Efficient Fully Homomorphic Encryption from (Standard) LWE

Zvika Brakerski
Weizmann Institute of Science

Vinod Vaikuntanathan
University of Toronto

Abstract— We present a fully homomorphic encryption scheme that is based solely on the (standard) learning with errors (LWE) assumption. Applying known results on LWE, the security of our scheme is based on the worst-case hardness of "short vector problems" on arbitrary lattices.

Our construction improves on previous works in two aspects:

1) We show that "somewhat homomorphic" encryption can be based on LWE, using a new *re-linearization* technique. In contrast, all previous schemes relied on complexity assumptions related to ideals in various rings.
2) We deviate from the "squashing paradigm" used in all previous works. We introduce a new *dimension-modulus reduction* technique, which shortens the ciphertexts and reduces the decryption complexity of our scheme, *without introducing additional assumptions*.

Our scheme has very short ciphertexts and we therefore use it to construct an asymptotically efficient LWE-based single-server private information retrieval (PIR) protocol. The communication complexity of our protocol (in the public-key model) is $k \cdot \mathrm{polylog}(k) + \log |\mathtt{DB}|$ bits per single-bit query (here, k is a security parameter).

1. Introduction

Fully-homomorphic encryption is one of the holy grails of modern cryptography. In a nutshell, a fully homomorphic encryption scheme is an encryption scheme that allows evaluation of arbitrarily complex programs on encrypted data. The problem was suggested by Rivest, Adleman and Dertouzos [34] back in 1978, yet the first plausible candidate came thirty years later with Gentry's breakthrough work in 2009 [13], [14] (although, there has been partial progress in the meanwhile [21], [31], [6], [22]).

Gentry's work showed for the first time a plausible construction of fully homomorphic encryption. However, his solution involved new and relatively untested cryptographic assumptions. Our work aims to base fully homomorphic encryption on standard, well-studied cryptographic assumptions.

Email: zvika.brakerski@weizmann.ac.il. The author's research was supported by ISF grant 710267, BSF grant 710613, and NSF contracts CCF-1018064 and CCF-0729011.

Email: vinodv@cs.toronto.edu. This research was conducted when the author was at Microsoft Research Redmond.

The main building block in Gentry's construction (a so-called "somewhat" homomorphic encryption scheme) was based on the (worst-case, quantum) hardness of problems on *ideal lattices*.[1] Although lattices have become standard fare in cryptography and lattice problems have been relatively well-studied, ideal lattices are a special breed that we know relatively little about. Ideals are a natural building block to construct fully homomorphic encryption in that they natively support both addition and multiplication (whereas lattices are closed under addition only). Indeed, all subsequent constructions of fully homomorphic encryption [36], [11], [7] relied on ideals in various rings in an explicit way. Our first contribution is the construction of a "somewhat" homomorphic encryption scheme whose security relies solely on the (worst-case, classical) hardness of standard problems on *arbitrary* (not necessarily ideal) *lattices*.

Secondly, in order to achieve *full* homomorphism, Gentry had to go through a so-called "squashing step" which forced him to make an additional very strong hardness assumption – namely, the hardness of the (average-case) sparse subset-sum problem. As if by a strange law of nature, all the subsequent solutions encountered the same difficulty as Gentry did in going from a "somewhat" to a fully homomorphic encryption, and they all countered this difficulty by relying on the same sparse subset-sum assumption. This additional assumption was considered to be the main caveat of Gentry's solution and removing it has been, perhaps, the main open problem in the design of fully homomorphic encryption schemes. Our second contribution is to remove the necessity of this additional assumption.

Thus, in a nutshell, we construct a fully homomorphic encryption scheme whose security is based solely on the classical hardness of solving standard lattice problems in the worst-case.[2] Specifically, our scheme is based on the learning with errors (LWE) assumption that is

[1] Roughly speaking, ideal lattices correspond to a geometric embedding of an ideal in a number field. See [25] for a precise definition.

[2] Strictly speaking, under this assumption, our scheme can evaluate polynomial-size circuits with a-priori bounded (but arbitrary) depth. A fully homomorphic encryption scheme independent of the circuit depth can be obtained by making an additional "circular security" assumption (see [8] for details).

known to be at least as hard as solving hard problems in general lattices. Thus our solution does not rely on lattices directly and is fairly natural to understand and implement.

To achieve our goals, we deviate from two paradigms that ruled the design of (a handful of) candidate fully homomorphic encryption schemes [13], [36], [11], [7]:

1) We introduce the *re-linearization* technique, and show how to use it to obtain a *somewhat* homomorphic encryption scheme that does not require hardness assumptions on *ideals*.
2) We present a *dimension-modulus reduction* technique, that turns our somewhat homomorphic scheme into a fully homomorphic one, without the need for the artificial *squashing* step and the sparse subset-sum assumption.

We provide a detailed overview of these new techniques in Sections 1.1 and 1.2 below.

Interestingly, the ciphertexts of the resulting fully homomorphic scheme are very short. This is a desirable property which we use, in conjunction with other techniques, to achieve very efficient private information retrieval protocols. See also Section 1.3 below.

1.1. Re-Linearization: Somewhat Homomorphic Encryption without Ideals

The starting point of Gentry's construction is a "somewhat" homomorphic encryption scheme. For a class of circuits $\mathcal{C}$, a $\mathcal{C}$-homomorphic scheme is one that allows evaluation of any circuit in the class $\mathcal{C}$. The simple, yet striking, observation in Gentry's work is that if a (slightly augmented) decryption circuit for a $\mathcal{C}$-homomorphic scheme resides in $\mathcal{C}$, then the scheme can be converted (or "bootstrapped") into a fully homomorphic encryption scheme.

It turns out that encryption schemes that can evaluate a non-trivial number of addition and multiplication operations[3] are already quite hard to come by (even without requiring that they are bootstrappable).[4] Gentry's solution to this was based on the algebraic notion of *ideals* in rings. In a very high level, the message is considered to be a ring element, and the ciphertext is the message masked with some "noise". The novelty of this idea is that the noise itself belonged to an ideal I. Thus, the ciphertext is of the form $m + xI$ (for some x in the ring). Observe right off the bat that the scheme is born additively homomorphic; in fact, that will be the case with all the schemes we consider in this paper. The ideal I has two main properties: first, a random element in the ideal is assumed to "mask" the message; and second, it is possible to generate a secret trapdoor that "annihilates" the ideal, i.e., implementing the transformation $m + xI \rightarrow m$. The first property guarantees security, while the second enables multiplying ciphertexts. Letting c_1 and c_2 be encryptions of m_1 and m_2 respectively,

$$\begin{aligned} c_1 c_2 &= (m_1 + xI)(m_2 + yI) \\ &= m_1 m_2 + (m_1 y + m_2 x + xyI)I \\ &= m_1 m_2 + zI \end{aligned}$$

When decrypting, the ideal is annihilated and the product $m_1 m_2$ survives. Thus, $c_1 c_2$ is indeed an encryption of $m_1 m_2$, as required. This nifty solution required, as per the first property, a hardness assumption on ideals in certain rings. Gentry's original work relied on hardness assumptions on *ideal lattices*, while van Dijk, Gentry, Halevi and Vaikuntanathan [11] presented a different instantiation that considered ideals over the integers.

Our somewhat homomorphic scheme is based on the hardness of the "learning with errors" (LWE) problem, first presented by Regev [33]. The LWE assumption states that if $\mathbf{s} \in \mathbb{Z}_q^n$ is an n dimensional "secret" vector, any polynomial number of "noisy" random linear combinations of the coefficients of $\mathbf{s}$ are computationally indistinguishable from uniformly random elements in $\mathbb{Z}_q$. Mathematically,

$$\left\{\mathbf{a}_i, \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i\right\}_{i=1}^{\mathrm{poly}(n)} \quad \stackrel{c}{\approx} \quad \left\{\mathbf{a}_i, u_i\right\}_{i=1}^{\mathrm{poly}(n)},$$

where $\mathbf{a}_i \in \mathbb{Z}_q^n$ and $u_i \in \mathbb{Z}_q$ are uniformly random, and the "noise" e_i is sampled from a noise distribution that outputs numbers much smaller than q (an example is a discrete Gaussian distribution over $\mathbb{Z}_q$ with small standard deviation).

The LWE assumption does not refer to ideals, and indeed, the LWE problem is at least as hard as finding short vectors in *any lattice*, as follows from the worst-case to average-case reductions of Regev [33] and Peikert [32]. As mentioned earlier, we have a much better understanding of the complexity of lattice problems (thanks to [23], [2], [27] and many others), compared to the corresponding problems on ideal lattices. In particular, despite considerable effort, the best known algorithms to solve the LWE problem run in

[3] All known scheme, including ours, treat evaluated functions as arithmetic circuits. Hence we use the terminology of "addition and multiplication" gates. The conversion to the boolean model (AND, OR, NOT gates) is immediate.

[4] We must mention here that we are interested only in *compact* fully homomorphic encryption schemes, namely ones where the ciphertexts do not grow in size with each homomorphic operation. If we do allow such growth in size, a number of solutions are possible. See, e.g., [35], [17], [26].

time nearly exponential in the dimension n.[5] The LWE assumption also turns out to be particularly amenable to the construction of simple, efficient and highly expressive cryptographic schemes (e.g., [33], [19], [4], [5], [10], [1] and many others). Our construction of a fully homomorphic encryption scheme from LWE is perhaps a very strong testament to its power and elegance.

Constructing a (secret-key) encryption scheme whose security is based on the LWE assumption is rather straightforward. To encrypt a bit $m \in \{0,1\}$ using secret key $\mathbf{s} \in \mathbb{Z}_q^n$, we choose a random vector $\mathbf{a} \in \mathbb{Z}_q^n$ and a "noise" e and output the ciphertext

$$c \quad = \quad (\mathbf{a}, b = \langle \mathbf{a}, \mathbf{s} \rangle + 2e + m) \qquad \in \mathbb{Z}_q^n \times \mathbb{Z}_q$$

The key observation in decryption is that the two "masks" – namely, the secret mask $\langle \mathbf{a}, \mathbf{s} \rangle$ and the "even mask" $2e$ – do not interfere with each other.[6] That is, one can decrypt this ciphertext by annihilating the two masks, one after the other: The decryption algorithm first re-computes the mask $\langle \mathbf{a}, \mathbf{s} \rangle$ and subtracts it from b, resulting in $2e + m \pmod q$. Since $e \ll q$, $2e + m \pmod q = 2e + m$. Removing the even mask is now easy – simply compute $2e + m$ modulo 2.[7]

As we will see below, the scheme is naturally additive homomorphic, yet multiplication presents a thorny problem. In fact, a recent work of Gentry, Halevi and Vaikuntanathan [18] showed that (a slight variant of) this scheme supports *just a single* homomorphic multiplication, but at the expense of a huge blowup to the ciphertext which made further advance impossible.

To better understand the homomorphic properties of this scheme, let us shift our focus away from the encryption algorithm, on to the decryption algorithm. Given a ciphertext $(\mathbf{a}, b)$, consider the symbolic linear function $f_{\mathbf{a},b} : \mathbb{Z}_q^n \to \mathbb{Z}_q$ defined as:

$$f_{\mathbf{a},b}(\mathbf{x}) = b - \langle \mathbf{a}, \mathbf{x} \rangle \pmod q = b - \sum_{i=1}^{n} \mathbf{a}[i] \cdot \mathbf{x}[i] \quad \in \mathbb{Z}_q$$

where $\mathbf{x} = (\mathbf{x}[1], \ldots, \mathbf{x}[n])$ denotes the variables, and $(\mathbf{a}, b)$ forms the public coefficients of the linear equation. Clearly, decryption of the ciphertext $(\mathbf{a}, b)$ is nothing but evaluating this function on the secret key $\mathbf{s}$ (and then taking the result modulo 2).[8]

Homomorphic addition and multiplication can now be described in terms of this function f. Adding two ciphertexts corresponds to the addition of two linear functions, which is again another linear function. In particular, $f_{(\mathbf{a}+\mathbf{a}', b+b')}(\mathbf{x}) = f_{\mathbf{a},b}(\mathbf{x}) + f_{(\mathbf{a}',b')}(\mathbf{x})$ is the linear function corresponding to the "homomorphically added" ciphertext $(\mathbf{a}+\mathbf{a}', b+b')$. Similarly, multiplying two such ciphertexts corresponds to a symbolic multiplication of these linear equations

$$\begin{aligned} & f_{(\mathbf{a},b)}(\mathbf{x}) \cdot f_{(\mathbf{a}',b)}(\mathbf{x}) \\ & \quad = \quad (b - \sum \mathbf{a}[i]\mathbf{x}[i]) \cdot (b' - \sum \mathbf{a}'[i]\mathbf{x}[i]) \\ & \quad = \quad h_0 \; + \; \sum h_i \; \cdot \; \mathbf{x}[i] \; + \; \sum h_{i,j} \; \cdot \; \mathbf{x}[i]\mathbf{x}[j] \end{aligned}$$

which results in a degree-2 polynomial in the variables $\mathbf{x} = (\mathbf{x}[1], \ldots, \mathbf{x}[n])$, with coefficients $h_{i,j}$ that can be computed from $(\mathbf{a}, b)$ and $(\mathbf{a}', b')$ by opening parenthesis of the expression above. Decryption, as before, involves evaluating this quadratic expression on the secret key $\mathbf{s}$ (and then reducing modulo 2). We now run into a serious problem – the decryption algorithm has to know all the coefficients of this quadratic polynomial, which means that the size of the ciphertext just went up from $n+1$ elements to (roughly) $n^2/2$.

This is where our re-linearization technique comes into play. Re-linearization is a way to reduce the size of the ciphertext back down to $n+1$. The main idea is the following: imagine that we publish "encryptions" of all the linear and quadratic terms in the secret key $\mathbf{s}$, namely all the numbers $\mathbf{s}[i]$ as well as $\mathbf{s}[i]\mathbf{s}[j]$, under a new secret key $\mathbf{t}$. Thus, these ciphertexts (for the quadratic terms) look like $(\mathbf{a}_{i,j}, b_{i,j})$ where

$$b_{i,j} = \langle \mathbf{a}_{i,j}, \mathbf{t} \rangle + 2e_{i,j} + \mathbf{s}[i] \cdot \mathbf{s}[j] \approx \langle \mathbf{a}_{i,j}, \mathbf{t} \rangle + \mathbf{s}[i] \cdot \mathbf{s}[j] \; .^{9}$$

Now, the sum $h_0 + \sum h_i \cdot \mathbf{s}[i] + \sum h_{i,j} \cdot \mathbf{s}[i]\mathbf{s}[j]$ can be written (approximately) as

$$h_0 + \sum h_i (b_i - \langle \mathbf{a}_i, \mathbf{t} \rangle) + \sum_{i,j} h_{i,j} \cdot (b_{i,j} - \langle \mathbf{a}_{i,j}, \mathbf{t} \rangle) \; ,$$

which, lo and behold, is a linear function in $\mathbf{t}$! The bottom-line is that multiplying the two linear functions $f_{(\mathbf{a},b)}$ and $f_{(\mathbf{a}',b')}$ and then re-linearizing the resulting expression results in a linear function (with $n+1$

[5] The nearly exponential time is for a large enough error (i.e., one that is a $1/\mathrm{poly}(n)$ fraction of the modulus q). For smaller errors, as we will encounter in our scheme, there are better – but not significantly better – algorithms. In particular, if the error is a $1/2^{n^\epsilon}$ fraction of the modulus q, the best known algorithm runs in time approx. $2^{n^{1-\epsilon}}$.

[6] We remark that using $2e$ instead of e as in the original formulation of LWE does not adversely impact security, so long as q is odd (since in that case, 2 is a unit in $\mathbb{Z}_q$).

[7] Although the simplified presentation of Gentry's scheme above seems to deal with just one mask (the "secret mask"), in reality, the additional "even mask" existed in the schemes of [13], [11] as well. Roughly speaking, they needed this to ensure semantic security, as we do.

[8] The observation that an LWE-based ciphertext can be interpreted as a linear equation of the secret was also used in [7].

[9] Actually, calling these "encryptions" is inaccurate: $\mathbf{s}[i] \cdot \mathbf{s}[j] \in \mathbb{Z}_q$ is not a single bit and therefore the "ciphertext" cannot be decrypted. However, we feel that thinking of these as encryptions may benefit the reader's intuition.

coefficients), whose evaluation on the new secret key $\mathbf{t}$ results in the product of the two original messages (upon reducing modulo 2). The resulting ciphertext is simply the coefficients of this linear function, of which there are at most $n+1$. This ciphertext will decrypt to $m \cdot m'$ using the secret key $\mathbf{t}$.

In this semi-formal description, we ignored an important detail which has to do with the fact that the coefficients $h_{i,j}$ are potentially large. Thus, even though $(b_{i,j} - \langle \mathbf{a}_{i,j}, \mathbf{t} \rangle) \approx \mathbf{s}[i]\mathbf{s}[j]$, it may be the case that $h_{i,j} \cdot (b_{i,j} - \langle \mathbf{a}_{i,j}, \mathbf{t} \rangle) \not\approx h_{i,j} \cdot \mathbf{s}[i]\mathbf{s}[j]$. This is handled by considering the binary representation of $h_{i,j}$, namely $h_{i,j} = \sum_{\tau=0}^{\lfloor \log q \rfloor} 2^\tau \cdot h_{i,j,\tau}$. If, for each value of τ, we had a pair $(\mathbf{a}_{i,j,\tau}, b_{i,j,\tau})$ such that

$$\begin{aligned} b_{i,j,\tau} &= \langle \mathbf{a}_{i,j,\tau}, \mathbf{t} \rangle + 2e_{i,j,\tau} + 2^\tau \mathbf{s}[i] \cdot \mathbf{s}[j] \\ &\approx \langle \mathbf{a}_{i,j,\tau}, \mathbf{t} \rangle + 2^\tau \mathbf{s}[i] \cdot \mathbf{s}[j] \ , \end{aligned}$$

then indeed

$$\begin{aligned} h_{i,j} \cdot \mathbf{s}[i]\mathbf{s}[j] &= \sum_{\tau=0}^{\lfloor \log q \rfloor} h_{i,j,\tau} 2^\tau \mathbf{s}[i]\mathbf{s}[j] \\ &\approx \sum_{\tau=0}^{\lfloor \log q \rfloor} h_{i,j,\tau} (b_{i,j,\tau} - \langle \mathbf{a}_{i,j,\tau}, \mathbf{t} \rangle) \ , \end{aligned}$$

since $h_{i,j,\tau} \in \{0,1\}$. This increases the number of pairs we need to post by a factor of $(\lfloor \log q \rfloor + 1)$, which is polynomial.

This process allows us to do one multiplication without increasing the size of the ciphertext, and obtain an encryption of the product under a new secret key. *But why stop at two keys* $\mathbf{s}$ *and* $\mathbf{t}$? Posting a "chain" of L secret keys (together with encryptions of quadratic terms of one secret key using the next secret key) allows us to perform up to L levels of multiplications without blowing up the ciphertext size. It is possible to achieve multiplicative depth $L = \epsilon \log n$ (which corresponds to a degree $D = n^\epsilon$ polynomial) for an arbitrary constant $\epsilon < 1$ under reasonable assumptions, but beyond that, the growth of the error in the ciphertext kicks in, and destroys the ciphertext. Handling this requires us to use the machinery of bootstrapping, which we explain in the next section.

In conclusion, the above technique allows us to remove the need for "ideal assumptions" and obtain *somewhat* homomorphic encryption from LWE.

1.2. Dimension-Modulus Reduction: Fully Homomorphic Encryption Without Squashing

As explained above, the "bootstrapping" method for achieving full homomorphism requires a $\mathcal{C}$-homomorphic scheme whose decryption circuit resides in $\mathcal{C}$. All prior somewhat homomorphic schemes fell short in this category and failed to achieve this requirement in a natural way. Thus Gentry, followed by all other previous schemes, resorted to "squashing": a method for reducing the decryption complexity at the expense of making an additional and fairly strong assumption, namely the sparse subset sum assumption.

We show how to "upgrade" our somewhat homomorphic scheme (explained in Section 1.1) into a scheme that enjoys the same amount of homomorphism but has a much smaller decryption circuit. All of this, without making any additional assumptions (beyond LWE)!

Our starting point is the somewhat homomorphic scheme from Section 1.1. Recall that a ciphertext in that scheme is of the form $(\mathbf{a}, b = \langle \mathbf{a}, \mathbf{s} \rangle + 2e + m) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, and decryption is done by computing $(b - \langle \mathbf{a}, \mathbf{s} \rangle \bmod q) \pmod 2$. One can verify that this computation, presented as a polynomial in the bits of $\mathbf{s}$, has degree at least $\max(n, \log q)$, which is more than the maximal degree D that our scheme can homomorphically evaluate. The bottom line is that decryption complexity is governed by $(n, \log q)$ which are too big for our homomorphism capabilities.

Our *dimension-modulus reduction* idea enbales us to take a ciphertext with parameters $(n, \log q)$ as above, and convert it into a ciphertext of the same message, but with parameters $(k, \log p)$ which are much smaller than $(n, \log q)$. To give a hint as to the magnitude of improvement, we typically set k to be of size the security parameter and $p = \mathrm{poly}(k)$. We can then set $n = k^c$ for essentially *any constant* c, and $q = 2^{n^\epsilon}$. We will thus be able to homomorphically evaluate functions of degree roughly $D = n^\epsilon = k^{c \cdot \epsilon}$ and we can choose c to be large enough so that this is sufficient to evaluate the $(k, \log p)$ decryption circuit.

To understand dimension-modulus reduction technically, we go back to re-linearization. We showed above that, posting proper public parameters, one can convert a ciphertext $(\mathbf{a}, b = \langle \mathbf{a}, \mathbf{s} \rangle + 2e + m)$, that corresponds to a secret key $\mathbf{s}$, into a ciphertext $(\mathbf{a}', b' = \langle \mathbf{a}', \mathbf{t} \rangle + 2e' + m)$ that corresponds to a secret key $\mathbf{t}$.[10] The crucial observation is that $\mathbf{s}$ and $\mathbf{t}$ need not have the same dimension n. Specifically, if we chose $\mathbf{t}$ to be of dimension k, the procedure still works. This brings us down from $(n, \log q)$ to $(k, \log q)$, which is a big step but still not sufficient.

Having the above observation in mind, we wonder if we can take $\mathbf{t}$ to have not only low dimension but also

[10] In the previous section, we applied re-linearization to a quadratic function of $\mathbf{s}$, while here we apply it to the ciphertext $(\mathbf{a}, b)$ that corresponds to a linear function of $\mathbf{s}$. This only makes things easier.

small modulus p, thus completing the transition from $(n, \log q)$ to $(k, \log p)$. This is indeed possible using some additional ideas, where the underlying intuition is that $\mathbb{Z}_p$ can "approximate" $\mathbb{Z}_q$ by simple scaling, up to a small error.

The public parameters for the transition from $\mathbf{s}$ to $\mathbf{t}$ will be $(\mathbf{a}_{i,\tau}, b_{i,\tau}) \in \mathbb{Z}_p^k \times \mathbb{Z}_p$, where

$$b_{i,\tau} = \langle \mathbf{a}_{i,\tau}, \mathbf{t} \rangle + e + \left\lfloor \frac{p}{q} \cdot 2^\tau \cdot \mathbf{s}[i] \right\rceil \ .^{11}$$

Namely, we scale $2^\tau \cdot \mathbf{s}[i] \in \mathbb{Z}_q$ into an element in $\mathbb{Z}_p$ by multiplying by p/q and rounding. The rounding incurs an additional error of magnitude at most $1/2$. It follows that

$$2^\tau \cdot \mathbf{s}[i] \approx \frac{q}{p} \cdot (b_{i,\tau} - \langle \mathbf{a}_{i,\tau}, \mathbf{t} \rangle) \ ,$$

which enables converting a linear equation in $\mathbf{s}$ into a linear equation in $\mathbf{t}$. The result of dimension-modulus reduction, therefore, is a ciphertext $(\hat{\mathbf{a}}, \hat{b}) \in \mathbb{Z}_p^k \times \mathbb{Z}_p$ such that $\hat{b} - \langle \hat{\mathbf{a}}, \mathbf{t} \rangle = m + 2\hat{e}$. For security, we need to assume the hardness of LWE with parameters k, p. We can show that in the parameter range we use, this assumption is as hard as the one used for the somewhat homomorphic scheme.[12]

In conclusion, dimension-modulus reduction allows us to achieve a bootstrappable scheme, based on the LWE assumption alone. We refer the reader to Section 3 for the formal presentation of the scheme, and the full version of this paper [8] for the detailed analysis.

As a nice byproduct of this technique, the ciphertexts of the resulting fully homomorphic scheme become very short! They now consist of $(k+1)\log p = O(k \log k)$ bits. This is a desirable property which is also helpful in achieving efficient private information retrieval protocols (see below).

1.3. Near-Optimal Private Information Retrieval

In (single-server) private information retrieval (PIR) protocols, a very large *database* is maintained by a *sender* (the sender is also sometimes called the server, or the database). A *receiver* wishes to obtain a specific entry in the database, without revealing any information about the entry to the server. Typically, we consider databases that are exponential in the security parameter and hence we wish that the receiver's running time and communication complexity are polylogarithmic in the size of the database N (at least $\log N$ bits are required to specify an entry in the database). The first polylogarithmic candidate protocol was presented by Cachin, Micali and Stadler [9] and additional polylograithmic protocols were introduced by Lipmaa [24] and by Gentry and Ramzan [20]. Of which, the latter achieves the best communication complexity of $O(\log^{3-o(1)}(N))$.[13] The latter two protocols achieve constant amortized communication complexity when retrieving large consecutive blocks of data. See a survey in [30] for more details on these schemes.

Fully homomorphic, or even somewhat homomorphic, encryption is known to imply polylogarithmic PIR protocols.[14] Most trivially, the receiver can encrypt the index it wants to query, and the database will use that to homomorphically evaluate the database access function, thus retrieving an encryption of the answer and sending it to the receiver. The total communication complexity of this protocol is the sum of lengths of the public key, encryption of the index and output ciphertext. However, the public key is sent only once, it is independent of the database and the query, and it can be used for many queries. Therefore it is customary to analyze such schemes in the *public key model* where sending the public key does not count towards the communication complexity. Gentry [12] proposes to use his somewhat homomorphic scheme towards this end, which requires $O(\log^3 N)$ bit communication.[15] We show how, using our somewhat homomorphic scheme, in addition to new ideas, we can bring down communication complexity to a near optimal $\log N \cdot \mathrm{polyloglog}\, N$ (one cannot do better than $\log N$). To obtain the best parameters, one needs to assume $2^{\widetilde{\Omega}(k)}$-hardness of polynomial-factor approximation for short vector problems in arbitrary dimension k lattices, which is supported by current knowledge. Details follow.

A major obstacle in the naive use of somewhat homomorphic encryption for PIR is that homomorphism is obtained with respect to the Boolean representa-

[11] A subtle technical point refers to the use of an error term e, instead of $2e$ as we did for re-linearization. The reason is roughly that $\frac{q}{p} \cdot 2$ is non-integer. Therefore we "divide by 2" before performing the dimension-reduction and "multiply back" by 2 after.

[12] For the informed reader we mention that while k, p are smaller than n, q and therefore seem to imply lesser security, we are able to use much higher relative noise in our k, p scheme since it does not need to support homomorphic operations. Hence the two assumptions are of roughly the same hardness.

[13] It is hard to compare the performance of different PIR protocols due to the multitude of parameters. To make things easier to grasp, we compare the protocols on equal grounds: We assume that the database size and the adversary's running time are exponential in the security parameter and assume the maximal possible hardness of the underlying assumption against known attacks. We also assume that each query retrieves a single bit. We will explicitly mention special properties of individual protocols that are not captured by this comparison.

[14] To be precise, one needs sub-exponentially secure such schemes.

[15] Gentry does not provide a detailed analysis of this scheme, the above is based on our analysis of its performance.

tion of the evaluated function. Therefore, the receiver needs to encrypt the index to the database in a bit-by-bit manner. The query is then composed of $\log N$ ciphertexts, which necessitate at least $\log^2 N$ bits of communication. As a first improvement, we notice that the index need not be encrypted under the somewhat homomorphic scheme. Rather, we can encrypt using any *symmetric* encryption scheme. The database will receive, an encrypted symmetric key (under the homomorphic scheme), which will enable it to convert symmetric ciphertexts into homomorphic ciphertexts without additional communication. The encrypted secret key can be sent as a part of the public key as it is independent of the query. This, of course, requires that our somewhat homomorphic scheme can homomorphically evaluate the decryption circuit of the symmetric scheme. Fully homomorphic schemes will certainly be adequate for this purpose, but known somewhat homomorphic schemes are also sufficient (depending on the symmetric scheme to be used). Using the most communication efficient symmetric scheme, we bring down the query complexity to $O(\log N)$. As for the sender's response, our dimension-modulus reduction technique guarantees very short ciphertexts (essentially as short as non-homomorphic LWE based schemes). This translates into $\log N \cdot \mathrm{polyloglog}\, N$ bits per ciphertext, and the communication complexity of our protocol follows. We remark that in terms of retrieving large blocks of consecutive data, one can slightly reduce the overhead to $O(\log N)$ bits of communication for every bit of retrieved data. We leave it as an open problem to bring the amortized communication down to a constant. We refer the reader to the full version [8] for more details.

Prior to this work, it was not at all known how to achieve even polylogarithmic PIR under the LWE assumption. We stress that even if the size of the public key does count towards the communication complexity, our protocol still has polylogarithmic communication.

1.4. Other Related Work

Aside from Gentry's scheme (and a variant thereof by Smart and Vercauteren [36] and an optimization by Stehlé and Steinfeld [37]), there are two other fully homomorphic encryption schemes [11], [7]. The innovation in both these schemes is the construction of a new *somewhat homomorphic* encryption scheme. Both these works then invoke Gentry's *squashing* and bootstrapping transformation to convert it to a fully homomorphic scheme, and thus the security of both these schemes relies on the sparse subset-sum assumption (plus other assumptions). The first of these schemes is due to van Dijk, Gentry, Halevi and Vaikuntanathan [11]. Their scheme works over the integers and relies on a new assumption which, roughly speaking, states that finding a number p given many "noisy" multiples of p is computationally hard. They cannot, however, reduce their assumption to worst-case hardness. The second is a recent work of Brakerski and Vaikuntanathan [7], who construct a somewhat homomorphic encryption scheme based on the ring LWE problem [25] whose security can be reduced to the worst-case hardness of problems on *ideal lattices*.

The efficiency of implementing Gentry's scheme has also gained much attention. Smart and Vercauteren [36], as well as Gentry and Halevi [16] conduct a study on reducing the complexity of implementing the scheme.

In a recent independent work, Gentry and Halevi [15] showed how the sparse subset sum assumption can be replaced by either the (decisional) Diffie-Hellman assumption or an ideal lattice assumption, by representing the decryption circuit as an arithmetic circuit with only one level of (high fan-in) multiplications.

2. PRELIMINARIES

We will let $\mathcal{D}$ denote a distribution over some finite set S. Then, the notation $x \xleftarrow{\$} \mathcal{D}$ means that x is chosen from the distribution $\mathcal{D}$, and $x \xleftarrow{\$} S$, means that x is chosen from the uniform distribution over S. We consider all logarithms to base 2.

In this work, we utilize "noise" distributions over integers. The only property of these distributions we use is their magnitude. Hence, we define a B-bounded distribution to be a distribution over the integers where the magnitude of a sample is bounded with high probability. A definition follows.

Definition 2.1 (B-bounded distributions)**.** *A distribution ensemble $\{\chi_n\}_{n\in\mathbb{N}}$, supported over the integers, is called B-bounded if* $\Pr_{e\xleftarrow{\$}\chi_n}[|e| > B] \leq 2^{-\widetilde{\Omega}(n)}$.

We denote scalars in plain (e.g. x) and vectors in bold lowercase (e.g. $\mathbf{v}$), and matrices in bold uppercase (e.g. $\mathbf{A}$). We will treat all vectors as column vectors. The ℓ_i norm of a vector is denoted by $\|\mathbf{v}\|_i$. Inner product is denoted by $\langle \mathbf{v}, \mathbf{u} \rangle$, recall that $\langle \mathbf{v}, \mathbf{u} \rangle = \mathbf{v}^T \cdot \mathbf{u}$. Let $\mathbf{v}$ be an n dimensional vector. For all $i = 1, \ldots, n$, the i^{th} element in $\mathbf{v}$ is denoted $\mathbf{v}[i]$. We use the convention that $\mathbf{v}[0] \triangleq 1$.

*2.1. Learning With Errors (*LWE*)*

The LWE problem was introduced by Regev [33] as a generalization of "learning parity with noise". For positive integers n and $q \geq 2$, a vector $\mathbf{s} \in \mathbb{Z}_q^n$, and a probability distribution χ on $\mathbb{Z}_q$, let $A_{\mathbf{s},\chi}$ be the distribution obtained by choosing a vector $\mathbf{a} \xleftarrow{\$} \mathbb{Z}_q^n$

uniformly at random and a noise term $e \xleftarrow{\$} \chi$, and outputting $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$. A formal definition follows.

Definition 2.2 (LWE)**.** *For an integer $q = q(n)$ and an error distribution $\chi = \chi(n)$ over $\mathbb{Z}_q$, the learning with errors problem $\mathsf{LWE}_{n,m,q,\chi}$ is defined as follows: Given m independent samples from $A_{\mathbf{s},\chi}$ (for some $\mathbf{s} \in \mathbb{Z}_q^n$), output $\mathbf{s}$ with noticeable probability.*

The (average-case) decision variant of the LWE *problem, denoted $\mathsf{DLWE}_{n,m,q,\chi}$, is to distinguish (with non-negligible advantage) m samples chosen according to $A_{\mathbf{s},\chi}$ (for uniformly random $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$), from m samples chosen according to the uniform distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$.*

For cryptographic applications we are primarily interested in the average case decision problem DLWE, where $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$. There are known quantum [33] and classical [32] reductions from the problem of approximating short vector problems on lattices in the worst-case to solving $\mathsf{DLWE}_{n,m,q,\chi}$ (on the average). Specifically, these reductions take χ to be (discretized versions of) the Gaussian distribution, which is B-bounded for an appropriate B. We only note here that the best known algorithms for these problems run in time nearly exponential in the dimension n [3], [29]. More generally, the best algorithms that solve LWE where the magnitude of the noise is a $1/2^k$ fraction of the modulus q, run in time $2^{\tilde{O}(n/k)}$.

We refer the reader to [8, Section 3] for definitions of bootstrappable and (leveled) fully homomorphic encryption schemes, and [13], [12] for an exposition of Gentry's bootstrapping theorem.

3. The New FHE Scheme

We first describe our bootstrappable encryption scheme BTS in Section 3.1. In Section 3.2, we apply the bootstrapping theorem to BTS, and obtain a fully homomorphic scheme based on LWE. We refer the reader to the full version of this paper [8] for a detailed description and analysis.

3.1. BTS*: A Bootstrappable Encryption Scheme*

Let $\kappa \in \mathbb{N}$ be the security parameter. The scheme is parameterized by dimensions $n, k \in \mathbb{N}$, a positive integer $m \in \mathbb{N}$, odd moduli $q, p \in \mathbb{N}$ (note that q and p need not be prime) and noise distributions χ (over $\mathbb{Z}_q$) and $\hat{\chi}$ (over $\mathbb{Z}_p$). n and q are referred to as the "long" dimension and modulus respectively, while k and p are the "short" dimension and modulus. χ and $\hat{\chi}$ are the long and short noise distributions. Additional parameters of the scheme are $m \in \mathbb{N}$ which is used towards public key generation, and $L \in \mathbb{N}$ which is an upper bound on the maximal multiplicative depth that the scheme can homomorphically evaluate. The message space of the encryption scheme is GF(2).

We encourage the reader to consider the following (non-optimal, but easier to understand) settings as a running example: $k = \kappa$, $n = k^4$, $q \approx 2^{\sqrt{n}}$, $p = (n^2 \log q) \cdot \mathrm{poly}(k) = \mathrm{poly}(k)$, $m = O(n \log q)$ and $L = 1/3 \log n = 4/3 \log k$. The distributions $\chi, \hat{\chi}$ can be thought of as being n- and k-bounded, respectively.

Key generation $\mathsf{BTS.Keygen}(1^\kappa)$*:*

The key generation algorithm first samples $L+1$ "long vectors" $\mathbf{s}_0, \ldots, \mathbf{s}_L \xleftarrow{\$} \mathbb{Z}_q^n$, and computes, for all $\ell \in [L]$, $0 \le i \le j \le n$, and $\tau \in \{0, \ldots, \lfloor \log q \rfloor\}$, the value $\psi_{\ell,i,j,\tau} := (\mathbf{a}_{\ell,i,j,\tau}, b_{\ell,i,j,\tau}) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ where:

$$b_{\ell,i,j,\tau} := \langle \mathbf{a}_{\ell,i,j,\tau}, \mathbf{s}_\ell \rangle + 2 \cdot e_{\ell,i,j,\tau} + 2^\tau \cdot \mathbf{s}_{\ell-1}[i] \cdot \mathbf{s}_{\ell-1}[j] \tag{1}$$

Here, $\mathbf{a}_{\ell,i,j,\tau} \xleftarrow{\$} \mathbb{Z}_q^n$, $e_{\ell,i,j,\tau} \xleftarrow{\$} \chi$ (recall that, according to our notational convention, $\mathbf{s}_{\ell-1}[0] \triangleq 1$). Note that the pair $(\mathbf{a}_{\ell,i,j,\tau}, b_{\ell,i,j,\tau})$ is similar to an encryption of $2^\tau \cdot \mathbf{s}_{\ell-1}[i] \cdot \mathbf{s}_{\ell-1}[j] \pmod q$ via an LWE-based scheme, except this "ciphertext" is not decryptable since the "message" is not a single bit value. Thus, we prefer to call them "pseudo-encryptions". Let $\Psi \triangleq \{\psi_{\ell,i,j,\tau}\}_{\ell,i,j,\tau}$ be the set of all these values.

Secondly, sample a "short" vector $\hat{\mathbf{s}} \xleftarrow{\$} \mathbb{Z}_p^k$ and compute additional parameters: For all $i \in [n]$, $\tau \in \{0, \ldots, \lfloor \log q \rfloor\}$, sample $\hat{\mathbf{a}}_{i,\tau} \xleftarrow{\$} \mathbb{Z}_p^k$, $\hat{e}_{i,\tau} \xleftarrow{\$} \hat{\chi}$, and compute

$$\hat{b}_{i,\tau} := \langle \hat{\mathbf{a}}_{i,\tau}, \hat{\mathbf{s}} \rangle + \hat{e}_{i,\tau} + \left\lfloor \frac{p}{q} \cdot \left(2^\tau \cdot \mathbf{s}_L[i]\right) \right\rceil \pmod p \ .$$

Set $\hat{\psi}_{i,\tau} := \left(\hat{\mathbf{a}}_{i,\tau}, \hat{b}_{i,\tau}\right) \in \mathbb{Z}_p^k \times \mathbb{Z}_p$, and

$$\hat{\Psi} := \{\hat{\psi}_{i,\tau}\}_{i \in [n], \tau \in \{0,\ldots,\lfloor \log q \rfloor\}} \ .$$

The key-generation algorithm proceeds to choose a uniformly random matrix $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$ and a vector $\mathbf{e} \xleftarrow{\$} \chi^m$, and compute $\mathbf{b} := \mathbf{A}\mathbf{s}_0 + 2\mathbf{e}$.

It then outputs the secret key $sk = \hat{\mathbf{s}}$, the evaluation key $evk = (\Psi, \hat{\Psi})$, and the public key $pk = (\mathbf{A}, \mathbf{b})$.[16]

Encryption $\mathsf{BTS.Enc}_{pk}(\mu)$*:*

Recall that $pk = (\mathbf{A}, \mathbf{b})$. To encrypt a message $\mu \in$ GF(2), sample a vector $\mathbf{r} \xleftarrow{\$} \{0,1\}^m$ and set (just like in Regev's scheme)

$$\mathbf{v} := \mathbf{A}^T \mathbf{r} \qquad \text{and} \qquad w := \mathbf{b}^T \mathbf{r} + \mu \ .$$

[16] The public key pk is essentially identical to the public key in Regev's scheme.

The output ciphertext contains the pair $(\mathbf{v}, w)$, in addition to a "level tag" which is used during homomorphic evaluation and indicates the "multiplicative depth" where the ciphertext has been generated. For freshly encrypted ciphertext, therefore, the level tag is zero. Formally, the encryption algorithm outputs $c{:=}((\mathbf{v}, w), 0)$.

Homomorphic evaluation $\mathsf{BTS.Eval}_{evk}(f, c_1, \ldots, c_t)$*:*

Here, the function $f : \{0,1\}^t \to \{0,1\}$ is represented by a binary arithmetic circuit with '$+$' gates of arbitrary fan-in and '$\times$' gates with fan-in 2. We further require that the circuit is *layered*, namely that it is composed of homogenous layers of either all '$+$' gates or all '$\times$' gates (it is easy to see that any arithmetic circuit can be converted to this form). Lastly, we require that the multiplicative depth of the circuit (the total number of '$\times$' layers) is exactly L.

We homomorphically evaluate the circuit f gate by gate. Namely, we will show how to perform homomorphic addition (of arbitrarily many ciphertexts) and homomorphic multiplication (of two ciphertexts). Combining the two, we will be able to evaluate any such function f. The last step in homomorphic evaluation is the "dimension and modulus reduction" step. Looking ahead, we note that it is this final step that makes the scheme bootstrappable.

Ciphertext structure during evaluation: During the homomorphic evaluation, we will generate ciphertexts of the form $c = ((\mathbf{v}, w), \ell)$, where the tag ℓ indicates the multiplicative level at which the ciphertext has been generated (hence fresh ciphertexts are tagged with 0). The requirement that f is layered will make sure that throughout the homomorphic evaluation all inputs to a gate have the same tag. In addition, we will keep the invariant that the output of each gate evaluation $c = ((\mathbf{v}, w), \ell)$, is such that

$$w - \langle \mathbf{v}, \mathbf{s}_\ell \rangle = \mu + 2 \cdot e \pmod q \ , \tag{2}$$

where μ is the correct plaintext output of the gate, and e is a noise term that depends on the gate's input ciphertexts. Note that it always holds that $\ell \leq L$ due to the bound on the multiplicative depth, and that the output of the homomorphic evaluation of the entire circuit is expected to have $\ell = L$.

Homomorphic evaluation of gates:

– *Addition gates.* Homomorphic evaluation of a '$+$' gate on inputs $c_1, \ldots, c_t$, where $c_i = ((\mathbf{v}_i, w_i), \ell)$, is performed by outputting

$$c_{\mathsf{add}} = ((\mathbf{v}_{\mathsf{add}}, w_{\mathsf{add}}), \ell) {:=} \left(\left(\sum_i \mathbf{v}_i, \sum_i w_i \right), \ell \right) \ .$$

Informally, one can see that

$$\begin{aligned} w_{\mathsf{add}} - \langle \mathbf{v}_{\mathsf{add}}, \mathbf{s}_\ell \rangle &= \sum_i (w_i - \langle \mathbf{v}_i, \mathbf{s}_\ell \rangle) \\ = \sum_i (\mu_i + 2e_i) &= \sum_i \mu_i + 2 \sum_i e_i \ , \end{aligned}$$

where μ_i is the plaintext corresponding to c_i (thus, satisfying the invariant Equation 2). The output of the homomorphic evaluation, thus, corresponds to the sum of the inputs, with the noise term being the sum of input noises.

– *Multiplication gates.* We show how to multiply ciphertexts c, c' where $c = ((\mathbf{v}, w), \ell)$ and $c' = ((\mathbf{v}', w'), \ell)$ (recall that multiplication gates have fan-in 2), to obtain an output ciphertext $c_{\mathsf{mult}} = ((\mathbf{v}_{\mathsf{mult}}, w_{\mathsf{mult}}), \ell + 1)$. Note that the level tag increases by 1.

We first consider an n-variate *symbolic* polynomial over the unknown vector $\mathbf{x}$:

$$\phi(\mathbf{x}) = \phi_{(w,\mathbf{v}),(w',\mathbf{v}')}(\mathbf{x}) \triangleq (w - \langle \mathbf{v}, \mathbf{x} \rangle) \cdot (w' - \langle \mathbf{v}', \mathbf{x} \rangle) \ . \tag{3}$$

We symbolically open the parenthesis of this quadratic polynomial, and express it as

$$\phi(\mathbf{x}) = \sum_{0 \leq i \leq j \leq n} h_{i,j} \cdot \mathbf{x}[i] \cdot \mathbf{x}[j] \ ,$$

where $h_{i,j} \in \mathbb{Z}_q$ are known (we can compute them from $(\mathbf{v}, w), (\mathbf{v}', w')$ by opening parenthesis in Eq. (3)).[17]

For technical reasons (related to keeping the error growth under control), we want to express $\phi(\cdot)$ as a polynomial with small coefficients. We consider the binary representation of $h_{i,j}$, letting $h_{i,j,\tau}$ be the τ^{th} bit in this representation. In other words $h_{i,j} = \sum_{\tau=0}^{\lfloor \log q \rfloor} h_{i,j,\tau} \cdot 2^\tau$, for $h_{i,j,\tau} \in \{0,1\}$.

We can express ϕ therefore as

$$\phi(\mathbf{x}) = \sum_{\substack{0 \leq i \leq j \leq n \\ \tau \in \{0, \ldots, \lfloor \log q \rfloor\}}} h_{i,j,\tau} \cdot \left(2^\tau \cdot \mathbf{x}[i] \cdot \mathbf{x}[j] \right) \ .^{18}$$

We recall that the evaluation key $evk = \Psi$ contains elements of the form $\psi_{\ell,i,j,\tau} = (\mathbf{a}_{\ell,i,j,\tau}, b_{\ell,i,j,\tau})$ such that

$$2^\tau \mathbf{s}_\ell[i] \mathbf{s}_\ell[j] \approx b_{\ell+1,i,j,\tau} - \langle \mathbf{a}_{\ell+1,i,j,\tau}, \mathbf{s}_{\ell+1} \rangle \ .$$

[17] We once again remind the reader that because of the notational trick of setting $\mathbf{x}[0] \triangleq 1$, this expression captures the constant term in the product, as well as all the linear terms, thus homogenizing the polynomial $\phi(\mathbf{x})$.

[18] This can be interpreted as a polynomial with small coefficients whose variables are $(2^\tau \cdot \mathbf{x}[i] \cdot \mathbf{x}[j])$.

The homomorphic multiplication algorithm will thus set

$$\mathbf{v}_{\mathsf{mult}} := \sum_{\substack{0\le i\le j\le n \\ \tau\in\{0,\ldots,\lfloor\log q\rfloor\}}} h_{i,j,\tau}\cdot \mathbf{a}_{\ell+1,i,j,\tau} \ , \text{ and}$$

$$w_{\mathsf{mult}} = \sum_{\substack{0\le i\le j\le n \\ \tau\in\{0,\ldots,\lfloor\log q\rfloor\}}} h_{i,j,\tau}\cdot b_{\ell+1,i,j,\tau} \ ,$$

A simple calculation shows that $w_{\mathsf{mult}} - \langle \mathbf{v}_{\mathsf{mult}}, \mathbf{s}_{\ell+1}\rangle = \sum_{i,j,\tau} h_{i,j,\tau}(b_{\ell+1,i,j,\tau} - \langle \mathbf{a}_{\ell+1,i,j,\tau}, \mathbf{s}_{\ell+1}\rangle) = \phi(\mathbf{s}_\ell) + 2e_1 = \mu\mu' + 2e_2$, where e_1 and e_2 are "small" error terms (thus, satisfying Invariant 2).

Dimension and Modulus Reduction. After homomorphic evaluation of f, we reduce the dimension and modulus of the resulting ciphertext c_f (from (n,q) to (k,p)) as follows. Consider the following function from $\mathbb{Z}^n$ into the rationals modulo p

$$\phi(\mathbf{x}) \triangleq \phi_{\mathbf{v},w}(\mathbf{x}) \triangleq \frac{p}{q}\cdot\left(\frac{q+1}{2}\cdot(w - \langle \mathbf{v},\mathbf{x}\rangle)\right) \pmod p \ .$$

Rearranging, one can find $h_0,\ldots,h_n \in \mathbb{Z}_q$ such that

$$\phi(\mathbf{x}) = \sum_{i=0}^{n} h_i\cdot(\frac{p}{q}\cdot\mathbf{x}[i]) \pmod p \ ,$$

Let $h_{i,\tau}$ be the τ^{th} bit of h_i, for all $\tau\in\{0,\ldots,\lfloor\log q\rfloor\}$. Then

$$\phi(\mathbf{x}) = \sum_{i=0}^{n}\sum_{\tau=0}^{\lfloor\log q\rfloor} h_{i,\tau}\cdot(\frac{p}{q}\cdot 2^\tau\cdot\mathbf{x}[i]) \ .$$

Using the parameters in $\hat{\Psi}$, we create a new ciphertext $\hat{c} = (\hat{\mathbf{v}},\hat{w}) \in \mathbb{Z}_p^k \times \mathbb{Z}_p$ by setting

$$\hat{\mathbf{v}} := 2\cdot\sum_{i=0}^{n}\sum_{\tau=0}^{\lfloor\log q\rfloor} h_{i,\tau}\cdot\hat{\mathbf{a}}_{i,\tau} \pmod p \ \in \mathbb{Z}_p^k$$

$$\hat{w} := 2\cdot\sum_{i=0}^{n}\sum_{\tau=0}^{\lfloor\log q\rfloor} h_{i,\tau}\cdot\hat{b}_{i,\tau} \pmod p \ \in \mathbb{Z}_p \ .$$

By a similar calculation as above, one can verify that the ciphertext $(\hat{\mathbf{v}},\hat{w})$ satisfies Invariant 2.

The output of $\mathsf{BTS.Eval}$ is the new ciphertext $\hat{c} \in \mathbb{Z}_p^k \times \mathbb{Z}_p$. Note that the bit-length of $\hat{c}$ is $(k+1)\log p$.

Decryption $\mathsf{BTS.Dec}_{\hat{\mathbf{s}}}(\hat{c})$*:*

To decrypt $\hat{c} = (\hat{\mathbf{v}},\hat{w}) \in \mathbb{Z}_p^k \times \mathbb{Z}_p$ (recall, again, that we only need to decrypt ciphertexts that are output by $\mathsf{BTS.Eval}$), compute $\mu^* := (\hat{w} - \langle\hat{\mathbf{v}},\hat{\mathbf{s}}\rangle \pmod p) \pmod 2$.

If indeed $\hat{w} - \langle\hat{\mathbf{v}},\hat{\mathbf{s}}\rangle = \mu + 2\hat{e} \pmod p$ then $\mu^* = \mu$ so long as $\hat{e}$ is small enough.

The following theorem summarizes the homomorphic properties of BTS. For a proof, see the full version [8].

Theorem 3.1. *Let $n = n(\kappa) \ge 5$ be any polynomial, $q \ge 2^{n^\epsilon} \ge 3$ for some $\epsilon \in (0,1)$ be odd, χ be any n-bounded distribution, and $m = (n+1)\log q + 2\kappa$. Let $k = \kappa$, $p = 16nk\log(2q)$ (odd) and $\hat{\chi}$ be any k-bounded distribution. Then* BTS *can homomorphically evaluate Boolean circuits of multiplicative depth $O(\epsilon\log n)$. Furthermore, under the* $\mathsf{DLWE}_{n,q,\chi}$ *and the* $\mathsf{DLWE}_{k,p,\hat{\chi}}$ *assumptions, the scheme* BTS *is also CPA secure.*

3.2. Bootstrapping and Full Homomorphism

We now show how to apply Gentry's bootstrapping theorem [13, Theorem 1] to achieve full homomorphism. In order to do this, we first need to bound the complexity of the (augmented) decryption circuit. Since our decryption is essentially the computation of an inner product, we first bound the complexity of this operation.

Lemma 3.2. *Let $(\hat{\mathbf{v}},\hat{w}) \in \mathbb{Z}_p^k \times \mathbb{Z}_p$. There exists an arithmetic circuit over $GF(2)$ with fan-in 2 gates and $O(\log k + \log\log p)$ depth, that on input $\hat{\mathbf{s}} \in \mathbb{Z}_p^k$ (written in binary) computes $(\hat{w} - \langle\hat{\mathbf{v}},\hat{\mathbf{s}}\rangle \pmod p) \pmod 2$.*

We can now apply the bootstrapping theorem to obtain a fully homomorphic scheme. In particular, there is a constant $C\in\mathbb{N}$ such that setting $n = k^{C/\epsilon}$ and the rest of the parameters as in Theorem 3.1, we see that BTS can homomorphically evaluate its own decryption circuit (and then some). This shows that BTS is a bootstrappable encryption scheme. Applying Gentry's bootstrapping theorem [13, Theorem 1], we get:

Theorem 3.3. *There exists a leveled FHE scheme based on the* $\mathsf{DLWE}_{n,q,\chi}$ *and* $\mathsf{DLWE}_{k,p,\hat{\chi}}$ *assumptions. Furthermore, if* BTS *is weakly circular secure (see [8] for a formal definition), then there exists an FHE scheme based on the same assumptions.*

REFERENCES

[1] S. Agrawal, D. Boneh, and X. Boyen, "Efficient lattice (H)IBE in the standard model," in *EUROCRYPT*, 2010, pp. 553–572.

[2] M. Ajtai, "The shortest vector problem in ℓ_2 is NP-hard for randomized reductions (extended abstract)," in *STOC*, 1998, pp. 10–19.

[3] M. Ajtai, R. Kumar, and D. Sivakumar, "A sieve algorithm for the shortest lattice vector problem," in *STOC*, 2001, pp. 601–610.

[4] A. Akavia, S. Goldwasser, and V. Vaikuntanathan, "Simultaneous hardcore bits and cryptography against memory attacks," in *TCC*, ser. Lecture Notes in Computer Science, O. Reingold, Ed., vol. 5444. Springer, 2009, pp. 474–495.

[5] B. Applebaum, D. Cash, C. Peikert, and A. Sahai, "Fast cryptographic primitives and circular-secure encryption based on hard learning problems," in *CRYPTO*, ser. Lecture Notes in Computer Science, S. Halevi, Ed., vol. 5677. Springer, 2009, pp. 595–618.

[6] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-DNF formulas on ciphertexts," in *Theory of Cryptography - TCC'05*, ser. Lecture Notes in Computer Science, vol. 3378. Springer, 2005, pp. 325–341.

[7] Z. Brakerski and V. Vaikuntanathan, " Fully homomorphic encryption from ring-LWE and security for key dependent messages," in *CRYPTO*, vol. 6841, 2011, p. 501.

[8] ——, "Efficient fully homomorphic encryption from (standard) LWE," Cryptology ePrint Archive, Report 2011/344, 2011, http://eprint.iacr.org/2011/344.

[9] C. Cachin, S. Micali, and M. Stadler, "Computationally private information retrieval with polylogarithmic communication," in *EUROCRYPT*, 1999, pp. 402–414.

[10] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert, "Bonsai trees, or how to delegate a lattice basis," in *EUROCRYPT*, 2010, pp. 523–552.

[11] M. Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully homomorphic encryption over the integers," in *EUROCRYPT*, 2010, pp. 24–43, full Version in http://eprint.iacr.org/2009/616.pdf.

[12] C. Gentry, "A fully homomorphic encryption scheme," Ph.D. dissertation, Stanford University, 2009, http://crypto.stanford.edu/craig.

[13] ——, "Fully homomorphic encryption using ideal lattices," in *STOC*, 2009, pp. 169–178.

[14] ——, "Toward basing fully homomorphic encryption on worst-case hardness," in *CRYPTO*, 2010, pp. 116–137.

[15] C. Gentry and S. Halevi, "Fully homomorphic encryption without squashing using depth-3 arithmetic circuits," Cryptology ePrint Archive, Report 2011/279, 2011, to appear in FOCS 2011.

[16] ——, "Implementing gentry's fully-homomorphic encryption scheme," in *EUROCRYPT*, ser. Lecture Notes in Computer Science, K. G. Paterson, Ed., vol. 6632. Springer, 2011, pp. 129–148.

[17] C. Gentry, S. Halevi, and V. Vaikuntanathan, "*i*-hop homomorphic encryption and rerandomizable Yao circuits," in *CRYPTO*, 2010, pp. 155–172.

[18] ——, "A simple BGN-type cryptosystem from LWE," in *EUROCRYPT*, 2010, pp. 506–522.

[19] C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *STOC*, C. Dwork, Ed. ACM, 2008, pp. 197–206.

[20] C. Gentry and Z. Ramzan, "Single-database private information retrieval with constant communication rate," in *ICALP*, ser. Lecture Notes in Computer Science, L. Caires, G. F. Italiano, L. Monteiro, C. Palamidessi, and M. Yung, Eds., vol. 3580. Springer, 2005, pp. 803–815.

[21] S. Goldwasser and S. Micali, "Probabilistic encryption and how to play mental poker keeping secret all partial information," in *STOC*. ACM, 1982, pp. 365–377.

[22] Y. Ishai and A. Paskin, "Evaluating branching programs on encrypted data," in *TCC*, ser. Lecture Notes in Computer Science, S. P. Vadhan, Ed., vol. 4392. Springer, 2007, pp. 575–594.

[23] A. K. Lenstra, H. W. Lenstra, and L. Lovsz, "Factoring polynomials with rational coefficients," *Mathematische Annalen*, vol. 261, pp. 515–534, 1982, 10.1007/BF01457454.

[24] H. Lipmaa, "An oblivious transfer protocol with log-squared communication," in *ISC*, ser. Lecture Notes in Computer Science, J. Zhou, J. Lopez, R. H. Deng, and F. Bao, Eds., vol. 3650. Springer, 2005, pp. 314–328.

[25] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," in *EUROCRYPT*, 2010, pp. 1–23, draft of full version was provided by the authors.

[26] C. A. Melchor, P. Gaborit, and J. Herranz, "Additively homomorphic encryption with *d*-operand multiplications," in *CRYPTO*, 2010, pp. 138–154.

[27] D. Micciancio, "The shortest vector in a lattice is hard to approximate to within some constant," *SIAM J. Comput.*, vol. 30, no. 6, pp. 2008–2035, 2000.

[28] ——, "A first glimpse of cryptography's holy grail," *Commun. ACM*, vol. 53, pp. 96–96, March 2010. [Online]. Available: http://doi.acm.org/10.1145/1666420.1666445

[29] D. Micciancio and P. Voulgaris, "A deterministic single exponential time algorithm for most lattice problems based on voronoi cell computations," in *STOC*, L. J. Schulman, Ed. ACM, 2010, pp. 351–358.

[30] R. Ostrovsky and W. E. Skeith III, "A survey of single-database private information retrieval: Techniques and applications," in *Public Key Cryptography*, ser. Lecture Notes in Computer Science, T. Okamoto and X. Wang, Eds., vol. 4450. Springer, 2007, pp. 393–411.

[31] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *EUROCRYPT*, 1999, pp. 223–238.

[32] C. Peikert, "Public-key cryptosystems from the worst-case shortest vector problem: extended abstract," in *STOC*, 2009, pp. 333–342.

[33] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," in *STOC*, H. N. Gabow and R. Fagin, Eds. ACM, 2005, pp. 84–93.

[34] R. Rivest, L. Adleman, and M. Dertouzos, "On data banks and privacy homomorphisms," in *Foundations of Secure Computation*. Academic Press, 1978, pp. 169–177.

[35] T. Sander, A. Young, and M. Yung, "Non-interactive cryptocomputing for NC^1," in *FOCS*, 1999, pp. 554–567.

[36] N. P. Smart and F. Vercauteren, "Fully homomorphic encryption with relatively small key and ciphertext sizes," in *Public Key Cryptography*, ser. Lecture Notes in Computer Science, P. Q. Nguyen and D. Pointcheval, Eds., vol. 6056. Springer, 2010, pp. 420–443.

[37] D. Stehlé and R. Steinfeld, "Faster fully homomorphic encryption," in *ASIACRYPT*, 2010, pp. 377–394.