Fully Homomorphic Encryption using Hidden Ideal Lattice

Thomas Plantard, Willy Susilo, *Senior Member, IEEE*, Zhenfei Zhang

***Abstract*—All the existing fully homomorphic encryption schemes are based on three different problems, namely bounded distance decoding problem over ideal lattice, approximate greatest common divisor problem over integers and learning with error problem. In this paper, we unify the first two families of problems by introducing a new class of problems, which can be reduced from both problems. Based on this new problem, namely *the bounded distance decoding over hidden ideal lattice*, we present a new fully homomorphic encryption scheme. Since it is a combination of the two problems to some extent, the performance of our scheme lies between the ideal lattice based schemes and the integer based schemes. Furthermore, we also show a lower and upper bound of the problem our scheme is based on. As a result, we present a security conjecture. Assuming this security conjecture holds, we can incorporate smaller parameters, which will result in a scheme that is more efficient than both lattice based and integer based schemes.**

Hence, our scheme makes a perfect alternative to the state-of-art ring learning with error based schemes.

Keywords: Hidden Lattice, Ideal Lattice, Bounded Distance Decoding problem, Fully Homomorphic Encryption, Approximate Greatest Common Divisor.

I. Introduction

A homomorphic public key encryption scheme that supports arbitrary operations on encrypted data has been a "holy grail" of cryptography for over 30 years [28]. Recently, Gentry [8] proposed the very first fully homomorphic encryption (FHE) scheme. Moreover, he presented a methodology of how to construct fully homomorphic encryption schemes. Gentry's framework consists of three steps.

- He firstly constructed a "somewhat homomorphic" encryption (SHE) scheme that supports evaluation of low degree polynomials.
- Then he "squashed" the decryption algorithm to obtain a lower circuit depth so that the somewhat scheme is capable of evaluating its own decryption circuit.
- Finally, he used a "bootstrapping" technique to achieve a fully homomorphic encryption scheme.

The main point here is to find a "bootstrappable" SHE scheme, which refers to the maximum depth of the supported circuits is greater than twice of the depth of the decryption circuit.

This work is supported by ARC Future Fellowship FT0991397.

T. Plantard, W. Susilo and Z. Zhang are with the Centre for Computer and Information Security Research (CCISR), School of Computer Science & Software Engineering (SCSSE), University Of Wollongong, Australia. Email: {thomaspl, wsusilo}@uow.edu.au, zz920@uowmail.uow.edu.au

A. Related Work

The initial construction of Gentry's framework is based on ideal lattice [8], [9], [10]. Its somewhat homomorphic encryption scheme is a GGH-type cryptosystem [16], i.e., the secret key/public key are "good"/"bad" basis of the lattice, and the underlying lattice problem is a Bounded Distance Decoding problem over ideal lattice (BDDi, see Definition 7). The encryption is to map a message to a vector close to the lattice using the bad basis, while with the good basis, one can perform the vector reduction to recover the message. For two vectors that are close to the ideal lattice, operations (additions and multiplications) on the vectors will result in a new vector whose distance to the lattice is short enough for a correct decryption. Therefore, the property of homomorphic encryption is provided.

However, during the evaluations, the noise (i.e., the distance between a ciphertext vector and the lattice) grows, and when it exceeds a threshold, the ciphertext cannot be decrypted correctly. Gentry used a "refresh" procedure to reduce the noise. Since the scheme is bootstrappable, it is capable of evaluating its own decryption circuit. During this evaluation, the ciphertext is recrypted. The original noise is eliminated and a new noise (much smaller) is induced. By doing this repetitively, one is able to evaluate circuit with any depth, therefore, a fully homomorphic encryption scheme is achieved.

Following Gentry's framework, a few FHE schemes are proposed, which can mainly be divided into three categories:

Ideal Lattice based schemes: One of the first variants proposed shortly after the initial construction was made by Smart and Vercauteren [31]. They used the "principal ideal lattice" where the lattice can be represented by two integers. Therefore, they maintained a smaller key size and a simpler encryption/decryption algorithm. However, one major obstacle of this scheme is its inefficient key generation algorithm. Indeed, one is required to find a lattice with a prime determinant, and this criteria is impractical with a large dimension, for instance, 2048, which will lead to a larger determinant, and hence making the probability of the determinant being prime to be smaller. Later, Gentry and Halevi presented an alternative solution to avoid this issue in [12], together with some other optimizations, which is by far the most efficient fully homomorphic encryption scheme using ideal lattice.

Other optimizations on fully homomorphic encryption schemes based on ideal lattice have been proposed by Stehlé and Steinfeld in [32]. They improved the efficiency of Gentry's original scheme. Part of their techniques are adopted in [12] as well. There is also an improvement proposed by Loftus et al. in [21], which deals with the CCA-1 security of Gentry

and Halevi's work. Nevertheless, this variant does not improve the efficiency of the system and therefore, Gentry and Halevi's scheme is still regarded as the most efficient fully homomorphic encryption scheme based on ideal lattice.

Integer based schemes: In [33], van Dijk et al. proposed another fully homomorphic encryption scheme where the somewhat homomorphic scheme is based on the general version of Approximate Greatest Common Divisor of integers (AGCD, see Definition 9). In [6], Coron et al. showed an optimization of this scheme, where the security is based on a partial version of the AGCD problem. The hardness of the partial version was soon re-evaluated in [26], [5]. In this paper, we mainly focus on the first variant of AGCD problem, since it is in general harder to solve.

So far, the best implementation of integer based FHE scheme was presented in [7]. We note that homomorphic encryption schemes based on AGCD problems provide an interesting alternative to the use of ideal lattice, but none of them is as efficient as Gentry and Halevi's scheme.

LWE based schemes: The state-of-the-art of FHE is based on the Learning With Error problem (LWE) [2] and Ring-LWE problem [3]. The work in [14], [15] delivers the best efficiency among all fully homomorphic encryption schemes to date.

The major advantage of using LWE is that one can freely select the moduli [1] and the ring [13]. By using some moduli with a special form (i.e., $2^n + 1$, where n is an integer), one can use some noise control techniques [1], or omit the second step in the framework [14]. Unfortunately, this technique cannot be adopted in schemes using ideal lattice, since the moduli is the determinant of the lattice, hence, it does not possess such freedom. We also note that there is no known reductions between Bounded Distance Decoding (BDD) problem over ideal lattice and the learning with error problem. In this work, we focus on the first direction, namely, construction of a FHE scheme with ideal lattice.

B. Our Contribution

In this paper, we unify two families of FHE schemes by presenting a new fully homomorphic encryption scheme using a hidden ideal lattice. We note Our scheme does not rely on the sparse sub set sum problem (SSSP), and therefore, the security of our squashed scheme remains the same as our SHE scheme. Using the conjectured security, our scheme outperforms all integer based/ideal lattice based FHE schemes, therefore, our scheme is a perfect alternative to the learning with error based schemes, whose security is based on a different type of problem.

Our scheme is based on an observation: it is not essential to publish the lattice to construct a fully homomorphic encryption scheme. In fact, one can operate on vectors close to a lattice, without knowing the lattice. If many bounded distance vectors of this lattice is provided, the lattice is unique. Therefore one can use these vectors, instead of a bad basis of the lattice, to encrypt. Informally, we call this lattice as "*the hidden lattice*". We show that with this new public key, one is still capable of conducting encryption/decryption correctly.

In terms of security, we base the CPA security on a bounded distance decoding problem over hidden ideal lattice (BDDH problem). We also show that this problem is harder than both the BDD over ideal lattice and AGCD problems, which are the two out of three main problems that have been used to design fully homomorphic encryption schemes do date.

In this paper, we show that BDDH problem over dimension n is equivalent to BDD problem over dimension $O(n^\xi)$, where $\xi \in [1, 2]$. Furthermore, we conjectured that the scheme is still secure even when $\xi = 2$.

Moreover, since our lattice is not public, the security of the squashed scheme does not rely on one instance of the Subset Sum Problem (SSP), as in Gentry's original framework. Rather, we only require the attacker to be incapable of solving many different SSP instances simultaneously. This allows us to use exponentially smaller set, if the number of the instances is big. This feature, along with the smaller lattice dimension, gives us a very practical FHE scheme.

Finally, we also propose two sets of parameters with respect to $\xi = 1$ and $\xi = 2$. The first scenario delivers strong security by operating on lattices of large dimensions. In this construction, our scheme is less efficient than Gentry and Halevi's scheme because our decryption polynomial degree is greater than Gentry-Halevi's scheme due to the fact that the lattice is hidden. For the second scenario, we are able to operate on ideal lattices with a smaller dimension than usual, for instance, 31 or 63, compared to 2048 in Gentry and Halevi's scheme. This further improves the efficiency of our scheme.

II. Background

A. Notations

Let λ be the security parameter of the system, i.e., it takes at least 2^λ operations to break the system.

Denote upper case bold letters (such as $\mathbf{M}$) for matrices. Vectors will be denoted with lower case bold letters (i.e., $\boldsymbol{v} = \langle v_0, \ldots, v_{n-1} \rangle$), while v_i is the $(i+1)$-th element of $\boldsymbol{v}$. Polynomials will be denoted in italic (i.e., $f(x)$).

For integers z and d, denote $[z]_d$ for the reduction of z mod d within $(-d/2, d/2]$. For a rational number q, let $\lfloor q \rceil$ be the closest integer to q. These notations are extended to vectors in a natural way, i.e., for a rational vector $\boldsymbol{v} = \langle v_0, v_1, \ldots, v_{n-1} \rangle$, $\lfloor \boldsymbol{v} \rceil = \langle \lfloor v_0 \rceil, \lfloor v_1 \rceil, \ldots, \lfloor v_{n-1} \rceil \rangle$.

For a vector $\boldsymbol{v}$, denote $Poly(\boldsymbol{v})$ as its polynomial form, i.e. $Poly(\boldsymbol{v}) = \sum_{i=0}^{n-1} v_i x^i$. When it is unambiguous, we use $v(x)$. For a polynomial $f(x)$, denote $Vec(f(x))$ as its vector form. For a vector $\boldsymbol{v}$, or its polynomial form $v(x) = \sum_{i=0}^{n-1} v_i x^i$, and a polynomial $f(x)$, denote $Rot(\boldsymbol{v}, f)$ and $Rot(v(x), f)$ as the rotation matrix, where the i-th row of this matrix equals to the coefficients of $v \times x^{i-1} \bmod f$. We use $f(x) = x^n + 1$ to build our ring, which is the classic ring for ideal lattice, therefore, when it is not ambiguous, we use $Rot(\boldsymbol{v})$.

For two vectors $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$, denote $\boldsymbol{v}_1 \times \boldsymbol{v}_2$ as the polynomial multiplication over the ring, i.e., $\boldsymbol{v}_1 \times \boldsymbol{v}_2 = Vec(v_1(x) \times v_2(x) \bmod f(x))$.

B. Homomorphic Encryption

A homomorphic encryption scheme ξ consists of four algorithms: KeyGen, Encrypt, Decrypt and Evaluate. The

first three algorithms follow the definition of a public key encryption scheme, while the last one is defined as follows: input a public key pk, a set of ciphertexts $\{c_i\}$ whose corresponding messages are $\{m_i\}$, and a circuit C, output another ciphertext c. This evaluation is correct if the following holds:

$$\begin{aligned}\text{DECRYPT}(\text{EVALUATE}(C, \{c_i\}, pk), sk) \\ = C(m_1, \ldots, m_t).\end{aligned} \tag{1}$$

Definition 1 (Homomorphic Encryption): The scheme $\xi = (\text{KEYGEN}, \text{ENCRYPT}, \text{DECRYPT}, \text{EVALUATE})$ is homomorphic for a class $\mathcal{C}$ of circuits if it is correct according to Equation 1 for all circuits $C \in \mathcal{C}$. ξ is fully homomorphic if it is correct for all boolean circuits. Further, ξ is compact, if for any circuit $C \in \mathcal{C}$ with a number of inputs polynomial in λ, the size of ciphertexts output by EVALUATE is bounded by a fixed value which is polynomial in λ.

Definition 2 (Bootstrappable Encryption): Let scheme $\xi = (\text{KEYGEN}, \text{ENCRYPT}, \text{DECRYPT}, \text{EVALUATE})$ be a compact homomorphic encryption scheme, and let $\mathcal{C}_\xi$ be the class of circuits regarding to which the scheme is correct. Denote D_ξ its decryption circuit. ξ is bootstrappable if $D_\xi \in \mathcal{C}_\xi$.

Remark 1: Gentry has shown that if a bootstrappable scheme can correctly evaluate bitwise additions and multiplications over two ciphertexts, then this scheme is fully homomorphic [8].

C. Lattice Basics

In the next two subsections, we show the definition of lattice and related problems that will be used throughout the paper. We refer the readers to [20], [24] for a more complex account.

Definition 3 (Lattice): Let $\boldsymbol{v}_i \in \mathbb{R}^n$ be d linearly independent vectors. An d-dimensional lattice with respect to $\{\boldsymbol{v}_i\}$, denoted as $\mathcal{L}(\{\boldsymbol{v}_i\})$, is the set of all integer linear combinations of $\{\boldsymbol{v}_i\}$. The $\boldsymbol{v}_i$-s are called a basis of $\mathcal{L}$. The determinant of a lattice is defined as $\det(\mathcal{L}) = \sqrt{\mathbf{B} \times \mathbf{B}^T}$, where $\mathbf{B}$ is a basis of $\mathcal{L}$.

Definition 4 (Ideal Lattice [23]): Let R be a polynomial ring $\mathbb{Z}[X]/f$, where $f \in \mathbb{Z}[X]$ is a monic irreducible polynomial of degree n. Let $\boldsymbol{v} \in \mathbb{Z}^n$. The ideal lattice over R with respect to $\boldsymbol{v}$, denoted by $\mathcal{L}(Rot(v, f))$ is the set of all integer linear combinations of $\boldsymbol{v}$ and its rotation vectors.

Definition 5 (Norm): Let $\boldsymbol{v} = \langle v_0, \ldots, v_{n-1} \rangle \in \mathbb{R}^n$ be a vector. The Euclidean norm is the function $\| \cdot \|$, defined by $\|\boldsymbol{v}\| = \sqrt{\sum_{i=0}^{n-1} v_i^2}$.

For two vectors $\boldsymbol{v}_1, \boldsymbol{v}_2 \in \mathbb{R}^n$, $\|\boldsymbol{v}_1 + \boldsymbol{v}_2\| \leq \|\boldsymbol{v}_1\| + \|\boldsymbol{v}_2\|$, and $\|\boldsymbol{v}_1 \times \boldsymbol{v}_2\| \leq \theta \|\boldsymbol{v}_1\| \cdot \|\boldsymbol{v}_2\|$, where θ is a constant factor that depends on the polynomial ring R, i.e., when $f(x) = x^n + 1$, $\theta = \sqrt{n}$.

The distance between $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ is defined by $\text{dist}(\boldsymbol{v}_1, \boldsymbol{v}_2) = \|\boldsymbol{v}_1 - \boldsymbol{v}_2\|$. For a vector $\boldsymbol{v} \in \mathbb{R}^n$ and a lattice $\mathcal{L}$, the distance between the two, denoted by $\text{dist}(\boldsymbol{v}, \mathcal{L}) = \min(\|\boldsymbol{v} - \boldsymbol{u}\|), \forall \boldsymbol{u} \in \mathcal{L}$.

Definition 6 (Successive Minima): Let $\mathcal{L}$ be a lattice and i be an integer. The i-th successive minima, denoted by $\lambda_i(\mathcal{L})$ is the smallest real number such that there exist i non-zero linear independent vector $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_i \in \mathcal{L}$ satisfying $\|\boldsymbol{v}_1\|, \ldots, \|\boldsymbol{v}_i\| \leq \lambda_i(\mathcal{L})$.

For the relationship of minima and the determinant we have $\prod_{i=1}^{n} \lambda_i \leq n^{\frac{n}{2}} \det(\mathcal{L})$, hence, we obtain

$$\lambda_2 \leq (n^{\frac{n}{2}} \det(\mathcal{L})/\lambda_1)^{\frac{1}{n-1}}. \tag{2}$$

D. Cryptographic Hard Problem

Definition 7 (BDDP over (Ideal) Lattice): Let $\gamma \in \mathbb{R}^+$ be a positive real. Let $\mathcal{L}$ be an n dimensional (ideal) lattice, and $\boldsymbol{v} \in \mathbb{Z}^n$, such that there exists a unique vector $\boldsymbol{u} \in \mathcal{L}$ satisfying $dist(\boldsymbol{v}, \boldsymbol{u}) \leq \gamma$. The γ-Bounded Distance Decoding problem over (ideal) lattice, denoted by γ-BDD$_n$ (γ-BDDi$_n$, resp.), is to find $\boldsymbol{u}$, given a basis of $\mathcal{L}$ and $\boldsymbol{v}$.

Definition 8 (Dec BDDP over (Ideal) Lattice): Let $\gamma \in \mathbb{R}^+$ be a positive real. Let $\mathcal{L}$ be an n dimensional (ideal) lattice, and $\boldsymbol{v} \in \mathbb{Z}^n$. The Decisional γ-Bounded Distance Decoding problem over (ideal) lattice, denoted by Dec γ-BDD$_n$ (Dec γ-BDDi$_n$, resp.), is to decide if there exists a unique vector $\boldsymbol{u} \in \mathcal{L}$ satisfying $dist(\boldsymbol{v}, \boldsymbol{u}) \leq \gamma$ or not, given a basis of $\mathcal{L}$ and $\boldsymbol{v}$.

There have been several definitions of BDD (see [22], [12] for comparison), due to the simplification of their proof reductions. In our case, we subsequently define BDD with slight modification to achieve the same goal. Nevertheless, it is clear that all these definitions capture the same notion.

Definition 9 (AGCD Problem): Let $c_i \in \mathbb{Z}$, τ integers such that there exist some unique integers $r_i \in \mathbb{Z}$ and a unique integer $p \in \mathbb{N}$ such that $\forall i, (c_i - r_i)|p$ and $\forall i, |r_i| \leq \gamma < p/2$. Then, the Approximate Greatest Common Divisor problem, denoted by γ-AGCDτ, is to find p, given c_i.

The above definition describes the general version of AGCD problem. By setting $r_1 = 0$ one obtains the partial version (P-AGCD). For the rest of this paper, as mentioned earlier, we only concern with the general version of this problem. Subsequently, as in the case of BDD, the problem of AGCD has been defined slightly differently in the literature (see [33], [19] for comparison). Again, we apply the same principle to achieve our goal. Nevertheless, all of these definitions still capture the same notion.

The classic way to attack this problem makes use of lattices [33]. Nevertheless, there are also attacks like [5] which do not use lattices.

Definition 10 (Subset Sum Problem): Let $\{c_1, c_2, \ldots, c_n\}$ be a set of positive integers. Let $c = \sum_{i=1}^{n} s_i c_i$, where $s_i \in \{0, 1\}$. Let $d \leftarrow \sum_{i=1}^{n} s_i$. The subset sum problem, denoted by d, n-SSP, is to find $\{s_i\}$, given $\{c_i\}$ and c.

For completeness, we also list the subset sum problem used in the squashing technique. When $d \ll n$ it becomes a sparse subset sum problem (SSSP). We note that our scheme does not rely on the hardness of this problem.

III. Hidden Lattice

In this section, we formally define the new problems related to the hidden lattice.

Definition 11 (Hidden (Ideal) Lattice): Let $\alpha \in \mathbb{R}^+$ be a positive real, $\boldsymbol{v}_i \in \mathbb{Z}^n$ be τ integer vectors such that there exists a unique (ideal) lattice $\mathcal{L}$ and some unique vectors $\boldsymbol{w}_i \in \mathcal{L}$ respecting $\forall 1 \leq i \leq \tau$, $dist(\boldsymbol{v}_i, \boldsymbol{w}_i) \leq \alpha$. Then $\mathcal{L}$ is called an α-*Hidden* (*Ideal*) *Lattice* hidden under $\{\boldsymbol{v}_i\}$.

A. Definitions of New Problems

Definition 12 (Hidden (Ideal) Lattice Problem): Let $\alpha \in \mathbb{R}^+$ be a positive real, $\boldsymbol{v}_i \in \mathbb{Z}^n$ be τ integer vectors such that there exists an α-Hidden (Ideal) Lattice $\mathcal{L}$ hidden under $\{\boldsymbol{v}_i\}$. The α-Hidden (Ideal) Lattice Problem, denoted by α-HLP$_{n,\tau}$ (α-HILP$_{n,\tau}$, resp.), is to find $\mathcal{L}$, given $\{\boldsymbol{v}_i\}$.

Informally, the HLP/HILP is defined as follows: given some vectors close to a (ideal) lattice, find such a lattice.

Definition 13 (BDDP over Hidden (Ideal) Lattice): Let $\alpha, \beta \in \mathbb{R}^+$ be some positive reals. Let $\boldsymbol{v}_i \in \mathbb{Z}^n$ be τ integer vectors such that there exists an α-Hidden (Ideal) Lattice, $\mathcal{L}$, hidden under $\{\boldsymbol{v}_i\}$. Let $\boldsymbol{u} \in \mathbb{Z}^n$ be an integer vector such that there exists a unique $\boldsymbol{w} \in \mathcal{L}$ respecting $dist(\boldsymbol{u}, \boldsymbol{w}) \leq \beta$. Then the *Bounded Distance Decoding problem over Hidden*(*ideal*) *lattice*, denoted by α, β-BDDH$_{n,\tau}$ (α, β-BDDHi$_{n,\tau}$, resp.), is to find $\boldsymbol{w}$, given $\{\boldsymbol{v}_i\}$ and $\boldsymbol{u}$.

Definition 14 (Dec BDDP over Hidden (Ideal) Lattice): Let $\alpha, \beta \in \mathbb{R}^+$ be some positive reals. Let $\boldsymbol{v}_i \in \mathbb{Z}^n$ be τ integer vectors such that there exists an α-Hidden (Ideal) Lattice, $\mathcal{L}$, hidden under $\{\boldsymbol{v}_i\}$. Let $\boldsymbol{u} \in \mathbb{Z}^n$ be an integer vector. Then the Decisional *Bounded Distance Decoding problem over Hidden* (*ideal*) *lattice*, denoted by Dec α, β-BDDH$_{n,\tau}$ (Dec α, β-BDDHi$_{n,\tau}$, resp.), is to decide if there exists a unique $\boldsymbol{w} \in \mathcal{L}$ such that $dist(\boldsymbol{u}, \boldsymbol{w}) \leq \beta$ or not, given $\{\boldsymbol{v}_i\}$ and $\boldsymbol{u}$.

B. Reductions from Existing Problems

In this subsection, we provide reductions of our new problems from some existing problems. The relations among the problems is described in Figure 1, where an arrow from problem A to problem B means that A is harder to solve than B. We omit the proof of the reductions from problems over general lattice to problems over ideal lattice (i.e. HILP to HLP), since if an algorithm can solve the problem in any lattice, it can solve the problem in an ideal lattice.

In this work, we do not provide an average case/worst case equivalence. We note that this is not surprising, considering that both of the previous works that we generalized (due to Gentry and Halevi's scheme [12] and van Dijk et. al's scheme [33]) also do not provide such a proof. Nevertheless, Gentry also provided an average/worst case equivalent for BDD over ideal lattice [11], but this work was not adopted in the subsequent work in Gentry and Halevi's scheme due to its impracticality. Further, both of the schemes [12], [33] also rely on anther problem (SSSP), in which the proof for the average case/worst case equivalent has never been investigated yet, to the best of our knowledge.

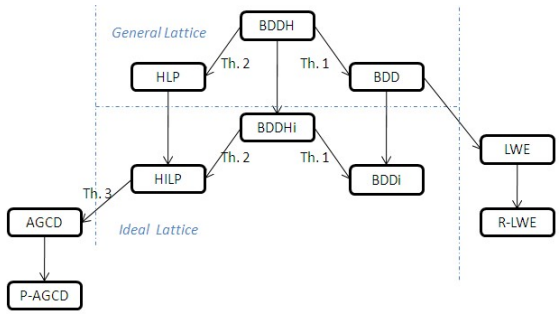

Fig. 1: Relations among problems.

Theorem 1: If an algorithm $\mathcal{A}$ solves α, β-BDDH$_{n,\tau}$ (α, β-BDDHi$_{n,\tau}$ resp.) with an advantage of ε, then there exist an algorithm $\mathcal{B}$ that solves the γ-BDD$_n$ (γ-BDDi$_n$ resp.) with an advantage of at least ε. The running time of $\mathcal{B}$ is polynomial in the running time of $\mathcal{A}$.

Proof: Let $\{\boldsymbol{v}_i\}, \boldsymbol{u}$ be the input of a γ-BDD$_n$ (γ-BDDi$_n$ resp.) problem, where $\{\boldsymbol{v}_i\}$ is a basis of $\mathcal{L}$. Set $\tau \leftarrow \#\boldsymbol{v}_i$, $\alpha \leftarrow 0$ and $\beta \leftarrow \gamma$. Call $\mathcal{A}$ with $\{\boldsymbol{v}_i\}$, $\boldsymbol{u}$. Since dist$(\boldsymbol{v}_i, \mathcal{L}) \leq \alpha$, and dist$(\boldsymbol{u}, \mathcal{L}) \leq \gamma$, $\{\boldsymbol{v}_1\}$ and $\boldsymbol{u}$ is in the correct form of the input of $\mathcal{A}$. Therefore, $\mathcal{A}$ returns the unique $\boldsymbol{w} \in \mathcal{L}$ such that $dist(\boldsymbol{u}, \boldsymbol{w}) \leq \beta = \gamma$. Return $\boldsymbol{w}$. ■

The above theorem is also correct for the decisional version of the problems. We omit the proof, which can be adapted in a similar fashion as above.

Theorem 2: If an algorithm $\mathcal{A}$ solves α, β-BDDH$_{n,\tau}$ (α, β-BDDHi$_{n,\tau}$, resp.) with an advantage of ε, then there exits an algorithm $\mathcal{B}$ that solves the α-HLP$_{n,\tau}$ (α-HILP$_{n,\tau}$, resp.) with an advantage of at least ε. The running time of $\mathcal{B}$ is polynomial in the running time of $\mathcal{A}$.

Proof: Let $\{\boldsymbol{v}_i\}$ be the input of an α-HLP$_{n,\tau}$ (α-HILP$_{n,\tau}$, resp.). Set $\beta \leftarrow \alpha$, $\tau \leftarrow \#\boldsymbol{v}_i$. For $1 \leq i \leq \tau$, set $\boldsymbol{u} \leftarrow \boldsymbol{v}_i$ and call $\mathcal{A}$ with $\{\boldsymbol{v}_i\}$, $\boldsymbol{u}$ to get the unique $\boldsymbol{w}_i \in \mathcal{L}$ where $\mathcal{L}$ is the α-Hidden (Ideal) Lattice hidden under $\{\boldsymbol{v}_i\}$, such that $dist(\boldsymbol{u}, \boldsymbol{w}_i) \leq \beta = \alpha$. Reconstruct the lattice $\mathcal{L}$ from the generating vectors $\{\boldsymbol{w}_i\}$. Return $\mathcal{L}$. ■

Theorem 3: If an algorithm $\mathcal{A}$ solves α-HILP$_{n,\tau}$ with an advantage of ε, then there exists an algorithm $\mathcal{B}$ that solves the γ-AGCD$_\tau$ with an advantage of at least ε. The running time of $\mathcal{B}$ is polynomial in the running time of $\mathcal{A}$.

Proof: Let c_i be the input of a γ-AGCD$_\tau$ problem. Set $n \leftarrow 1$, $\alpha \leftarrow \gamma$ and $\boldsymbol{v}_i \leftarrow \langle c_i \rangle$. Call $\mathcal{A}$ with $\{\boldsymbol{v}_i\}$ to get the unique $\mathcal{L}$ such that dist$(v_i, \mathcal{L}) \leq \alpha$. The basis of $\mathcal{L}$ is equal to the vector $\langle p \rangle$ such that $c_i = q_i p + r_i$ with $|r_i| \leq \alpha$. Return p. ■

IV. The SHE Scheme

The general idea of our work is to give some vectors close to the lattice, instead of giving directly the lattice, for enabling encryption. Only the secret key holder knows the lattice, and hence, can perform the correct decryption. The ciphertexts are vectors close to the lattice with a bounded distance, therefore we do not lose the property of homomorphism of the ciphertext.

A. Generic Construction

1) Parameters: The construction below uses the following parameters. Section VIII provides the concrete values for the parameters.

- ρ: the norm of the random noise vector;
- η: the bit length of the norm of generating polynomial;
- γ: the bit length of the norm of the random multiplier vector;
- τ: the number of vectors in the public key;
- ζ: the norm of noise used in encryption;
- n: the dimension of the hidden lattice.

2) The scheme: Our somewhat homomorphic encryption scheme uses following four algorithms:

KEYGEN(λ)

- Set parameters $\rho, \eta, \gamma, \tau, \zeta, n$ as in Section VIII with respect to λ, n is a power of 2;
- Pick an irreducible polynomial of degree n, $f(x) = x^n+1$ (see Remark 2);
- Pick a vector $\boldsymbol{v}$ randomly in $\{\boldsymbol{u} \in \mathbb{Z}^n, 2^{\eta-1} < \|\boldsymbol{u}\| < 2^{\eta}, \sum_{i=0}^{n-1} u_i \bmod 2 = 1\}$;
- Generate the rotation matrix $\mathbf{V} \leftarrow \mathit{Rot}(\boldsymbol{v}, f)$, i.e., when $f(x) = x^n + 1$,

$$\mathit{Rot}(\boldsymbol{v}, f) = \begin{vmatrix} v_0 & v_1 & v_2 & \dots & v_{n-1} \\ -v_{n-1} & v_0 & v_1 & \dots & v_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -v_1 & -v_2 & -v_3 & \dots & v_0 \end{vmatrix}.$$

- $d \leftarrow |\det(\mathbf{V})|$ is the determinant of $\mathbf{V}$ (see Remark 3);
- Pick $\tau-1$ vectors $\boldsymbol{g}_i$ randomly in $\{\boldsymbol{u} \in \mathbb{Z}^n, 2^{\gamma-1} < \|\boldsymbol{u}\| < 2^{\gamma}\}$ and another vector $\boldsymbol{g}_\tau$ randomly in $\{\boldsymbol{u} \in \mathbb{Z}^n, \|\boldsymbol{u}\| < 2^{\gamma}, \sum_{j=0}^{n-1} u_j \bmod 2 = 1\}$;
- Pick $\tau - 1$ vectors $\boldsymbol{r}_i$ randomly in $\{\boldsymbol{u} \in \{-1, 0, 1\}^n, \|\boldsymbol{u}\| \le \rho\}$ and another vector $\boldsymbol{r}_\tau$ randomly in $\{\boldsymbol{u} \in \{-1, 0, 1\}^n, \|\boldsymbol{u}\| \le \rho, \sum_{j=0}^{n-1} u_j \bmod 2 = 1\}$;
- Compute τ vectors $\boldsymbol{\pi}_i \leftarrow \boldsymbol{g}_i \times \boldsymbol{v} + \boldsymbol{r}_i$ for $1 \le i \le \tau$;
- Find the integer polynomial $w(x)$, such that $w(x) \times v(x) = d \bmod f(x)$ (see Remark 3), denote $\mathbf{W} \leftarrow \mathit{Rot}(\boldsymbol{w}, f)$;
- Output $sk \leftarrow \{d, \boldsymbol{w}\}$ and $pk \leftarrow \{\boldsymbol{\pi}_i\}$.

Remark 2: In [31], Smart and Vercauteren showed that one can use any irreducible polynomial for the ideal lattice to build FHE cryptosystems, however, Gentry and Halevi [12] restricted it to $x^n + 1$, where n is a power of 2, for enabling a faster operations. Recent result in [30] shows that n being a power of 2 is not essential, even if that leads to $x^n + 1$ being not irreducible. In our case, we adopt the setting from Gentry and Halevi's scheme in [12] to have fast operations.

Remark 3: The complexity of finding $w(x)$ depends on d. If d is prime, one can execute XGCD(v, f) to find w, where XGCD is the extended GCD algorithm [31]. However, this is only feasible on a small dimension. In a large dimension, having d to be prime is costly. One needs to use Gentry and Halevi's technique, where d needs to be *odd*.

ENCRYPT(m, pk)

- Pick $\tau + 1$ integer vectors $\{\boldsymbol{s}_1, \dots, \boldsymbol{s}_\tau, \boldsymbol{s}_{\tau+1}\}$ satisfying:
  - $\sum_{j=1}^{n} s_{i,j} \bmod 2 = 0$, $1 \le i \le \tau - 1$
  - $\sum_{j=1}^{n} s_{\tau,j} \bmod 2 = m$, $\sum_{j=1}^{n} s_{\tau+1,j} \bmod 2 = 0$;
  - Denote $\boldsymbol{s} \leftarrow \langle \boldsymbol{s}_1, \dots, \boldsymbol{s}_\tau, \boldsymbol{s}_{\tau+1} \rangle$, $\|\boldsymbol{s}\| \le \zeta$.
- Output $\boldsymbol{\psi} \leftarrow \sum_{i=1}^{\tau} \boldsymbol{s}_i \times \boldsymbol{\pi}_i + \boldsymbol{s}_{\tau+1}$

DECRYPT($\boldsymbol{\psi}$, sk)

- $\boldsymbol{\psi}' \leftarrow \lfloor \boldsymbol{\psi} \times \boldsymbol{w}/d \rceil$;
- Return $m \leftarrow \psi'(1) \bmod 2$.

EVALUATE ($\psi_1, \dots, \psi_t, C, pk$)

- For each addition or multiplication gate in C, call ADD or MULT algorithm;
- Return the output of C.

ADD($\boldsymbol{\psi}_1$, $\boldsymbol{\psi}_2$)

- Return $\psi \leftarrow \boldsymbol{\psi}_1 + \boldsymbol{\psi}_2$.

MULT($\boldsymbol{\psi}_1$, $\boldsymbol{\psi}_2$)

- Return $\boldsymbol{\psi} \leftarrow \boldsymbol{\psi}_1 \times \boldsymbol{\psi}_2$.

B. Correctness

Below we review two definitions provided by Gentry's that will be used in our proof.

Definition 15 (r_{Enc}): r_{Enc} represents the maximum possible distance between a ciphertext $\boldsymbol{\psi}$ generated by ENCRYPT algorithm and the hidden lattice $\mathcal{L}$.

Definition 16 (r_{Dec}): r_{Dec} represents the decryption radius: the minimum distance such that any $\boldsymbol{\psi}$ (generated by ENCRYPT or EVALUATE) can be decrypted correctly, if dist$(\boldsymbol{\psi}, \mathcal{L}) \le r_{Dec}$.

These definitions are also applicable to r_{pk}, i.e., the maximum distance between a public key $\boldsymbol{\pi}_i$ and the hidden lattice. As per definition, we have $r_{pk} = \rho$. Our noise of a ciphertext comes from the production of $\boldsymbol{s}$ and $\boldsymbol{r}_i$. Hence, we have $r_{Enc} \le \theta\rho\zeta$.[1] Meanwhile, the result of [12] shows that $r_{Dec} \sim 2^{\eta}$. So we prove the following under the assumption that $\theta\rho\zeta \ll 2^{\eta}$.

We firstly show the correctness of the DECRYPT algorithm. Essentially, any ciphertext $\boldsymbol{\psi}$ is a vector close to $\mathcal{L}(\mathbf{V})$, and can be decrypted correctly as long as $r_{Enc} < r_{Dec}$. Without losing generality, we assume $\boldsymbol{\psi} = \boldsymbol{a} + \boldsymbol{b}$, for certain $\boldsymbol{a} \in \mathbb{Z}^n$, $\boldsymbol{b} \in \mathcal{L}$, $\|\boldsymbol{a}\| \le \theta\rho\zeta$, where $\boldsymbol{a} = \sum_{i=1}^{\tau} \boldsymbol{r}_i \times \boldsymbol{s}_i + \boldsymbol{s}_{\tau+1}$. We firstly prove $\boldsymbol{\psi}' = \boldsymbol{a} \pmod 2$ as follows: Because $\boldsymbol{b} \in \mathcal{L}$, hence, $\boldsymbol{a} = \boldsymbol{\psi} \bmod \mathbf{V} = \boldsymbol{\psi} - \lfloor \boldsymbol{\psi} \cdot \mathbf{V}^{-1} \rceil \cdot \mathbf{V}$. Since $\mathbf{V}^{-1} = \mathbf{W}/d$, and $\mathcal{L}(\mathbf{V})$ and $\mathcal{L}(\langle 2 \rangle)$ are co-prime, we obtain $\boldsymbol{a} \bmod 2 = \lfloor \boldsymbol{\psi} \cdot \mathbf{W}/d \rceil \bmod 2 = \lfloor \boldsymbol{\psi} \times \boldsymbol{w}/d \rceil \bmod 2$. Therefore, $\boldsymbol{\psi}' \bmod 2 = \sum_{i=1}^{\tau} \boldsymbol{r}_i \times \boldsymbol{s}_i + \boldsymbol{s}_{\tau+1} \bmod 2$. Then we show $\psi'(1) \bmod 2 = m$ as follows: $\psi'(1) \bmod 2 = \sum_{i=1}^{\tau} r_i(1)s_i(1) + s_{\tau+1}(1) \bmod 2 = r_\tau(1)s_\tau(1) = m \pmod 2$. Therefore, the DECRYPT algorithm is correct.

Then we prove the correctness of MULT. We omit the proof of ADD. Assume $\psi_1 = \boldsymbol{a}_1 + \boldsymbol{b}_1$ and $\psi_1 = \boldsymbol{a}_2 + \boldsymbol{b}_2$ for certain $\boldsymbol{a}_1, \boldsymbol{a}_2 \in \mathbb{Z}^n$, $\boldsymbol{b}_1, \boldsymbol{b}_2 \in \mathcal{L}$, $\|\boldsymbol{a}_1\|, \|\boldsymbol{a}_2\| \le \theta\rho\zeta$, where $a_j(x) = \sum_{i=1}^{\tau} r_{j,i}(x)s_{j,i}(x) + s_{j,\tau+1}(x) \bmod f(x)$. Hence, $\psi(x) \leftarrow \psi_1(x) \times \psi_2(x) \bmod f(x) = a_1(x)a_2(x) + a_1(x)b_2(x) +$

[1] This value is indeed an upper bound. The actual r_{Enc} is expected to be much smaller. We provide more details in Section VIII.

$a_2(x)b_1(x) + b_1(x)b_2(x) \bmod f(x)$. Since the vector form of $\psi(x) - a_1(x)a_2(x)$ is in $\mathcal{L}$, as long as $\|a_1(x)a_2(x)\| < r_{Dec}$, decrypting $\boldsymbol{\psi}$ will return $a_1(1)a_2(1) = m_1 \times m_2$. Hence, MULT is correct.

C. Comparison of the SHE with other schemes

We note that our SHE scheme is a generalization of the integer based scheme [33] as well as the ideal lattice based scheme [8].

By setting $\tau = 1$ and let $\boldsymbol{r}_i = 0$, we obtain the ideal lattice based scheme. In this case, the lattice is not hidden anymore because the noise is zero. The public key of our scheme is a vector of the lattice, and the rotation of the vector forms a bad basis of the lattice, which will be used in the lattice based scheme. The rest of the construction follows the ideal lattice based scheme.

To obtain the integer based scheme, we set $n = 1$. Then the hidden lattice is a lattice with dimension 1. Its determinant is the secret key p used in the integer based scheme, while the public keys are τ number of integers $g_i p + r_i$.

We note that the LWE based schemes are still by far the most efficient ones. As mentioned earlier, it allows one to change the determinant of the lattice (modulus switching), and therefore, the growth of the noise is much slower than the lattice based scheme. Hence, essentially this technique boosts the system exponentially. Unfortunately, this technique cannot be directly applied to the lattice-based scheme. Therefore we omit the comparison with the LWE-based FHE schemes.

V. SEMANTIC SECURITY

The semantic security [17] of the scheme is defined as follows. [2]

Definition 17 (Semantic Security): The semantic security model is defined as follows:

1) The challenger runs KEYGEN algorithm and outputs a secret key sk and a public key pk;
2) The attacker is given an encryption oracle that computes the functionality ENCRYPT(m,pk);
3) The attacker then generates two ciphertexts m_0 and m_1;
4) The challenger generates $c = \text{ENCRYPT}(m_b, pk)$, where $b \in \{0, 1\}$;
5) The attacker outputs b'.

An encryption scheme is semantically secure if the advantage of the attacker to win the game $(Pr[b = b'] - 1/2)$ is negligible.

Theorem 4: If an algorithm $\mathcal{A}$ breaks the semantic security with advantage ε, then there exist an algorithm $\mathcal{B}$ that solves the Dec α, β-BDDHi$_{n,\tau}$ with advantage of $\frac{\varepsilon}{8}$. The running time of $\mathcal{B}$ is polynomial in the running time of $\mathcal{A}$.

Proof: Fix parameters $\rho, \eta, \gamma, \tau, \zeta, n$ as in KEYGEN. Set $\alpha \leftarrow \rho$, $\beta \leftarrow \theta\zeta\rho$. Let $\{\boldsymbol{v}_i\}$ and $\boldsymbol{u}$ a decisional α, β-BDDHi$_{n,\tau}$ problem. Assume $\boldsymbol{v}_i = \boldsymbol{g}_i\mathbf{B} + \boldsymbol{r}_i$, where $\mathbf{B}$ is a basis of $\mathcal{L}$, $\|\boldsymbol{r}_i\| \leq \alpha$, $\boldsymbol{g}_i, \boldsymbol{r}_i \in \mathbb{Z}^n$. Call algorithm $\mathcal{A}$ with $m_b, b \in \{0, 1\}$ and $\{\boldsymbol{v}_i\}$. For $\boldsymbol{v}_\tau$, it has one chance on four that $\sum_{j=0}^{n} r_{i,j} \bmod 2 = 1$ and $\sum_{j=0}^{n} g_{i,j} \bmod 2 = 1$. We do not have parity requirement for $\sum_{j=0}^{n} r_{i,j} \bmod 2$ and $\sum_{j=0}^{n} g_{i,j} \bmod 2$ when $i \neq \tau$. Therefore, the input vectors have at least $\frac{1}{4}$ probability of being in the correct form of the public key if $\det(\{\boldsymbol{v}_i\})$ is odd. As a result, the overall probability is $\frac{1}{8}$.

Algorithm $\mathcal{A}$ then returns m_b^*. Algorithm $\mathcal{B}$ outputs $m_b^* + m_b + 1$. If the vectors are in the correct form of the public keys (meaning that for $\boldsymbol{v}_\tau$, we require $\sum_{j=0}^{n} r_{\tau,j} = 1$ and $\sum_{j=0}^{n} g_{\tau,j} = 1$), $m_b^* = m_b$ with a probability of $\frac{1}{2} + \varepsilon$. Meanwhile, if $\{\boldsymbol{v}_i\}$ is not in the correct form, $\mathcal{A}$ will return a random bits. Which implies the probability of $m_b^* = m_b$ is $1/2$. Overall, $\mathcal{B}$ has an probability of $\frac{1}{8}(\frac{1}{2} + \varepsilon) + \frac{7}{8} \cdot \frac{1}{2} = \frac{1}{2} + \frac{\varepsilon}{8}$ to obtain correct result. Hence, the overall advantage is $\frac{\varepsilon}{8}$. ■

VI. ATTACKS

A. Practical Public Key Attack

In this section, we present the best known attack, which is an adaptation of the attack proposed in [33] to solve the AGCD of k integers. We simply generalize the attack from a lattice dimension equal to 1 to any n. We note that the attack in [33] was already a generalization of the AGCD$_k$ attack in [19] from 2 integers to any number of integers.

$$\mathbf{B} = \begin{vmatrix} Id(\theta\rho) & Rot(\boldsymbol{\pi}_2) & Rot(\boldsymbol{\pi}_3) & \dots & Rot(\boldsymbol{\pi}_k) \\ 0 & Rot(\boldsymbol{\pi}_1) & 0 & \dots & 0 \\ 0 & 0 & Rot(\boldsymbol{\pi}_1) & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & Rot(\boldsymbol{\pi}_1) \end{vmatrix}$$

Let $\mathcal{L}(\mathbf{B})$ be a lattice with a basis matrix with $k \leq \tau$ public keys as above. $Id(a) = a \cdot \mathbf{I}_n$ where $\mathbf{I}_n$ is an $n \times n$ identity matrix. For $\boldsymbol{\pi}_1 = \boldsymbol{r}_1 + \boldsymbol{g}_1\mathbf{V}$ and any of $\boldsymbol{\pi}_i = \boldsymbol{r}_i + \boldsymbol{g}_i\mathbf{V}$ where $2 \leq i \leq k$, we have $\boldsymbol{\pi}_1\boldsymbol{g}_i - \boldsymbol{\pi}_i\boldsymbol{g}_1 = \boldsymbol{r}_1\boldsymbol{g}_i - \boldsymbol{r}_i\boldsymbol{g}_1$. Therefore, the lattice $\mathcal{L}(\mathbf{B})$ contains a vector $\boldsymbol{u}$ such that $\boldsymbol{u} = \langle \theta\rho\boldsymbol{g}_1, \boldsymbol{r}_2\boldsymbol{g}_1 - \boldsymbol{r}_1\boldsymbol{g}_2, \boldsymbol{r}_3\boldsymbol{g}_1 - \boldsymbol{r}_1\boldsymbol{g}_3, \dots, \boldsymbol{r}_k\boldsymbol{g}_1 - \boldsymbol{r}_1\boldsymbol{g}_k \rangle$. Finding such a vector breaks the public key security, since one can recover $\boldsymbol{r}_1$ form $\boldsymbol{g}_1$ and $\boldsymbol{\pi}_1$.

In lattice reductions, it has been shown that the best lattice reduction algorithm cannot find $\boldsymbol{u}$ if $\frac{\lambda_2(\mathcal{L}(\mathbf{B}))}{\|\boldsymbol{u}\|} < c^{nk}$ for some constant c. A recent work in [4] shows that the smallest c that a reduction algorithm is reachable is 1.009.[3]

Since $\lambda_2(\mathcal{L}) \leq \sqrt{nk}^{\frac{nk}{nk-1}} (\frac{\det(\mathcal{L})}{\|\boldsymbol{u}\|})^{\frac{1}{nk-1}}$ (see Equation 2), we obtain that $\boldsymbol{u}$ should not be found using a lattice reduction if $\sqrt{nk}^{\frac{nk}{nk-1}} \det(\mathbf{B})^{\frac{1}{nk-1}} < c^{nk} \|\boldsymbol{u}\|^{\frac{nk}{nk-1}}$. Therefore, we need to guarantee that $\det(\mathbf{B}) < c^{nk(nk-1)} \|\boldsymbol{u}\|^{nk}$. We know that $\det(\mathbf{B}) = \rho^n \det(Rot(\boldsymbol{\pi}_1))^{k-1}$. Using the Hadamard upper bound [18], we obtain $\det(Rot(\boldsymbol{\pi}_1)) \leq \|\boldsymbol{\pi}_1\|^n$, therefore, we have $\det(\mathbf{B}) \leq \rho^n \|\boldsymbol{\pi}_1\|^{n(k-1)} \leq \rho^n (\theta \|\boldsymbol{g}_1\| \|\boldsymbol{v}\|)^{n(k-1)}$.

Meanwhile, we have $\|\boldsymbol{u}\| > \theta\rho\|\boldsymbol{g}_1\|$, and therefore we can expect the attack to be successful if $\rho^n (\theta \|\boldsymbol{g}_1\| \|\boldsymbol{v}\|)^{n(k-1)} \geq c^{nk(nk-1)} (\theta\rho\|\boldsymbol{g}_1\|)^{nk}$. To relax the condition a bit further,

[2] As in Gentry's work [8], the definition is without the reference to the EVALUATE algorithm.

[3] In [4], the authors also provided an enumeration technique, that allows one to obtain $c = 1.009$ with 2^{35} operations, however, the enumeration technique only works for small dimensions, while in our case the dimension is approximately τn. Hence, enumeration technique is not applicable in our scenario.

the attack will be successful if $\rho^n(\theta\|\boldsymbol{g}_1\|\|\boldsymbol{v}\|)^{n(k-1)} \geq c^{n^2k^2}(\theta\rho\|\boldsymbol{g}_1\|)^{nk}$. As a result, we have the following equation:

$$\eta(k-1) \geq \log_2\theta + nk^2\log_2 c + \gamma + (k-1)\log_2\rho, \quad (3)$$

which is

$$\log_2 c \leq -\frac{\gamma + \log_2\theta + (k-1)(\log_2\rho - \eta)}{nk^2}.$$

To allow the best advantage of the attacker (where the attacker can use c as great as possible), we need to maximize the right hand side of the inequation. Denote A the right hand side of the inequation, and let $\kappa = \frac{1}{k}$, then

$$A = -\frac{\gamma + \log_2\theta + \log_2\rho - \eta}{n}\kappa^2 - \frac{\log_2\rho - \eta}{n}\kappa$$

Since the coefficient of κ^2 term is negative (because $\gamma > \eta$), the maximum value of A is achieved when

$$\kappa = \frac{\eta - \log_2\rho}{2(\gamma + \log_2\theta + \log_2\rho - \eta)},$$

which is

$$k = \frac{2(\gamma + \log_2\theta + \log_2\rho - \eta)}{\eta - \log_2\rho}.$$

Considering that $\log_2\rho$ is neglible compared with η, hence we know that the best attack occurs when $k \sim O(\frac{\gamma}{\eta})$.

Remark 4: The best attack occurs when $O(\frac{\gamma}{\eta})$ public keys are used. This is also the case with the AGCD attack against the integer based fully homomorphic encryption scheme.

To sum up, we need Equation 3 to be false for all k between 2 and τ. Moreover, in our parameter settings, we have $\frac{\gamma}{\eta} > \tau$. Hence, to allow the greatest advantage to the attacker, he/she should use all public keys in this attack. Therefore, taking $\gamma \sim O(n\eta)$ and $\tau \sim O(n)$, we conjecture that the best attack requires a lattice dimension that is quadratic with n.

B. Best Known Message Attack

The best known message attack come from the idea of attacking a GGH type cryptosystem in [25], i.e., converting a BDD problem into finding the shortest vector in a certain lattice.

Let $\boldsymbol{u} \leftarrow \langle 1, \boldsymbol{s}_1, \ldots, \boldsymbol{s}_\tau, \boldsymbol{s}_{\tau+1}\rangle$.We know that $\boldsymbol{u}$ is the shortest vector in the lattice whose basis is shown as follows:

$$\mathbf{B} = \begin{vmatrix} 1 & 0 & 0 & 0 & \ldots & 0 & \boldsymbol{\psi} \\ 0 & I_n & 0 & 0 & \ldots & 0 & Rot(\boldsymbol{\pi}_1) \\ 0 & 0 & I_n & 0 & \ldots & 0 & Rot(\boldsymbol{\pi}_2) \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \ldots & I_n & Rot(\boldsymbol{\pi}_\tau) \end{vmatrix}$$

Therefore, using an equivalent estimation as in the best known public key attack, we estimate that no lattice reduction will be able to find $\boldsymbol{u}$ from the lattice if the following equation holds:

$$\log_2\theta + \gamma + \eta < \tau(n\tau - 1)\log_2 c + \tau\log_2\zeta. \quad (4)$$

It is worthing pointing out that for the message attack, the attacker is obliged to put all the public key in the lattice, which implies that the dimension of the attacking lattice is strictly $n\tau + 1$, which is slightly different from the public key attack.

C. Other Attacks

1) Birthday Paradox on Public Keys: For two public keys $\boldsymbol{\pi}_i$ and $\boldsymbol{\pi}_j$, one can guess the corresponding noise $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$, and construct two lattices $\mathcal{L}_1 \leftarrow \mathcal{L}(Rot(\boldsymbol{\pi}_i - \boldsymbol{r}_i))$ and $\mathcal{L}_2 \leftarrow \mathcal{L}(Rot(\boldsymbol{\pi}_j - \boldsymbol{r}_j))$. A collision will be found when $\mathcal{L}_1 = \mathcal{L}_2$, which implies $\mathcal{L}_1 = \mathcal{L}_2$ is the hidden lattice. To stop the birthday paradox attack, it requires that the number of possible $\boldsymbol{r}_i$ in a single public key is greater than $2^{\lambda/2}$.

2) Brute Force on Ciphertext: The ciphertext is protected by the noise $\boldsymbol{s}$. To attack the ciphertext, one guesses $\boldsymbol{s}$. Therefore, our scheme requires that the number of possible $\boldsymbol{s}$ to be greater than 2^λ.

3) AGCD Attack: There is also another AGCD attack in [7], which is a generalization of the partial AGCD attack presented in [5]. For two integers $c_1 = g_1p + r_1$ and $c_2 = g_2p + r_2$, the attack is to find the greatest common divisor of $\prod_{i=0}^{2^\gamma-1}(c_1 - i)$ and $\prod_{i=0}^{2^\gamma-1}(c_2 - i)$, where γ is the maximum bit-length of r_i. Then, one can recover p from this divisor. This attack runs in $O(2^\gamma)$ time. We note that this attack is not directly applicable to our scheme, since we are dealing with vectors instead of integers. The best adaption of this attack is to replace c_i with $\det(Rot(\pi_i))$, and guess r_i with all possible noise. Nevertheless, this attack will be no better than a birthday paradox attack, due to the extra cost of computing the product of the determinant.

D. Security Conjecture1

In Subsection III-B, we have shown that a $\text{BDDH}_{n,\tau}$ is harder than a BDD_n problem. In the previous part of this section we also show that if there exists a $\text{BDD}_{O(n\tau)}$ solver, then one is able to solve $\text{BDDH}_{n,\tau}$. Generally speaking, we have the following relations between these problem, assuming $\tau \geq n$,

$$BDD_n \geq BDDH_{n,\tau} \geq BDD_{O(n^2)},$$

where $A \geq B$ means B is harder than A.

Thus, we know that $\text{BDDH}_{n,\tau}$ is equivalent to a $\text{BDD}_{O(n^\xi)}$ where $1 \leq \xi \leq 2$, and our construction is based on $\text{BDD}_{O(n^\xi)}$.

Conjecure 1: If a $\text{BDD}_{O(n^2)}$ problem is secure, then $\text{BDDH}_{n,\tau}$ is also secure, if $\tau \geq n$.

In Section VIII we will present two parameter settings. We note that the first parameter setting does not rely on this security conjecture. However, using this conjuecture will further improve the performance of the scheme as will be demonstrated in the second part of our parameter setting.

VII. Bootstrapping

In this section, we show how to bootstrap our scheme. We firstly squash the decryption algorithm to obtain a low degree decryption polynomial. Then we show that our cryptosystem is able to evaluate this polynomial homomorphically.

A. Squashed scheme

In order to bootstrap our scheme, we adopt the squashing technique used in Gentry's scheme. Essentially, we need to evaluate $\boldsymbol{\psi} \times Rot(\boldsymbol{w})$. The multiplication circuit in the

decryption algorithm is squashed into several additions. The squashed scheme takes as follows:

KEYGEN$^*(pk, sk)$

- Generate a vector $\boldsymbol{w}^* = \langle w_1^*, \ldots, w_n^* \rangle$, where $w_i^* \leftarrow \lceil 2^{\eta^*} \times w_i/d \rfloor$, and $\eta^* = \eta + \gamma + 1$ is an integer.
- For each coefficient w_i^* of vector $\boldsymbol{w}^* = \langle w_1^*, \ldots, w_n^* \rangle$, generate an l dimentional integer vector $\boldsymbol{y}_i = \langle y_{i,1}, \ldots, y_{i,l} \rangle$ and a binary vector $\boldsymbol{z}_i = \langle z_{i,1}, \ldots, z_{i,l} \rangle$, such that $w_i^* = \sum_{k=1}^{l} y_{k,i} z_{k,i}$, and the hamming weight of $\boldsymbol{z}_i$ is t.
- Repeat the last step for all coefficients of $\boldsymbol{w}^*$;
- Output $pk^* \leftarrow \{\boldsymbol{y}_k\}_{k=0}^{n-1}$ and $sk^* \leftarrow \{\boldsymbol{z}_k\}_{k=0}^{n-1}$.

Remark 5: In our parameter setting, we use $t = 1$, which indicates that one of coefficients of $\boldsymbol{y}_i = \langle y_{i,1}, \ldots, y_{i,l} \rangle$ is w_i^*. To do so, one can randomly assign w_i^* to one of the coefficient of $\boldsymbol{y}_i$ and fill in the rest of the coefficients with approximately same length.

SQUASH$(\boldsymbol{\psi}, pk^*)$

- Given a ciphertext $\boldsymbol{\psi} = \langle \psi_0, \ldots, \psi_{n-1} \rangle$, for each coefficient, generate $\boldsymbol{x}_i = \psi_i \boldsymbol{y}_i / 2^{\eta^*}$ for $0 \le i \le n-1$, and keep ω precisions behind the decimal point;
- Output $\{\boldsymbol{x}_i\}_{i=1}^{n}$.

DECRYPT$^*(\{\boldsymbol{x}_i\}, sk^*)$

- $\psi_i^* \leftarrow [\boldsymbol{x}_i \boldsymbol{z}_i]$;
- $\boldsymbol{\psi}^* \leftarrow \langle \psi_1^*, \ldots, \psi_n^* \rangle$;
- $\boldsymbol{\psi}' \leftarrow$ row sum of the matrix $Rot(\boldsymbol{\psi}^*)$
- Output $m \leftarrow \psi'(1) \bmod 2$.

1) Correctness: Our proof takes two stages. We firstly prove that the decryption is correct under the assumption that the roundoff will not introduce errors. Under this assumption, we have $\psi_i^* = \boldsymbol{x}_i \boldsymbol{z}_i = \psi_i \boldsymbol{y}_i \boldsymbol{z}_i / 2^{\eta^*} = \psi_i w_i^* / 2^{\eta^*} = \psi_i w_i / d$. Hence, the decryption is correct.

Now we show the requirement of our assumption. For the i-th coefficient, let $\Delta \leftarrow w_i^* - 2^{\eta^*} \times w_i/d$, then $w_i^* = \Delta + 2^{\eta^*} w_i/d$. Therefore $\psi_i^* = [\psi_i \times w_i^* / 2^{\eta^*}] = [\psi_i \times \Delta / 2^{\eta^*} + \psi_i \times w_i/d]$. The first term needs to be smaller than $1/2$. Since $\Delta < 1/2$ by definition, setting $2^{\eta^*} > \psi_i$ for all $i < n$ will guarantee there is no error in the decryption. As a result, our decryption algorithm is correct when $\eta^* \ge \eta + \gamma + 1$.

2) Security of the Squashed Scheme: On a high level, for each coefficient of the secret key w_i^*, $0 \le i \le n-1$, we squash it into a subset with l elements. To recover the secret key, an attacker is required to recover all w_i'-s. This is mainly due to the hidden ideal lattice we used. Since the attacker does not know the lattice, it is incapable of verifying if the w_i^* it recovered is correct. The only method that can be conducted is to recover all w_i^*-s and then to use them to decrypt a certain ciphertext and check the decryption correctness. As a result, the security of our squashed scheme is $f(t,l)^n$, where $f(t,l)$ is the number of operations that required to solve a t, l-SSP, which equals to $\binom{l}{t}$ for a small set.

Conjecure 2: The best attack to solve n different t, l-SSP is via a brute force attack. The complexity is $\binom{l}{t}^n$.

In fact, as stated earlier, this is another main advantage of hiding the lattice other than using smaller dimensions. In contrast, in Gentry and Halevi's implementation, one is required to provide security for each SSP. If one have recovered one coefficient, it is able to recover the whole secret key. This resulted into a big set of at least 1024 element for each w_i^* that has been adopted.

As we shall see in the next section, since we have increased the security by an exponential factor of n, we are able to use exponentially smaller sets (in terms of number of elements). For instance, we squash w_i^* into a set of only 6 elements, with only one of them being w_i^*. The attacker will have to decide which one out of the six is picked. As a result we obtain a security of 6^n.

However, for each coefficient of $\boldsymbol{\psi}'$, we will need to perform $t \times n$ additions, compared with only t additions in Gentry and Halevi's scheme. In this case, since the decryption is additions of $t \times n$ floating points, we need $\log_2(t \times n) + 1$ digits precision after the decimal point, as shown in Table I. As a result, the degree of decryption polynomial, denoted by q, is increased.

B. Bootstrapability

The bootstrapability of the squashed scheme depends on the degree of the binary form of the decryption polynomial. The result of [12] shows that to evaluate m monomials with a degree q polynomial homomorphically, the noise of the resulting ciphertext is around $\sqrt{m}(r_{Enc})^q$. Since we have previously shown that $r_{Enc} \le \theta\rho\zeta$, now we evaluate the number of monomials.

Since our decryption algorithm has essentially the same structure with Gentry and Halevi's scheme, the following result of the squashed the decryption follows their observation. The number of degree-ℓ monomials in the squashed decryption algorithm is

$$m = \prod_{i=0}^{\lfloor \log_2 \ell \rfloor} \binom{\ell}{2^i}.$$

For instance, if $\ell = 31$, then $m = \binom{31}{1}\binom{31}{2}\binom{31}{4}\binom{31}{8}\binom{31}{16} \sim 2^{75}$.

Then the squashed decryption requires a multiplication between x_i and z_i. However, it is not necessary to encrypt x_i. Therefore, this multiplication does not increase the number of monomials. Finally, since we need to support a product of two homomorphically-decrypted bits, our scheme must support polynomials with m degree-ℓ monomials. As a result, our scheme is expected to be able to handle a homomorphic decryption plus one more multiplication/addition if Equation 5 holds.

$$2^{\eta} \ge \sqrt{m}(\theta\rho\zeta)^{\ell}. \tag{5}$$

VIII. PARAMETERS

In this section, we provide two sets of parameters. In the first set we have $\xi = 1$. In this case, we provide a theoretical result where We do not rely on the security conjecture 1. In this case, we work on hidden lattices which dimension are quite large, and as a result, we only need a constant number of public

Number of Additions ($t \times n$)	Precisions of Floating Point (ω)	Degree of Decryption Polynomial (q)	Number of Monomials (m)
$2 \sim 3$	2	3	9
$4 \sim 7$	3	7	5145
$8 \sim 15$	4	15	$\sim 2^{34}$
$16 \sim 31$	5	31	$\sim 2^{75}$
$32 \sim 63$	6	63	$\sim 2^{176}$
$\dots$	$\dots$	$\dots$	$\dots$
$512 \sim 1023$	10	1023	$\sim 2^{3180}$

TABLE I: Relations between the # Additions and the Decryption Polynomial

keys. Essentially it is quite close to Gentry-Halevi's scheme. The major difference is that our lattice is hidden, so we do not rely on the SSP.

However, since the lattice is hidden, we need to do n times more additions compared with Gentry-Halevi's scheme, which increases our circuit depth dramatically, and makes our scheme impractical. We remark that it is natural that our scheme is less efficient, since our scheme is based on a harder problem, where the dimension of the lattice is remaining the same with Gentry-Halevi's scheme.

For the second set, we rely on our conjectures. We work on hidden lattices with small dimensions, and the number of public keys is approximately the same as the dimension. In the configuration, we use $c = 1.007$, which implies that no efficient reduction algorithm should be able to find vectors smaller than $c^n \det^{\frac{1}{n}}$ with sufficiently large n. Nevertheless, one should adapt c with the development of new reduction algorithms.

A. Parameters without relying on Conjecture 1

From subsection III-B it is straightforward to see that the BDDH problem is harder than the corresponding BDD problem over the lattices with the same dimension for the same parameters. Therefore, we propose a set of parameters of our FHE scheme where even to solve the BDD problem is hard.

Similar to [12], we do not provide an asymptotic complexity. Table II highlights the parameters for security levels 2^{80}. Since our parameters are bounded by several requirements, and most parameters are interconnected, we tested all the possible secure combinations to achieve the smallest public key size.

We use $\lambda = 80$ as an example. It requires a lattice with dimension 1024, with 2 public keys. To stop the birthday paradox attack, we allow 5 coefficients of each noise for each π_i to be -1, or 1, while the rest are 0. Hence, the maximum norm of the noise is $\sqrt{5}$. This allows $\binom{1024}{5} > 2^{\lambda/2}$ possibilities. As for the security of the encryption noise $\boldsymbol{s}$, $\boldsymbol{s}$ maintains $\tau + 1$ blocks. Each block has n coefficients. Further, the parity of each block (i.e., the sum of all bits) follows the requirement of the ENCRYPT algorithm. We set maximum 5 coefficients to be 1 or -1 (4 coefficients if the encrypted message is 0), with the rest 0-s. As a result, there are maximum 5 blocks with non-zero entries besides the τ-th block. The total possibility is at least $\binom{\tau+1}{5}(\binom{n}{2}2^2)^5 > 2^{80}$. Hence, we are secure against the brute force attack. We also use a $1, 2$-SSP setting to achieve a 2^{1024} security level of the squashed secret keys. This setting gives us a decryption polynomial of degree 3 as we stated eailer.

Now we look into more details of r_{Enc}. Recall that $\psi(x) = \sum_{i=1}^{\tau} s_i(x)\pi_i(x) + s_{\tau+1}(x) \bmod f(x)$. Therefore, the maximum noise of each ψ_i is 5, since $\boldsymbol{s}$ contains only 5 non-zero coefficients. As a result, the worst case $r_{Enc} = \sqrt{1024}\sqrt{5}\sqrt{5}$, although in most cases, it will be much smaller.

To bootstrap, we need $2^\eta \geq \sqrt{m}(r_{Enc})^{1023}$, where $m = \binom{1023}{1}\binom{1023}{2}\cdots\binom{1023}{512} \sim 2^{3180}$. Therefore, the setting $\eta = 9080$ will allow us to bootstrap. Having set all parameters as above, we choose the smallest γ allowed according to Equation 3 and 4. As a result, we obtain parameters in the second row of Table II. For the completeness, we also list the parameters for security level $\lambda = 128$ and 160.

B. Parameters assuming Conjecture 1

As for the conjectured security, we assume that the best known attack works on a lattice which dimension is quadratic with the hidden lattice. Therefore, we build our hidden lattice which dimension is square root of the dimension where a normal lattice problem is secure. However, due to the fact that the dimension is smaller than the proved scheme, the noise in each vector has to be increased to deliver the same security.

We also use $\lambda = 80$ as an example. The parameters for security level $\lambda = 128$ and 160 are listed in Table II. For $\lambda = 80$ it requires a lattice with dimension 31, with 57 public keys. To stop the birthday paradox attack, we allow the coefficient of each noise for each π_i to be $-1, 0$, or 1, so the maximum norm of the noise is $\sqrt{32}$. This allows $3^{32} > 2^{\lambda/2}$ possibilities. Further, $\boldsymbol{s}$ maintains $\tau+1$ blocks. Each block has 32 coefficients. We set maximum 11 coefficients to be 1 or -1 (10 coefficients if the encrypted message is 0), with the rest 0-s. As a result, there are maximum 5 blocks with non-zero entries besides the τ-th block. The total possibility is at least $\binom{\tau+1}{5}(\binom{n}{2}2^2)^5 > 2^{80}$. Hence, we are secure against the brute force attack. We also use a $1, 6$-SSP setting to achieve a $6^{32} \sim 2^{82}$ security level of the squashed secret keys.

As for the r_{Enc}, the maximum noise of each ψ_i is 11, since $\boldsymbol{s}$ contains only 11 non-zero coefficients. As a result, the worst case $r_{Enc} = 11\sqrt{32}$. To bootstrap, it requires $2^\eta \geq \sqrt{2^{75}}(11\sqrt{31})^{31}$, which gives us $\eta = 222$. Finally, we choose the smallest γ allowed according to Equation 3 and 4. As a result, we obtain parameters in the third row of Table II.

C. Comparison of Parameters

Here we focus on the performance of our second parameter setting. We omit the comparison with parameter setting 1, as this setting is highly inefficient as stated earlier.

For our conjectured cryptosystem, our SHE scheme uses a ciphertext size of $(222+18255)\times 31 \sim 573$ kbits. The SHE also uses a public key space of 111×573 kbits ~ 63.6 Mbits. The squashed scheme uses a public key of size $6 \times 31 \times (222 + 18255)$ bits ~ 3.4 Mbits. Finally, one needs to encrypt the

	ρ	η	γ	τ	ζ	n	t	l	Conjecture 1
$\lambda = 80$	$\sqrt{5}$	9080	1280000	310	$\sqrt{5}$	1023	1	2	N
$\lambda = 80$	$\sqrt{10}$	222	18255	111	$\sqrt{11}$	31	1	6	Y
$\lambda = 128$	8	595	88411	301	$\sqrt{17}$	63	1	5	Y
$\lambda = 160$	8	604	91127	307	$\sqrt{21}$	63	1	6	Y

TABLE II: Parameters

	GH Scheme $\lambda = 72$	Our Parameter Setting 2 $\lambda = 80$
Lattice dimension	2048	31
Lattice determinant	$2^{780,000}$	$2^{573,000}$
Set size	1024	6
Subset size	15	1
Public key size (Mbits)	552	173.5
Ciphertext size (kbits)	780	573

TABLE III: Comparisons with Gentry and Halevi's implementation

squashed secret key, which further adds another $6 \times 31 \times 573$ kbits ~ 106.5 Mbits. As a result, our whole system uses a public key size of $63.6 + 3.4 + 106.5 \sim 173.5$ Mbits.

To compare with the van Dijk et al.'s scheme, it is almost straightforward to see that we are more efficient, since their scheme works with approx $2^{31.6}$ integers, each of the length $2^{31.6}$ bits, to obtain a security level of $\lambda = 80$. Therefore we mainly compare our scheme with Gentry and Halevi's implementation.

To compare with Gentry and Halevi's implementation, our first parameter setting uses similar parameters with their SHE scheme. Nevertheless, due to the fact that the lattice is hidden, we achieve a better bootstappability by removing the necessity of relying on the SSP problem. While for our conjectured version of the scheme, we improve the efficiency by both the reducing the dimension and removing the SSP. To complete this subsection, we list the parameters used in Gentry and Halevi's implementation with $\lambda = 72$ (see [12]). It is easy to see that ours is more efficient than Gentry and Halevi's scheme in terms of space. The running time of the system is mainly influenced by the size of the ciphertext and the squashed decryption polynomial (both degree and number of monomials). We also note that those parameters in our scheme are smaller than Gentry and Halevi's system, therefore, it is straightforward to see that the running time of our scheme is better.

IX. Conclusion

To date, the existing fully homomorphic encryption schemes are based on three family of problems. In this paper, we introduced a new family of lattice problems based on hidden ideal lattice that unifies the two of these families. We show that the ideal lattice version of those problems are reducible from both the bounded distance decoding problem over ideal lattice and the approximate greatest common divisor problem.

Based on this new problem, we proposed a new fully homomorphic encryption scheme using hidden ideal lattice. Our scheme can be seen as a hybrid between Gentry and Halevi's scheme with ideal lattice, and the van Dijk et al.'s scheme with integers, if we view the integer scheme as an ideal lattice with dimension equals to 1. We combined the strength of both schemes. On one hand, by hiding the lattice (as in the integer scheme), we showed that the dimension of lattice of the best attack to this cryptosystem is at least quadratic in the dimension of the generating ideal lattice, therefore, our scheme can operate with an ideal lattice with a smaller dimension. On the other hand, by using a lattice with dimension greater than 1 (as in Gentry and Halevi's scheme), we improved the security and efficiency of the scheme.

To sum up, our scheme makes the best alternative to the state-of-the-art fully homomorphic encryptions schemes based on ring learning with errors.

References

[1] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In S. Goldwasser, editor, *ITCS*, pages 309–325. ACM, 2012.

[2] Z. Brakerski and V. Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In R. Ostrovsky, editor, *FOCS*, pages 97–106. IEEE, 2011.

[3] Z. Brakerski and V. Vaikuntanathan. Fully homomorphic encryption from Ring-LWE and security for key dependent messages. In Rogaway [29], pages 505–524.

[4] Y. Chen and P. Q. Nguyen. BKZ 2.0: Better lattice security estimates. In D. H. Lee and X. Wang, editors, *ASIACRYPT*, volume 7073 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2011.

[5] Y. Chen and P. Q. Nguyen. Faster algorithms for approximate common divisors: Breaking fully-homomorphic-encryption challenges over the integers. In Pointcheval and Johansson [27], pages 502–519.

[6] J.-S. Coron, A. Mandal, D. Naccache, and M. Tibouchi. Fully homomorphic encryption over the integers with shorter public keys. In Rogaway [29], pages 487–504.

[7] J.-S. Coron, D. Naccache, and M. Tibouchi. Public key compression and modulus switching for fully homomorphic encryption over the integers. In Pointcheval and Johansson [27], pages 446–464.

[8] C. Gentry. Fully homomorphic encryption using ideal lattices. In M. Mitzenmacher, editor, *STOC*, pages 169–178. ACM, 2009.

[9] C. Gentry. *A Fully Homomorphic Encyrption Scheme*. PhD thesis, Stanford University, September 2009.

[10] C. Gentry. Computing arbitrary functions of encrypted data. *Commun. ACM*, 53(3):97–105, 2010.

[11] C. Gentry. Toward basing fully homomorphic encryption on worst-case hardness. In T. Rabin, editor, *CRYPTO*, volume 6223 of *Lecture Notes in Computer Science*, pages 116–137. Springer, 2010.

[12] C. Gentry and S. Halevi. Implementing Gentry's fully-homomorphic encryption scheme. In K. G. Paterson, editor, *EUROCRYPT*, volume 6632 of *Lecture Notes in Computer Science*, pages 129–148. Springer, 2011.

[13] C. Gentry, S. Halevi, C. Peikert, and N. P. Smart. Ring switching in bgv-style homomorphic encryption. In I. Visconti and R. D. Prisco, editors, *SCN*, volume 7485 of *Lecture Notes in Computer Science*, pages 19–37. Springer, 2012.

[14] C. Gentry, S. Halevi, and N. P. Smart. Better bootstrapping in fully homomorphic encryption. In M. Fischlin, J. Buchmann, and M. Manulis, editors, *Public Key Cryptography*, volume 7293 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2012.

[15] C. Gentry, S. Halevi, and N. P. Smart. Fully homomorphic encryption with polylog overhead. In Pointcheval and Johansson [27], pages 465–482.

[16] O. Goldreich, S. Goldwasser, and S. Halevi. Public-key cryptosystems from lattice reduction problems. In B. S. K. Jr., editor, *CRYPTO*, volume 1294 of *Lecture Notes in Computer Science*, pages 112–131. Springer, 1997.

[17] S. Goldwasser and S. Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

[18] G. H. Golub and C. F. van Loan. *Matrix computations (3. ed.)*. Johns Hopkins University Press, 1996.

[19] N. Howgrave-Graham. Approximate integer common divisors. In J. H. Silverman, editor, *CaLC*, volume 2146 of *Lecture Notes in Computer Science*, pages 51–66. Springer, 2001.

[20] Y.-K. Liu, V. Lyubashevsky, and D. Micciancio. On bounded distance decoding for general lattices. In J. Díaz, K. Jansen, J. D. P. Rolim, and U. Zwick, editors, *APPROX-RANDOM*, volume 4110 of *Lecture Notes in Computer Science*, pages 450–461. Springer, 2006.

[21] J. Loftus, A. May, N. Smart, and F. Vercauteren. On CCA-Secure fully homomorphic encryption. In *Selected Areas in Cryptography*, 2011.

[22] V. Lyubashevsky and D. Micciancio. On bounded distance decoding, unique shortest vectors, and the minimum distance problem. In S. Halevi, editor, *CRYPTO*, volume 5677 of *Lecture Notes in Computer Science*, pages 577–594. Springer, 2009.

[23] D. Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. *Computational Complexity*, 16(4):365–411, 2007.

[24] D. Micciancio. Lattice-based cryptography. In H. C. A. van Tilborg and S. Jajodia, editors, *Encyclopedia of Cryptography and Security (2nd Ed.)*, pages 713–715. Springer, 2011.

[25] P. Q. Nguyen. Cryptanalysis of the Goldreich-Goldwasser-Halevi cryptosystem from crypto '97. In M. J. Wiener, editor, *CRYPTO*, volume 1666 of *Lecture Notes in Computer Science*, pages 288–304. Springer, 1999.

[26] P. Q. Nguyen. Breaking fully-homomorphic-encryption challenges. In D. Lin, G. Tsudik, and X. Wang, editors, *CANS*, volume 7092 of *Lecture Notes in Computer Science*, pages 13–14. Springer, 2011.

[27] D. Pointcheval and T. Johansson, editors. *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, volume 7237 of *Lecture Notes in Computer Science*. Springer, 2012.

[28] R. Rivest, L. Adleman, and M. Dertouzos. On data banks and privacy homomorphisms. In *Foundations of Secure Computation*, pages 169–177. Academic Press, 1978.

[29] P. Rogaway, editor. *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, volume 6841 of *Lecture Notes in Computer Science*. Springer, 2011.

[30] P. Scholl and N. P. Smart. Improved key generation for gentry's fully homomorphic encryption scheme. In L. Chen, editor, *IMA Int. Conf.*, volume 7089 of *Lecture Notes in Computer Science*, pages 10–22. Springer, 2011.

[31] N. P. Smart and F. Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. In P. Q. Nguyen and D. Pointcheval, editors, *Public Key Cryptography*, volume 6056 of *Lecture Notes in Computer Science*, pages 420–443. Springer, 2010.

[32] D. Stehlé and R. Steinfeld. Faster fully homomorphic encryption. In M. Abe, editor, *ASIACRYPT*, volume 6477 of *Lecture Notes in Computer Science*, pages 377–394. Springer, 2010.

[33] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. Fully homomorphic encryption over the integers. In H. Gilbert, editor, *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 24–43. Springer, 2010.

Thomas Plantard received the MS and Ph.D. degrees in computer science from the Universite de Bordeaux in 2002 and the Universite Montpellier 2, France, in 2005, respectively. Since September 2006, he has a postdoctoral position at the University of Wollongong, Australia. His research interests include cryptography and lattice theory.

Willy Susilo (IEEE Senior Member since 2002) received a Ph.D. in Computer Science from University of Wollongong, Australia. He is a Professor at the School of Computer Science and Software Engineering and the director of Centre for Computer and Information Security Research (CCISR) at the University of Wollongong. He is currently holding the prestigious ARC Future Fellow awarded by the Australian Research Council (ARC). His main research interests include cryptography and information security. His main contribution is in the area of digital signature schemes. He has served as a program committee member in dozens of international conferences. He has published numerous publications in the area of digital signature schemes and encryption schemes.

Zhenfei Zhang received his Master of Internet Technology degree and Master of Engineering degree from University of Wollongong in 2009 and 2007, respectively. Since March 2010, he has been pursuing a Ph.D. degree in computer science from Centre for Computer and Information Security Research (CCISR) at the University of Wollongong, under the supervison of Prof. Willy Susilo and Dr. Thomas Plantard. His reserach interest includes fully homomorphic encryption and lattice theory.