1,d})$, where $\overline{sk}_{i,i+1}$ denotes a tuple of ℓ ciphertexts for the ℓ bits of secret key sk_i encrypted under pk_{i+1}. The $\mathsf{Eval}^{(d)}$ algorithm for Hom^d then works as follows. The ciphertexts for the bits of C at level i are encrypted with pk_i (with level 1 corresponding to the inputs). Given level i ciphertexts $\psi_{i,1}, \psi_{i,2}$, that we assume decrypt under sk_i to bit values $\pi_1 = \mathsf{Dec}(sk_i, \psi_{i,1})$ and $\pi_2 = \mathsf{Dec}(sk_i, \psi_{i,2})$, and are given as inputs to a multiply (resp. add) gate mod 2, algorithm $\mathsf{Eval}^{(d)}$ computes a level $i+1$ ciphertext ψ_{i+1} for the gate output value $\pi = \pi_1 \times \pi_2 \bmod 2$ as follows: It first individually encrypts the bits of $\psi_{i,1}$ and $\psi_{i,2}$ under pk_{i+1} to get a tuple of bit ciphertexts $\overline{\psi}_{i,1}$ and $\overline{\psi}_{i,2}$ (at this stage, the plaintexts are twice encrypted); then it inputs all the pk_{i+1}-encrypted ciphertexts $(\overline{sk}_{i,i+1}, \overline{\psi}_{i,1}, \overline{\psi}_{i,2})$ to the Eval algorithm of Hom with public key pk_{i+1} and circuit $\mathsf{Dec}-\mathsf{Mult}$; hence, by the bootstrappability of Hom, except for negligible probability (over $(\mathrm{sk}_{i+1}, pk_{i+1})$ and the randomness used to compute the pk_{i+1}-encrypted ciphertexts $\overline{\psi}_{i,1}$ and $\overline{\psi}_{i,2}$), the resulting ciphertext ψ_{i+1} decrypts to $\mathsf{Dec}(sk_{i+1}, \psi_{i+1}) = \mathsf{Dec}(sk_i, \psi_1) \times \mathsf{Dec}(sk_i, \psi_2) \bmod 2$, as required. By a union bound over all gates in the circuit C, we see that $\mathsf{Hom}^{(d)}$ is correct for all circuits of depth at most d.

Note that the above error probability analysis uses the fact that the bits of the ciphertexts $\psi_{i,1}$ and $\psi_{i,2}$ are independent of (sk_{i+1}, pk_{i+1}). Gentry also described in [10, Se. 4.3] a variant where all d levels use the same key pair: the above probabilistic argument does not carry over to this situation, but we circumvent this issue in Section 6.

3 Summary of Gentry's Fully Homomorphic Scheme

In this section, we review Gentry's fully homomorphic encryption scheme [11, 10].

3.1 The somewhat homomorphic scheme

We first recall Gentry's somewhat homomorphic encryption scheme (see [10, Se. 5.2 and Ch. 7]) which supports a limited number of multiplications. It is the basis for the bootstrappable scheme presented later. The somewhat homomorphic scheme, described in Figure 1, produces ciphertexts in the ring $R = \mathbb{Z}[x]/f$ for a suitable irreducible degree n monic polynomial f. In this paper, we will assume $f = x^n + 1$ with n a power of 2. Here n is a function of the security parameter λ.

The key generation procedure generates two coprime ideals I and J of R. The ideal I has basis B_I. To simplify the scheme (and optimize its efficiency), a convenient choice, which we assume in this paper, is to take $I = (2)$: Reduction of v modulo I corresponds to reducing the coefficients of the vector/polynomial v modulo 2. The ideal J is generated by an algorithm $\mathsf{IdealGen}$, that given (λ, n), generates a 'good' secret basis B_J^{sk} (consisting of short, nearly orthogonal vectors) and computes its HNF to obtain a 'bad' public basis B_J^{pk}. Suggestions for concrete implementations of $\mathsf{IdealGen}$ are given in [10, Se. 7.6], [10, Ch. 18] and [27]. To obtain our $\widetilde{O}(\lambda^3)$ bit complexity bound, we will assume that J is a degree 1 prime ideal, which is the case with the implementation of [27] and can be obtained by rejection from the distribution considered in [10, Ch. 18]. The latter rejection method can be shown efficient by using Chebotarev's density theorem (see, e.g., [17]). Associated with $\mathsf{IdealGen}$ is a parameter r_{Dec}, which is a lower bound on the radius of the largest origin-centered ball which is contained inside $\mathcal{P}(B_J^{sk})$. In all cases we have $r_{Dec} \geq \lambda_1(J)/\mathcal{P}oly(n)$ (see, e.g., [10, Le. 7.6.2]). Using Babai's rounding-off algorithm [4] with B_J^{sk}, the decryptor can recover the point of J closest to any target vector within distance r_{Dec} of J (see [10, Le. 7.6.1]).

The plaintext space is a subset of $\mathcal{P}(I)$, that we assume to be $\{0, 1\}$. The encryption algorithm uses a sampling algorithm Samp, which given $(B_I, \boldsymbol{x})$ for a vector $x \in R$, samples a 'short' vector in the coset $x + I$. Concrete implementations of Samp are given in [10, Se. 7.5 and 14.1]. Associated with Samp is a parameter r_{Enc}, which is a (possibly probabilistic) bound on the norms of vectors output by Samp. For both implementations, one can set $r_{Enc} = \mathcal{P}oly(n)$. To encrypt a message π, a sample $\pi + i$ from the coset $\pi + I$ is generated, and the result is reduced modulo the public basis B_J^{pk}: $\psi = \pi + i \bmod B_J^{pk}$. It is assumed that $r_{Enc} < r_{Dec}$. Therefore, by reducing ψ modulo the secret basis B_J^{sk} one can recover $\pi + i$, and then plaintext π can be recovered by reducing modulo B_I.

Homomorphic addition and multiplication of the encrypted plaintexts π_1, π_2 modulo B_I are supported by performing addition and multiplication respectively in the ring R on the corresponding ciphertexts modulo B_J^{pk}. Namely, for $\psi_1 = \pi_1 + i_1 \bmod B_J^{pk}$, $\psi_2 = \pi_2 + i_2 \bmod B_J^{pk}$ with $i_1, i_2 \in I$, we have $\psi_1 + \psi_2 \bmod B_J^{pk} \in (\pi_1 + \pi_2) + I$ and $\psi_1 \times \psi_2 \bmod B_J^{pk} \in (\pi_1 \times \pi_2) + I \bmod B_J^{pk}$. However, for correct decryption of these new ciphertexts, we need that $\|(\pi_1 + i_1) + (\pi_2 + i_2)\|$

and $\|(\pi_1 + i_1) \times (\pi_2 + i_2)\|$ are not larger than r_{Dec}. This limits the degree of polynomials that can be evaluated homomorphically. Note that our choice for J implies that a ciphertext reduced modulo B_J^{pk} is simply an integer modulo $\det(J)$ and thus homomorphic evaluations modulo B_J^{pk} reduces to integer arithmetic modulo $\det(J)$ (such as in [27]).

- $\mathsf{KeyGen}(\lambda)$: Run $\mathsf{IdealGen}(\lambda, n)$ to generate private/public bases (B_J^{sk}, B_J^{pk}) for ideal J such that $\mathcal{P}(B_J^{sk})$ contains an origin-centered ball of radius $r_{Dec} \approx \lambda_1(J)$. Return public key $pk = B_J^{pk}$ and secret key $sk = B_J^{sk}$.
- $\mathsf{Enc}(pk, \pi)$: Given plaintext $\pi \in \{0, 1\}$ and public key pk, run $\mathsf{Samp}(I, \pi)$ to get $\pi' \in \pi + I$ with $\|\pi'\| \le r_{Enc}$. Return ciphertext $\psi = \pi' \bmod B_J^{pk}$.
- $\mathsf{Dec}(sk, \psi)$: Given ciphertext ψ and secret key sk, returns $\pi = (\psi \bmod B_J^{sk}) \bmod I$.
- $\mathsf{Eval}(pk, C, (\psi_1, \ldots, \psi_t))$: Given public key pk, circuit C and ciphertexts $\psi_1, \ldots, \psi_t$, for each add or multiply gate in C, perform a $+$ or $\times$ operation in $R \bmod B_J^{pk}$, respectively, on the corresponding ciphertexts. Return the ciphertext ψ corresponding to the output of C.

Fig. 1. Gentry's Somewhat Homomorphic Encryption Scheme SomHom.

3.2 A tweaked somewhat homomorphic scheme

Gentry [10, Ch. 8] introduced tweaks to SomHom to simplify the decryption algorithm towards constructing a fully homomorphic scheme. The tweaked scheme SomHom′ differs from the original scheme in the key generation and decryption algorithm, as detailed in Figure 2.

- $\mathsf{KeyGen}'(\lambda)$: Run $\mathsf{KeyGen}(\lambda)$ to obtain (B_J^{sk}, B_J^{pk}). From B_J^{sk}, compute a vector $\boldsymbol{v}_J^{sk} \in J^{-1}$ such that $\mathcal{P}(\mathrm{rot}_f(\boldsymbol{v}_J^{sk})^{-1})$ contains a ball of radius $r'_{Dec} = r_{Dec}/(8\sqrt{2}n^{2.5})$ (see [10, Le. 8.3.1]). Return public key $pk = B_J^{pk}$ and secret key $sk = B_J^{sk}$.
- $\mathsf{Dec}'(sk, \psi)$: Given ciphertext ψ and secret key sk, returns $\pi = \psi - \lfloor \boldsymbol{v}_J^{sk} \times \psi \rceil \bmod I$.

Fig. 2. Algorithms of the Tweaked Somewhat Homomorphic Encryption Scheme SomHom′ that differ from SomHom.

Gentry shows the following about the correctness of the tweaked decryption scheme.

Lemma 3.1 (Adapted from [10, Le. 8.3.1 and 8.4.2]). *A ciphertext $\psi = \pi + i \bmod B_J^{pk}$ with $\|\pi + i\| \le r'_{Dec}$ is correctly decrypted to π by* Dec′. *Moreover, if $\|\pi + i\| \le r'_{Dec}$, then each coefficient of $\boldsymbol{v}_J^{sk} \times \psi$ is within $1/8$ of an integer.*

Let C be a mod 2 circuit consisting of add and multiply gates with two inputs and one output. We let $g(C)$ denote the generalized circuit obtained from C by replacing the add and multiply gates mod 2 by the $+$ and $\times$ operations of the ring R, respectively. We say that circuit C is *permitted*, if for any set of inputs $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t$ to $g(C)$ with $\|\boldsymbol{x}_k\| \le r_{Enc}$ for $k = 1, \ldots, t$, we have $\|g(C)(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t)\| \le r'_{Dec}$. A permitted circuit which is evaluated homomorphically on encryptions of plaintexts $\pi_1, \ldots, \pi_t$ will yield a ciphertext $\psi = g(C)(\pi_1 + i_1, \ldots, \pi_t + i_t) \bmod B_J^{pk}$ that correctly decrypts to $C(\pi_1, \ldots, \pi_t)$, and such that the coefficients of $\boldsymbol{v}_J^{sk} \times \psi$ are within $1/8$ of an integer. As in [7, Le 3.4], we characterize the permitted circuits by the maximal degree of the polynomial evaluated by the circuit. Note that Gentry [11, 10] considers the circuit depth, which is less flexible.

Lemma 3.2. *Let C denote a mod 2 circuit, and let $g(C)$ denote corresponding generalized circuit over R, evaluating a polynomial $h(x_1, \ldots, x_t) \in \mathbb{Z}[x_1, \ldots, x_t]$ of (total) degree d. Then the circuit C is permitted if $\gamma_\times^{d-1} \|h\|_1 r_{Enc}^d \le r'_{Dec}$. In particular, assuming that h has coefficients in $\{0, 1\}$, the circuit C is permitted if d satisfies*

$$d \le \frac{\log r'_{Dec}}{\log(r_{Enc} \cdot \gamma_\times \cdot (t+1))}.$$

Proof. As observed above, circuit C is permitted as long as $\|g(C)(\pi_1 + i_1, \ldots, \pi_t + i_t)\| \leq r'_{Dec}$ whenever $\|\pi_k + i_k\| \leq r_{Enc}$ for $k = 1, \ldots, t$. Since $g(C)$ evaluates a polynomial h, and the norm of each term in h is upper bounded by $\gamma_\times^{d-1} r_{Enc}^d$, the triangle inequality implies that $\|g(C)(\pi_1 + i_1, \ldots, \pi_t + i_t)\| \leq \gamma_\times^{d-1} \|h\|_1 r_{Enc}^d$, as claimed. The bound on d follows from the fact that $\|h\|_1 \leq (t+1)^d$ since h has $\{0, 1\}$ coefficients and degree d and thus at most $(t+1)^d$ non-zero monomials. □

Remark. The polynomial h referred to above is the one evaluated by the *generalized* circuit $g(C)$. For arbitrary circuits C mod 2, the polynomial h may differ from the polynomial h' evaluated by the circuit C mod 2; in particular, the polynomial h may have non-binary integer coefficients, and some may be multiples of 2. However, for circuits C for which h has binary coefficients (the condition in the lemma), we have $h = h'$ (this condition on h is also needed, but is not explicitly stated in [7]).

3.3 Gentry's squashed bootstrappable scheme

To make it bootstrappable, Gentry [10, Ch. 10] modified SomHom' by 'squashing' the decryption circuit, i.e., moving some of the decryption computation to the encryption stage, by providing additional information in the public key. The modifications to SomHom' result in the squashed bootstrappable scheme SqHom described in Figure 3. The scheme introduces three new integer parameters $(p, \gamma_{set}, \gamma_{sub})$. Note that we incorporated Optimization 2 from [10, Ch. 12], which is made possible thanks to the choice $I = (2)$.

- $\mathsf{KeyGen}''(\lambda)$:
  - Run KeyGen' to get B_J^{pk} and $\boldsymbol{v}_J^{sk}$.
  - Generate a uniform γ_{set}-bit vector $\boldsymbol{s} = (s_1, \ldots, s_{\gamma_{set}})$ with Hamming weight γ_{sub} and $s_{\gamma_{set}} = 1$.
  - Generate $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_{\gamma_{set}-1}$ uniformly and independently from J^{-1} mod B_I. Compute $\boldsymbol{t}_{\gamma_{set}} = \boldsymbol{v}_J^{sk} - \sum_{k<\gamma_{set}} s_k \boldsymbol{t}_k$.
  - Return $sk = \boldsymbol{s}$ and $pk = (B_J^{pk}; \boldsymbol{t}_1, \ldots, \boldsymbol{t}_{\gamma_{set}})$.
- $\mathsf{Enc}''(pk, \pi)$: Run Enc of SomHom' to generate ciphertext ψ. For $k = 1, \ldots, \gamma_{set}$, compute c_k on $p+1$ bits (1 bit before the binary point, and p bits after) such that $|c_k - [\boldsymbol{t}_k \times \boldsymbol{\psi}]_0 \bmod 2| \leq 2^{-p}$, where $[g]_0$ denotes the constant coefficient of the polynomial $g \in R$. Return ciphertext $(\psi; c_1, \ldots, c_{\gamma_{set}})$.
- $\mathsf{Dec}''(sk, (\psi; c_1, \ldots, c_{\gamma_{set}}))$: Given expanded ciphertext $(\psi; c_1, \ldots, c_{\gamma_{set}})$ and secret key sk, return $\pi = [\psi]_0 - \lfloor \sum_k s_k c_k \rceil \bmod 2$.
- Eval'': Same as for SomHom' (while recomputing the c_k's, like in algorithm Enc'').

Fig. 3. Algorithms of the Squashed Scheme SqHom.

Note that $\sum_k s_k c_k \approx \sum_k s_k [\boldsymbol{t}_k \times \boldsymbol{\psi}]_0 \bmod 2 = ([(\sum_k s_k \boldsymbol{t}_k) \times \boldsymbol{\psi}]_0) \bmod 2 = [\boldsymbol{v}_J^{sk} \times \boldsymbol{\psi}]_0 \bmod 2$. Hence, in terms of decryption correctness, SqHom differs from SomHom' only due to the rounding errors. The following lemma provides a sufficient precision p (see also [7, Le. 6.1]). In Section 5, we will show that the precision p may be almost halved, using a probabilistic error analysis.

Lemma 3.3 (Adapted from [7, Le. 6.1]). *If $p \geq 3 + \log_2 \gamma_{sub}$, a ciphertext $(\psi; c_1, \ldots, c_{\gamma_{set}})$ of* SqHom *with $\psi = \pi + i \bmod B_J^{pk}$ and $\|\pi + i\| \leq r'_{Dec}$ is correctly decrypted by the decryption algorithm* Dec''*, and $\sum_k s_k c_k$ is within $1/4$ of an integer.*

Proof. We know by Lemma 3.1 that when $\|\pi + i\| \leq r'_{Dec}$, then $\pi = \psi - \lfloor \boldsymbol{v}_J^{sk} \times \psi \rceil \bmod 2$ and each coefficient of $\boldsymbol{v}_J^{sk} \times \psi$ is within $1/8$ of an integer. Hence it suffices to show that $|[\boldsymbol{v}_J^{sk} \times \psi]_0 - \sum_k s_k c_k \bmod 2| \leq 1/8$. Since $c_k = [\boldsymbol{t}_k \times \psi]_0 + \Delta_k \bmod 2$ with $|\Delta_k| \leq 2^{-p}$ for $k \leq \gamma_{set}$, we have:

$$\left| [\boldsymbol{v}_J^{sk} \times \psi]_0 - \sum_k s_k c_k \bmod 2 \right| \leq \left| [\boldsymbol{v}_J^{sk} \times \psi]_0 - \sum_k s_k \cdot [\boldsymbol{t}_k \times \psi]_0 \bmod 2 \right| + \left| \sum_k s_k \Delta_k \right|$$

$$\leq \left| [\boldsymbol{v}_J^{sk} \times \psi]_0 - [\boldsymbol{v}_J^{sk} \times \psi]_0 \bmod 2 \right| + \sum_k s_k \left| \Delta_k \right|$$
$$\leq \gamma_{sub} \cdot 2^{-p}.$$

The condition on p provides the result. □

For bootstrappability, we need the augmented decryption circuits $\mathsf{Dec} - \mathsf{Mult}$ and $\mathsf{Dec} - \mathsf{Add}$ to be implementable by a circuit with degree d' less than the degree capacity of the scheme. This is summarized in the following, in terms of the size γ_{sub} of the hidden subset in the secret key.

Theorem 3.1 (Adapted from [7, Th. 6.2]). *Assuming that $\sum_k s_k c_k$ is within $1/4$ of an integer, the augmented decryption circuits $\mathsf{Dec} - \mathsf{Mult}$ and $\mathsf{Dec} - \mathsf{Add}$ for scheme SqHom with precision parameter p can be evaluated by a circuit of degree $d' \leq \gamma_{sub} \cdot 2^9 p^{1.71}$.*

Proof. To decrypt ψ, we have to compute $\pi = [\psi]_0 - \lfloor \sum_k s_k c_k \rceil \bmod 2$. We proceed as follows:

1- Compute $a_k = s_k \cdot c_k$ for $k = 1, \ldots, \gamma_{set}$.

2- Let $a_{k,0}.a_{k,1} \ldots a_{k,p}$ denote the binary representation of a_k. To sum the a_k's:

  2.1- For $j = 0, \ldots, p$, compute W_j, the Hamming weight of the bit vector $(a_{0,j}, \ldots, a_{\gamma_{set},j})$.

  2.2- Compute $\pi = [\psi]_0 - \sum_{j \leq p} W_j \cdot 2^{-j} \bmod 2$.

Note that because only γ_{sub} of the a_k's are non-zero, each Hamming weight W_j is at most γ_{sub} and hence its binary representation has at most $\lceil \log_2(\gamma_{sub} + 1) \rceil$ bits. Step 1 requires only a single multiplication mod 2 for each output bit, hence has degree 2. For Step 2.1, we use the following.

Lemma 3.4 (Adapted from [7, Le. 6.3]). *Let $(\sigma_1, \ldots, \sigma_t)$ be a binary vector, and $W = W_n \ldots W_0$ be the binary representation of its Hamming weight. Then for any k, the bit W_k can be expressed as a the evaluation in the σ_j's of an integer polynomial of degree exactly 2^k.*

We conclude that Step 2.1 can be computed by a circuit of degree $2^{\lceil \log_2(\gamma_{sub}+1) \rceil} \leq 2\gamma_{sub}$. Using the 'three-for-two' trick [15], van Dijk et al. [7] show that Step 2.2 can be done with a circuit of degree $\leq 2^{\lceil \log_{3/2}(p+1) \rceil + 4} \leq 2^6 p^{1.71}$. The total degree of the decryption circuit is therefore $\leq \gamma_{sub} \cdot 2^8 p^{1.71}$, and hence that of $\mathsf{Dec} - \mathsf{Mult}$ (resp. $\mathsf{Dec} - \mathsf{Add}$) is at most $\gamma_{sub} \cdot 2^9 p^{1.71}$. □

Combining Theorem 3.1 with Lemmata 3.2 and 3.3, we get:

Corollary 3.1. *If $p = \lceil 3 + \log_2 \gamma_{sub} \rceil$, the scheme SqHom is bootstrappable as long as*

$$\gamma_{sub} \cdot 2^9 \log^{1.71}(\gamma_{sub} + 4) \leq \frac{\log r'_{Dec}}{\log(r_{Enc} \cdot \gamma_\times \cdot (t+1))}.$$

In this article, we are concerned with the total bit-complexity of refreshing a ciphertext (`Recrypt`) and homomorphically evaluating an elementary gate, as this is the most important step in the fully homomorphic scheme derived via Theorem 2.2. This consists in: expanding a ciphertext (i.e., the second part of Enc''); re-encrypting the bits of the expanded ciphertext under a new public key (with Enc); homomorphically evaluating Dec; and homomorphically evaluating either Add or Mult.

4 A Less Pessimistic Hardness Analysis of the SSSP

The semantic (CPA) security of Gentry's somewhat homomorphic schemes SomHom and SomHom' relies on the hardness of a bounded distance decoding problem. As explained in Section 2, this hardness assumption is asymptotically well understood (with the lattice reduction 'rule of the thumb'). When converted into the bootstrappable scheme SqHom, another hardness assumption is added, namely that of the so-called `SplitKey` distinguishing problem. To be precise, a semantic attack against SqHom either leads to an efficient ideal lattice BDD algorithm or to an efficient algorithm for the `SplitKey` distinguishing problem (see [11, Th. 10]). In [11, Th. 11.1.3], the following Sparse Vector Subset Sum Problem (SVSSP) is shown to reduce to the `SplitKey` distinguishing problem.

Definition 4.1 (SVSSP$_{\gamma_{sub},\gamma_{set}}$). *Let γ_{sub} and γ_{set} be functions of the hardness parameter λ. Let J be as generated by* KeyGen*, and B_{IJ} be the HNF of ideal IJ. The decisional SVSSP is as follows: Distinguish between $(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_{\gamma_{set}})$ chosen uniformly in $R \cap \mathcal{P}(B_{IJ})$ and the same but conditioned on the existence of a vector $\boldsymbol{s} \in \{0,1\}^{\gamma_{set}}$ of Hamming weight γ_{sub} with $\sum_k s_k \boldsymbol{a}_k = 0 \bmod IJ$.*

For our choice $I = (2)$, we have $B_{IJ} = 2B_J^{pk}$, where B_J^{pk} is the HNF of J. This choice of I is important here, as otherwise, for some J's, the matrix B_{IJ} has all but one of its diagonal entries equal to 1, and the lattice to be defined below can be canonically embedded into a γ_{set}-dimensional lattice. In the following, we use $q = \det(B_{IJ}) = 2^n \det(J)$. Note that the problem becomes interesting only when $\binom{\gamma_{set}}{\gamma_{sub}} = o(q)$. The exhaustive search algorithm runs in time $\binom{\gamma_{set}}{\gamma_{sub}}$, whereas a simple birthday paradox attack runs in time $\approx \binom{\gamma_{set}}{\gamma_{sub}}^{1/2}$. To achieve 2^λ hardness, we require that $\binom{\gamma_{set}}{\gamma_{sub}}^{1/2} \geq 2^\lambda$, or, more simply, that $\gamma_{sub} = \Omega(\lambda)$ and $\gamma_{sub} \leq \gamma_{set}/2$. We now consider a third attack, based on lattice reduction. Consider the lattice L defined by the columns of the following block matrix:

$$\left(\begin{array}{c|c} Id_{\gamma_{set}} & 0 \\ \hline A & B_{IJ} \end{array}\right),$$

where $Id_{\gamma_{set}}$ is the identity matrix and the columns of A are the $\boldsymbol{a}_k$'s. This lattice has dimension $\gamma_{set} + n$ and determinant q. Furthermore, the existence of the solution vector $\boldsymbol{s}$ implies that $\lambda_1(L) \in [1, \sqrt{\gamma_{sub}}]$, because the vector $\mathbf{s}' := (s_1, \ldots, s_{\gamma_{set}}, 0, \ldots, 0)^t$ belongs to L and $\|\boldsymbol{s}'\| = \sqrt{\gamma_{sub}}$.

Suppose we are limited to a computational power of 2^λ. The lattice reduction rule of the thumb suggests that we cannot find vectors in L of norms significantly smaller than $U := c^{\frac{\gamma_{set}+n}{\lambda}}$. There are $\leq m := U = 2^{\frac{\gamma_{set}+n}{\lambda}}$ pairs of non-zero multiples $\pm k \cdot \mathbf{s}'$ of $\mathbf{s}'$ with norm $\leq U$. At the same time, Minkowski's theorem (Theorem 2.1) asserts that there are far more lattice vectors of norm $\leq U$.

Lemma 4.1. *Assuming that $\frac{\pi^{\frac{\gamma_{set}+n}{2}}}{\Gamma\left(\frac{\gamma_{set}+n+2}{2}\right)} \cdot U^{\gamma_{set}+n} \geq (2^\lambda m) \cdot 2^{\gamma_{set}+n} \cdot q$, we have $|L \cap \mathcal{B}(\mathbf{0}, U)| \geq 2^\lambda \cdot m$.*

It is reasonable to assume that the lattice points that are not multiples of $\mathbf{s}'$ do not provide information towards solving SVSSP. Also, there is no reason to expect lattice reduction to find one of these relevant vectors rather than any lattice vector of norm $\leq U$. Under these assumptions, if the computational effort of lattice reduction is limited to 2^λ and if we wish to bound the likeliness of finding a relevant vector by $2^{-\lambda}$, it suffices to set the parameters so that:

$$c^{\frac{(\gamma_{set}+n)^2}{\lambda}} \geq 2^\lambda \cdot (\gamma_{set} + n)^{\Omega(\gamma_{set}+n)} \cdot q.$$

As $\gamma_{set}, n = \Omega(\lambda)$, the above is implied by $\frac{(\gamma_{set}+n)^2}{\lambda} = \widetilde{\Omega}(\gamma_{set} + \log q)$. Also, we will always have $n + \gamma_{set} = \widetilde{\Omega}(\lambda)$, so that the condition can be simplified into $\frac{(\gamma_{set}+n)^2}{\lambda} = \widetilde{\Omega}(\log q)$. Note that this condition is less restrictive than the corresponding one used in [11, 27, 7] (i.e., $\gamma_{set} = \Omega(\log q)$).

Remark. In our variant of algorithm KeyGen″, our instances of SVSSP always satisfy $s_{\gamma_{set}} = 1$. This does not result in any security reduction, as an attacker can always efficiently guess an i such that $s_i = 1$ and then permute indices i and γ_{set}.

Remark. Our analysis differs in three ways from the one from [11] relying on [24]: for consistency with the hardness analysis of the ideal BDD, we consider an approximate SVP solver rather than an exact SVP solver; furthermore, we do not consider the 'replay' attack from [24] (which would lead to larger $O(\cdot)$ constants), as contrarily to the case of server-aided RSA, only one instance of the SSSP is given; finally, the $\boldsymbol{a}_i$'s are vectors rather than integers, because the HNF matrix $B_{IJ} = 2B_J^{pk}$ has no trivial diagonal coefficients (i.e., diagonal coefficients equal to 1).

5 Improved Ciphertext Refreshing Algorithm

The main component in the degree of the decryption algorithm comes from (as explained in the proof of Theorem 3.1) the addition of the rationals $s_k c_k = [s_k \boldsymbol{t}_k \times \boldsymbol{\psi}]_0 \bmod 2$. This accounts for degree γ_{sub}, and all other components of degree are negligible compared to this one.

Recall that $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_{\gamma_{set}-1}$, and hence also $[\boldsymbol{t}_1 \times \boldsymbol{\psi}]_0 \bmod 2, \ldots, [\boldsymbol{t}_{\gamma_{set}-1} \times \boldsymbol{\psi}]_0 \bmod 2$'s are chosen *independently with identical distribution* (iid), and that $\boldsymbol{t}_{\gamma_{set}} = \boldsymbol{v}_J^{sk} - \sum_{k<\gamma_{set}} s_k \boldsymbol{t}_k \bmod 2$. We are to exploit the iid-ness of the first $\boldsymbol{t}_i$'s to obtain a sufficient precision p that is essentially half of that of Section 3.3. This will have the effect of taking the square root of the decryption circuit degree.

5.1 Using less precision

We start by summing the $s_k[\boldsymbol{t}_k \times \boldsymbol{\psi}]_0$'s for $k < \gamma_{set}$, since they are iid, and then we add the remaining $c_{\gamma_{set}}$. The first sum will be represented on 6 bits (1 bit before the point, and 5 bits after) and we will ensure that it is within 1/16 of $\sum_{k<\gamma_{set}} s_k[\boldsymbol{t}_k \times \boldsymbol{\psi}]_0 \bmod 2$, with high probability. We take $c_{\gamma_{set}}$ within distance 1/16 of $[\boldsymbol{t}_{\gamma_{set}-1} \times \boldsymbol{\psi}]_0 \bmod 2$ and represent it also on 6 bits. The last sum will provide a result within distance 1/8 of $\sum_{k\leq\gamma_{set}} s_k[\boldsymbol{t}_k \times \boldsymbol{\psi}]_0 \bmod 2$, and can be done with a circuit of constant degree. Using Lemma 3.1, we obtain that the result is within 1/4 of an integer.

We now concentrate on the first sum. Let the c_k's be fixed-point approximations to the $[\boldsymbol{t}_k \times \boldsymbol{\psi}]_0$'s, with some precision p. We have $\varepsilon_k \leq 2^{-p}$ with $\varepsilon_k = c_k - [\boldsymbol{t}_k \times \boldsymbol{\psi}]_0$. As the c_k's for $k < \gamma_{set}$ are independent, so are the corresponding ε_k's. Furthermore, we will ensure that $E[\varepsilon_k] = 0$ for any $k < \gamma_{set}$. The following lemma leads to a probabilistic error bound for the sum of the c_k's.

Lemma 5.1. *Let $\varepsilon_1, \ldots, \varepsilon_t$ be iid variables with values in $[-\varepsilon, \varepsilon]$ and such that $E[\varepsilon_k] = 0$ for all k. Then $|\sum_{k\leq t} \varepsilon_k| \leq \sqrt{t}\varepsilon \cdot \omega(\sqrt{\log \lambda})$ with probability negligibly small with respect to λ.*

Proof. We apply Hoeffding's inequality to the ε_i's. We have that $\Pr[|\sum \varepsilon_k| \geq x] \leq \exp\left(-\frac{x^2}{2\varepsilon^2}\right)$, for any $x > 0$. Taking $x = \sqrt{t}\varepsilon \cdot \omega(\sqrt{\log \lambda})$ leads to the result. □

We use this lemma with $\varepsilon = 2^{-p}$ and $t = \gamma_{sub} - 1$ (i.e., the number of non-zero $s_k\varepsilon_k$'s for $k < \gamma_{sub}$). It indicates that taking $p = \frac{1}{2}\log_2 \gamma_{sub} + \omega(\log\log\lambda)$ suffices to ensure that with probability negligibly close to 1 we have $|\sum_{k<\gamma_{set}} s_k(c_k - [\boldsymbol{t}_k \times \boldsymbol{\psi}]_0) \bmod 2| \leq 1/32$. Truncating the result to 5 bits after the binary point cannot add more than an error of 1/32.

5.2 Expliciting the computation of the c_k's in Enc''

In order to be able apply Lemma 5.1, we have to ensure that $E[\varepsilon_k] = 0$ for any $k < \gamma_{set}$. To guarantee that this is the case and that this computation enjoys a limited complexity bound, the computation of the c_k's needs to be implemented carefully.

We are given $\boldsymbol{t}_k$ and $\boldsymbol{\psi}$, and wish to compute a $(1+p)$-bit approximation c_k to $[\boldsymbol{t}_k \times \boldsymbol{\psi}]_0 \bmod 2$. As J is a degree 1 prime ideal, vector $\boldsymbol{\psi}$ is in fact an integer modulo $\det(J)$. We are thus interested in computing $[\boldsymbol{t}_k]_0 \cdot \psi$ modulo 2. We explicit this computation in Figure 4.

Inputs: Vectors $\boldsymbol{t}_k$ and $\boldsymbol{\psi}$, and precision p.
Output: A precision $(1+p)$ number $c_k \in [-1, 1]$ such that $|c_k - ([\boldsymbol{t}_k \times \boldsymbol{\psi}]_0 \bmod 2)| \leq 2^{-p}$.
1. $p' := \log_2 \det(J) + p + 1$;
2. Compute the closest precision $(1+p')$ number $\bar{t}_k \in [-1, 1]$ to $[\boldsymbol{t}_k]_0$.
3. Compute $c'_k := \bar{t}_k \psi$ exactly.
4. Reduce c'_k modulo 2, while preserving its sign (the result belongs to $[-1, 1]$).
5. Round c'_k to the closest precision $(1+p)$ number $c_k \in [-1, 1]$.

Fig. 4. Computing coefficient c_k for algorithm Enc''.

Lemma 5.2. *The algorithm of Figure 4 is correct. Furthermore, if the vector* $\boldsymbol{t}_k$ *is chosen uniformly in* $J^{-1} \bmod 2$ *with uniformly random choice of sign when a coordinate of* $\boldsymbol{t}_k$ *belongs to* $\{-1, 1\}$*, then* $E[\varepsilon_k] = 0$*, where* $\varepsilon_k = c_k - ([\boldsymbol{t}_k \times \boldsymbol{\psi}]_0 \bmod 2)$.

Proof. At Step 2 of the algorithm, we have $|\bar{t}_k - [\boldsymbol{t}_k]_0| \leq 2^{-p'-1}$. As ψ is exact and belongs to $[0, \det(J))$, we have $|\bar{t}_k\psi - [\boldsymbol{t}_k]_0\psi| \leq 2^{-p'-1}\det(J) \leq 2^{-p-1}$. Thus, at Step 3, we have $|c'_k - [\boldsymbol{t}_k \times \boldsymbol{\psi}]_0| \leq 2^{-p-1}$. The rounding of Step 5 leads to $|c_k - ([\boldsymbol{t}_k \times \boldsymbol{\psi}]_0 \bmod 2)| \leq 2^{-p-1} + 2^{-p-1} = 2^{-p}$.

To prove the second statement, we use the symmetry of the distribution of $\boldsymbol{t}_k$: a given sample $\boldsymbol{t}_k$ is as likely as its opposite $-\boldsymbol{t}_k$. This implies that $E[[\boldsymbol{t}_k \times \boldsymbol{\psi}]_0 \bmod 2] = 0$. We now use the same property to show that $E[c_k] = 0$. At Step 2, changing $\boldsymbol{t}_k$ into $-\boldsymbol{t}_k$ has the effect of changing $\bar{t}_k$ into $-\bar{t}_k$. This implies that at Step 3, changing $\boldsymbol{t}_k$ into $-\boldsymbol{t}_k$ has the effect of changing c'_k into $-c'_k$. Due to the symmetry of the rounding to nearest, this carries over to c_k and ε_k at Step 5. □

Note that the choice of rounding to nearest is not benign: the rounding mode needs to be symmetric with respect to 0 for the above proof to hold. Rounding downwards or upwards, two other standard rounding modes [1], would break the proof.

5.3 Decreasing the decryption circuit depth

We now want to compute $\sum_{k<\gamma_{set}} s_k c_k \bmod 2$, where the c_k's are fixed-point reals with precision $p = \frac{1}{2}\log_2 \gamma_{sub} + \omega(\log\log\lambda)$. Instead of computing the Hamming weights W_j for $j \in \{0, \ldots, p\}$ as in the proof of Theorem 3.1, we compute only the bits $W_{j,\ell}$ (for $0 \leq \ell \leq \lceil \log_2 \gamma_{sub} \rceil$) that are going to contribute to $\sum_{k<\gamma_{set}} s_k c_k \bmod 2$: the most significant bits are rendered useless by the reduction modulo 2. Most interestingly, these unnecessary most significant bits were the ones requiring the higher degree circuits to evaluate. More precisely, we have:

$$\sum_{k<\gamma_{set}} s_k c_k = \sum_{j=0}^{p} W_j 2^{-j} = \sum_{j=0}^{p} \sum_{\ell=0}^{\lceil \log_2 \gamma_{sub} \rceil} W_{j,\ell} 2^{-j+\ell} = \sum_{j=0}^{p} \sum_{\ell=0}^{j+1} W_{j,\ell} 2^{-j+\ell} \bmod 2.$$

Lemma 3.4 now implies that the desired sum mod 2 can be computed correctly with probability negligibly close to 1 with respect to λ, by evaluating an arithmetic circuit of size $\mathcal{P}oly(\gamma_{sub})$ corresponding to a polynomial of degree exactly $2^{p+1} = \sqrt{\gamma_{sub}} \cdot \omega(\sqrt{\log \lambda})$. This concludes the description of the improved decryption circuit. Overall, we get:

Theorem 5.1. *Assuming that $\sum_k s_k \cdot [\boldsymbol{t}_k \times \boldsymbol{\psi}]_0$ is within $1/8$ of an integer, the augmented decryption circuits* $\mathsf{Dec} - \mathsf{Mult}$ *and* $\mathsf{Dec} - \mathsf{Add}$ *for scheme* SqHom *with precision parameter $p' = \sqrt{\gamma_{sub}} \cdot \omega(\sqrt{\log \lambda})$ can be evaluated by a circuit of degree $d' \leq \sqrt{\gamma_{sub}} \cdot \omega(\sqrt{\log \lambda})$.*

Consequently, we have the following bootstrap condition.

Corollary 5.1. *If $p = \frac{1}{2} \log_2 \gamma_{sub} + \omega(\log \log \lambda)$, the scheme* SqHom *is bootstrappable as long as*

$$\sqrt{\gamma_{sub}} \cdot \omega(\sqrt{\log \lambda}) \leq \frac{\log r'_{Dec}}{\log(r_{Enc} \cdot \gamma_\times \cdot (t+1))}.$$

6 Asymptotic Efficiency

We now use the background results and the improvements described in the previous sections to derive bounds for the asymptotic complexity of Gentry's fully homomorphic scheme.

6.1 Optimizing the parameters in Gentry's fully homomorphic encryption

The table below summarizes and compares the conditions on the parameters $\gamma_{set}, \gamma_{sub}, n, \ldots$ for Gentry's fully homomorphic encryption scheme to be 2^λ-secure and correct.

The semantic security of the somewhat homomorphic scheme is related to the hardness of γ-BDD for $\gamma = r'_{Dec}/r_{Enc}$. More precisely, in the case of [11], the decisional variant of γ-BDD reduces to the semantic security. There is no known algorithm that performs better for the decisional variant of γ-BDD than for the computational variant. Recall that $r'_{Dec} = r_{Dec}/\mathcal{P}oly(n) = \lambda_1(J)/\mathcal{P}oly(n)$. Recall also that J is an ideal lattice, and thus we have $\lambda_1(J) \geq \det(J)^{1/n} = q^{1/n}/2$ (where q is the SVSSP determinant of Section 4). As a consequence, it suffices to ensure that γ-BDD is hard to solve for $\gamma = q^{1/n}/(r_{Enc}\mathcal{P}oly(n))$. We use the lattice reduction rule of the thumb to derive a sufficient condition.

As the encryptor is limited to polynomial-time algorithms, we can safely assume that $n = \mathcal{P}oly(\lambda)$. Also, since $f = x^n + 1$, we have $\gamma_\times = \sqrt{n}$. Finally, by choosing $r_{Enc} = \mathcal{P}oly(\lambda)$, the ciphertexts have sufficient entropy to prevent any exhaustive search.

Condition	[11]	This article
BDD resistant to lattice attacks	$\frac{q^{1/n}}{\mathcal{P}oly(\lambda)} \leq 2^{n/\lambda}$	
SSSP resistant to birthday paradox	$\binom{\gamma_{set}}{\gamma_{sub}}^{1/2} \geq 2^\lambda$	
SSSP resistant to lattice attacks	$\gamma_{set} = \widetilde{\Omega}(\log q)$	$\frac{(\gamma_{set}+n)^2}{\lambda} = \widetilde{\Omega}(\gamma_{set} + n + \log q)$
Bootstrappability achieved	$\gamma_{sub} \leq \frac{\log(q^{1/n})}{\Theta(\log \lambda)}$	$\sqrt{\gamma_{sub}} \leq \frac{\log(q^{1/n})}{\mathcal{P}oly(\log \lambda)}$

To fulfill these conditions, we choose $\gamma_{sub} = \Theta(\lambda)$ (for the second row). We can then choose $n = \widetilde{\Theta}(\lambda^{1.5})$ and $\log q = \widetilde{\Theta}(\lambda^2)$ (for the first and last rows). Finally, we set[5] $\gamma_{set} = \Theta(\lambda)$. These values may be compared to the corresponding values in [10, Ch. 12], namely $n \approx \lambda^2$, $\log q \approx \lambda^3$ and $\gamma_{set} \approx \lambda^3$.

6.2 Bit complexity

The `Recrypt` procedure consists in expanding the ciphertext ψ as described in algorithm Enc'' of SqHom, encrypting the bits of the expanded ciphertext with the new public key pk_2, and then applying algorithm Dec'' homomorphically, using the encrypted ciphertext bits and the encrypted secret key sk_1 (under pk_2). We also consider the cost of homomorphically evaluating an elementary add/mult gate.

Let us first bound the cost of computing the c_k's in Enc'', calling γ_{set} times the algorithm from Figure 4. First, note that Steps 1 and 2 should not be done within Enc'', but at the key generation time, i.e., in KeyGen''. Note that during the third step of KeyGen'', one should also pay attention to perform the reduction modulo (2) such that the assumption of Lemma 5.2 holds. The quantity c'_k obtained at Step 3 of the algorithm from Figure 4 is encoded on $O(\log q)$ bits, and its computation can be performed in $\widetilde{O}(\log q)$ bit operations, using fast integer arithmetic [26]. The costs of Steps 4 and 5 are negligible. Overall, the computation of the c_k's in Enc'' can be done in $\widetilde{O}(\gamma_{set} \log q) = \widetilde{O}(\lambda^3)$ bit operations.

The secret key is made of $\gamma_{set} = \Theta(\lambda)$ bits. The bit-length of the encrypted secret key is $\gamma_{set} \log q = \widetilde{O}(\lambda^3)$. To encrypt the bits of the c_k's under pk_2, we use $\mathsf{Samp} = 0$, as explained in [10, Re. 4.1.1], i.e., we consider as encrypted values the bits themselves.

Let us now explain how algorithm Dec'' is implemented. We concentrate on the most expensive part, i.e., the (homomorphic) computations of $O(\log \gamma_{sub}) = \widetilde{O}(1)$ Hamming weights of vectors in $\{0,1\}^{\gamma_{set}}$. Let $(\alpha_1, \ldots, \alpha_{\gamma_{set}})$ be such a vector. As explained in [11, Le. 5] (which relies on [5, Le. 11]), it suffices to compute the developed form of the polynomial $\prod_{k \le \gamma_{set}} (x - \alpha_k)$. Recall that in Section 5 we showed that we are interested in only a few coefficients of the result, corresponding to monomials of degrees $\widetilde{O}(\sqrt{\gamma_{sub}})$. We are to compute the full developed form anyway, and then throw away the spurious coefficients. Note that our circuit is over the integers, and evaluates an integer polynomial whose coefficients of interest have small multiplicative degrees in the inputs. We compute the developed form of $\prod_{k \le \gamma_{set}} (x - \alpha_k)$ with a binary tree:

- At level 0 of the tree, we have the linear factors $(x - \alpha_k)$.
- At level i of the tree, we have $\gamma_{set}/2^i$ polynomials of degree 2^i that are the products of the linear factors corresponding to their binary subtrees.
- A father of two nodes is obtained by multiplying his two sons, with a quasi-linear time multiplication for polynomials over rings that uses only ring operations (see, e.g., [9, Th. 8.23]).

The size of each circuit that allows to move from sons at level $i-1$ to father at level i is $\widetilde{O}(2^i)$. The overall number of add/mult integer gates is therefore $\widetilde{O}(\gamma_{set})$. While evaluating this circuit homomorphically, each gate corresponds to a multiplication/addition modulo B_J^{pk}, i.e., thanks to our choice for J, to a multiplication/addition of two integers modulo $\det(J)$, whose bit-length is $O(\log q)$. The overall complexity of Dec'' is $\widetilde{O}(\gamma_{set} \log q) = \widetilde{O}(\lambda^3)$.

[5] Note that we have $\gamma_{set}/\gamma_{sub} = O(1)$, versus $O(\lambda^2)$ for Gentry's choice of parameters [10, Ch. 12]. Thus, in our version of the SSSP, the hidden subset is not really 'sparse' anymore, which intuitively seems to be a weakening of the SSSP hardness assumption.

To summarize, the complexity of `Recrypt` for 1 bit of plaintext is $\widetilde{O}(\lambda^3)$ bit operations (compared to the complexity bound $\widetilde{O}(\lambda^6)$ claimed in [10, Ch. 12]). And the cost of homomorphically evaluating an elementary add/mult gate is also $\widetilde{O}(\lambda^3)$. The secret key is encoded on $\Theta(\lambda)$ bits and the public key is $(B_J^{pk}; \bar{\boldsymbol{t}}_1, \ldots, \bar{\boldsymbol{t}}_{\gamma_{set}})$ is encoded on $\widetilde{O}(n \log q + \gamma_{set} \log q) = \widetilde{O}(\lambda^{3.5})$ bits.

7 Other Fully Homomorphic Schemes

We now show how our improvements can be adapted to variants of Gentry's fully homomorphic scheme and to the scheme of van Dijk et al..

7.1 Smaller keys

In [10, Se. 4.3], Gentry suggests to re-use the same key-pair for all levels of the fully homomorphic scheme derived from Theorem 2.2. This allows one to significantly decrease the key-sizes of the boostrapped fully homomorphic scheme. This strategy can be proved secure if the underlying bootstrappable homomorphic encryption scheme is assumed/known KDM-secure [10, Th. 4.3.2]. Our lower-degree decryption may fail with non-negligible probability after the first refreshing of a ciphertext, as our technique does not handle the non-independence of the ciphertext and the secret key. To circumvent this issue, we randomize the ciphertext to waive its possible non-independence with the secret key. Note that this technique is similar in flavor to Gentry's modified scheme providing circuit privacy [11, Se. 7].

Consider algorithm Enc'' of SqHom. The condition required for the probabilistic technique described in Section 5 to work is that the ciphertext $\boldsymbol{\psi} = \pi + \boldsymbol{r} \bmod B_J^{pk}$ (where $\boldsymbol{r} \in (2)$ and $\|\boldsymbol{r}\| \leq r'_{Dec}$) is independent of the $\boldsymbol{t}_i$'s. This fact, together with the iid-ness of the $\boldsymbol{t}_i$'s, implies that the rounding errors ε_i in computing the c_i's, are iid, as required to apply Hoeffding's bound. In the key-reuse application, the internal randomness $\boldsymbol{r}$ of $\boldsymbol{\psi}$ may depend on the $\boldsymbol{t}_i$'s (due to a previous refreshing). To circumvent this, we randomize the ciphertext $\boldsymbol{\psi} = \pi + \boldsymbol{r} \bmod B_J^{pk}$ into another ciphertext $\boldsymbol{\psi'} = \pi + \boldsymbol{r'} \bmod B_J^{pk}$ for the same message π but with internal randomness $\boldsymbol{r'} \in (2)$ which is almost independent of the $\boldsymbol{t}_i$'s. More precisely, given the $\boldsymbol{t}_i$'s, the distribution of $\boldsymbol{r'}$ is within negligible statistical distance from the ($\boldsymbol{t}_i$-independent) distribution $2U$, where U is the uniform distribution on the origin-centered ball of radius r'_{Dec}/ρ with ρ any negligible function of λ such that $\log \rho = \widetilde{O}(1)$ (e.g., $\rho = \lambda^{-\log \lambda}$).

We compute $\boldsymbol{\psi}'$ by adding to $\boldsymbol{\psi}$ an encryption of 0 with sufficiently large randomness compared to the randomness in $\boldsymbol{\psi}$, i.e., we set $\boldsymbol{\psi}' = \boldsymbol{\psi} + \boldsymbol{\zeta} \bmod B_J^{pk}$, where $\boldsymbol{\zeta}$ is sampled from $2U$. If we replace the decryption radius r'_{Dec} by $r''_{Dec} = \frac{r'_{Dec}}{1+2/\rho}$ in Lemma 3.2, then the correctness of the scheme is preserved, as $\boldsymbol{\psi}$ and $\boldsymbol{\psi}'$ both decode to the same plaintext via algorithm Dec'. This has a negligible effect for the asymptotic efficiency (see Section 6.1). Assume that $\boldsymbol{\psi} = \pi + \boldsymbol{r} \bmod B_J^{pk}$ with $\|\boldsymbol{r}\| \leq r'_{Dec}$. Let us consider the statistical distance between the distributions $\boldsymbol{r} + 2U$ and $2U$. As a ball of radius $r'_{Dec}/\rho - r'_{Dec}$ is contained in the intersection of the two balls of radius r'_{Dec}/ρ corresponding to U and $\boldsymbol{r} + U$, we obtain that the statistical distance under scope is at most $n \cdot \rho$, and hence negligible.

7.2 Gentry's provably secure scheme

In his PhD thesis [10], Gentry describes an instantiation of the scheme from [11] that enjoys strong security proofs. First, in Chapter 11, the computational SSSP is shown to reduce to the `Splitkey` distinguishing problem, which is more natural than the decisional SVSSP we considered in Section 4, and had already been studied in other contexts, such as server-aided RSA [24]. Second, in Chapters 14–19, the somewhat homomorphic scheme is shown semantically secure under the worst-case assumption that solving variants of the Shortest Vector Problem for ideal lattices is hard. Two crucial assumptions for these security proofs to hold are that J is a prime ideal with prime determinant (such as in [27]), and that I is an ideal with prime determinant and small norm (polynomial in n). These requirements are not met by the choice $I = (2)$ that we assumed above, since $\det I = 2^n$.

A workaround consists in taking I to be a factor of the ideal (2) that satisfies the requirements. For instance in the case $f = x^n + 1$ with n a power of 2, one has $x^n + 1 = (x + 1)^n \bmod 2$, which means that $(2) = \mathfrak{p}^n$, where $\mathfrak{p} = \langle 2, x + 1\rangle$ is a prime ideal with $\det \mathfrak{p} = 2$. We still choose J to be a degree 1 ideal. In this context, the obtained SVSSP has dimension $\gamma_{set} + 1$ (instead of $\gamma_{set} + n$), because the trailing $n - 1$ diagonal coefficients of the HNF B_{IJ} are all 1 (see the proof of [10, Th. 11.1.5]). This leads to the following stronger condition for ensuring that the SSSP remains hard: $\frac{\gamma_{set}^2}{n} = \widetilde{\Omega}(\gamma_{set} + \log q)$. As the other conditions of Section 6.1 remain unchanged, the lowest value we can choose for γ_{set} becomes $\Theta(\lambda^{1.5})$.

Encryption and homomorphic evaluation can be performed with $\mathfrak{p}$ replacing (2). Decryption requires more care. In algorithm Dec' (of the tweaked somewhat homomorphic scheme), one needs to evaluate $\lfloor \boldsymbol{v}_J^{sk} \times \psi \rceil \bmod I$. This can be done by first computing $\pi' = \lfloor \boldsymbol{v}_J^{sk} \times \psi \rceil \bmod (2) = \pi + i \bmod (2)$ and then reducing π' modulo $\mathfrak{p}$ to get plaintext π. Since $\mathfrak{p} = \langle 2, x + 1\rangle$, the quantity $\pi' \bmod \mathfrak{p}$ is the sum of the coefficients of $\pi' \bmod 2$. As explained in [10, Le. 12.3.3], since ideal $\mathfrak{p}$ is of degree 1, we may also first compute the sum of the coefficients of $\boldsymbol{v}_J^{sk} \times \psi$ modulo 2, and then round the result to the closest integer (this requires to decrease the decryption radius by a factor of n, which is negligible). Now, since J is a degree-1 prime ideal and since ψ is reduced modulo the HNF B_J, ciphertext ψ is just an integer modulo $\det J$. Overall, the operation $\lfloor \boldsymbol{v}_J^{sk} \times \psi \rceil \bmod I$ can be performed by first multiplying every component of $\boldsymbol{v}_J^{sk}$ by ψ, rounding the results modulo 2 and keeping $\widetilde{O}(1)$ bits after the rounding point, and then summing the latter quantities modulo 2 and rounding to the closest integer. This requires $\widetilde{O}(n \log q)$ bit operations. The ciphertext expansion can be performed in essentially the same way, except that it is now performed γ_{set} times instead of 1. This dominates the cost of the decryption procedure, and leads to a bit-complexity of $\widetilde{O}(\gamma_{set} n \log q) = \widetilde{O}(\lambda^5)$ for the `Recrypt` procedure.

7.3 Fully homomorphic encryption over the integers

In [7], van Dijk et al. describe another fully homomorphic scheme, whose security relies on the hardness of the Approximate Greatest Common Divisor Problem (as well as the hardness of the SSSP, for the corresponding squashed scheme).

We assume that we use the tree-based decryption circuit that we described in Section 6.2 and gates over the integers (for the generalized circuit). Then without our improvements, the best set of parameters making all known attacks cost at least 2^λ leads to a $\widetilde{O}(\lambda^{17})$ bit-complexity of the `Recrypt` procedure (for a single plaintext bit). An easy improvement consists in evaluating the gates of the generalized circuit modulo 2^κ, where κ is a known upper bound for the bit-lengths of the integer ciphertexts corresponding to the coefficients of the Hamming weights that are used for decrypting (with the tree-based decryption, the most significant bits are also computed, but are useless). This leads to a $\widetilde{O}(\lambda^{16})$ bit-complexity. By using our improvements, the parameters of the scheme can now be chosen so that the bit-complexity of `Recrypt` is lowered to $\widetilde{O}(\lambda^{7.25})$. To obtain that bound, we use the binary tree decryption circuit with gates modulo 2^κ, and we do not re-encrypt the bits of the expanded ciphertext (as explained in [10, Re. 4.1.1]).

It might also be possible to combine our improvements with the variant described in [7, Se. 3.3]. However, this requires some care, as the bootstrappability of this variant depends on the specific decryption circuit that is used, rather than the evaluated function.

Open Problems

Our improved decryption strongly relies on the choices of $f = x^n+1$ and $I|(2)$. It would be interesting to waive these assumptions, in case other choices for f and I prove interesting. An important open question is to assess whether the improvements described in the present article help making Gentry's fully homomorphic scheme more practical (see [27, Se. 6] for a practical study of Gentry's original scheme).

At the end of [10, Se. 12.3], Gentry suggests using non-independent `SplitKey` vectors $\boldsymbol{t}_i$ to decrease the computational costs. The idea is to encode n vectors $\boldsymbol{t}_{i,j} = x^j \boldsymbol{t}_i \bmod x^n + 1$ using only $\boldsymbol{t}_i$. This leads to a faster `Recrypt` procedure when several bits are encoded, using the plaintext domain $\mathbb{Z}_2[x]/f(x)$ (more precisely, it becomes faster in the sense of amortized cost per plaintext bit). However, it is not clear how to homomorphically decrypt with such a variant, as one is now restricted to more complex circuit gates than addition and multiplication modulo 2.

Acknowledgments. We thank Ali Akhavi, Guillaume Hanrot, Steven Galbraith, Craig Gentry and Paul Zimmermann for helpful discussions. The first author was partly supported by the LaRedA ANR grant, and the second author by a Macquarie University Research Fellowship (MQRF) and ARC Discovery Grant DP0987734.

References

1. IEEE Standards Committee 754. ANSI/IEEE standard 754-1985 for binary floating-point arithmetic. Reprinted in SIGPLAN Notices, 22(2):9–25, 1987.
2. M. Ajtai, R. Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *Proceedings of the 33rd Symposium on the Theory of Computing (STOC 2001)*, pages 601–610. ACM Press, 2001.
3. M. Ajtai, R. Kumar, and D. Sivakumar. Sampling short lattice vectors and the closest lattice vector problem. In *Proceedings of the 17th Annual IEEE Conference on Computational Complexity (CCC 17)*, pages 53–57, 2002.
4. L. Babai. On Lovász' lattice reduction and the nearest lattice point problem. *Combinatorica*, 6:1–13, 1986.
5. J. Boyar, R. Peralta, and D. Pochuev. On the multiplicative complexity of boolean functions over the basis $(\wedge, \oplus, 1)$. *Theoretical Computer Science*, 235(1):43–57, 2000.
6. J. W. S. Cassels. *An Introduction to the Geometry of Numbers, 2nd edition.* Springer-Verlag, 1971.
7. M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. Fully homomorphic encryption over the integers. Accepted to Eurocrypt 2010. Available at `http://eprint.iacr.org/2009/616`.
8. N. Gama and P. Q. Nguyen. Finding short lattice vectors within Mordell's inequality. In *STOC '08 – Proc. 40th ACM Symposium on the Theory of Computing*. ACM, 2008.
9. J. von zur Gathen and J. Gerhardt. *Modern Computer Algebra, 2nd edition.* Cambridge University Press, 2003.
10. C. Gentry. *A fully homomorphic encryption scheme.* PhD thesis, Stanford University, 2009. Manuscript available at `http://crypto.stanford.edu/craig`.
11. C. Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of STOC 2009*, pages 169–178. ACM Press, 2009.
12. C. Gentry. Computing arbitrary functions of encrypted data. *Communications of the ACM*, 53(3):97–105, 2010.
13. W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.
14. R. Kannan. Improved algorithms for integer programming and related lattice problems. In *Proceedings of the 15th Symposium on the Theory of Computing (STOC 1983)*, pages 99–108. ACM Press, 1983.

15. R. M. Karp. A survey of parallel algorithms for shared-memory machines. Technical report, 1988.
16. P. N. Klein. Finding the closest lattice vector when it's unusually close. In *Proc. of SODA*, pages 937–941. ACM Press, 2000.
17. J. C. Lagarias and A. M. Odlyzko. Effective versions of the Chebotarev density theorem. In *Algebraic Number Theory, A. Frohlich (ed.)*, pages 409–464. Academic Press, 1977.
18. A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261:513–534, 1982.
19. C. Ludwig. A faster lattice reduction method using quantum search. In *Proc. of ISAAC*, volume 2906 of *Lecture Notes in Computer Science*, pages 199–208. Springer-Verlag, 2003.
20. D. Micciancio. Improving lattice-based cryptosystems using the Hermite normal form. In *Proceedings of the 2001 Cryptography and Lattices Conference (CALC'01)*, volume 2146 of *Lecture Notes in Computer Science*, pages 126–145. Springer-Verlag, 2001.
21. D. Micciancio and S. Goldwasser. *Complexity of lattice problems: a cryptographic perspective.* Kluwer Academic Press, 2002.
22. D. Micciancio and O. Regev. *Post-Quantum Cryptography*, chapter Lattice-based Cryptography. Springer-Verlag, 2008.
23. D. Micciancio and P. Voulgaris. Faster exponential time algorithms for the shortest vector problem. In *Proc. of SODA*. ACM Press, 2010.
24. P. Q. Nguyen and I. Shparlinski. On the insecurity of a server-aided RSA protocol. In *Proc. of ASIACRYPT*, volume 2248 of *Lecture Notes in Computer Science*, pages 21–35. Springer-Verlag, 2001.
25. C. P. Schnorr. A hierarchy of polynomial lattice basis reduction algorithms. *Theoretical Computer Science*, 53:201–224, 1987.
26. A. Schönhage and V. Strassen. Schnelle Multiplikation grosser Zahlen. *Computing*, 7:281–292, 1971.
27. N. P. Smart and F. Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. Accepted to PKC 2010. Available at `http://eprint.iacr.org/2009/571`.