A Weakly Homomorphic Encryption with LDN

Chao Feng*
School of Information Science and Engineering, Shandong University, Jinan, China
*Corresponding author, Email: chaojuan99@hotmail.com

Yang Xin, Hongliang Zhu, and Yixian Yang
National Engineering Laboratory for Disaster Backup and Recovery, Beijing University of Posts and Telecommunications, Beijing, China
Email: {yangxin, hongliangzhu, yxyang}@bupt.edu.cn

***Abstract*—As an effective solution to protect the privacy of the data, homomorphic encryption has become a hot research topic. Existing homomorphic schemes are not truly practical due to their huge key size. In this paper, we present a simple weakly homomorphic encryption scheme using only elementary modular arithmetic over the integers rather than working with ideal lattices. Compared with DGHV's construction, the proposed scheme has shorter public key and ciphertext size. The main appeal of our approach is the conceptual simplicity. We reduce the security of weakly homomorphic scheme to "learning divisor with noise (LDN)".**

***Index Terms*—Cryptography; Weakly Homomorphic Encryption; LDN; DGHV**

I. Introduction

After the discovery of public-key cryptography by Diffie and Hellman [1] in the year 1976, the necessity for total privacy of digital data has become stronger, especially since the internet has become an indispensable part of our private and work lives. One way to achieve confidentiality in various applications, such as online banking, electronic voting, virtual networks, etc., is homomorphic encryption scheme. The question for constructing such a scheme arose within a year of the development of RSA [2]. Rivest et al. [3] firstly proposed a beautiful idea saying that whether there exists an encryption scheme that permitting computation on encrypted data without decryption. In a nutshell, a fully homomorphic encryption scheme is an encryption scheme that allows a worker to perform arbitrary computations on encrypted data. At a high-level, the essence of fully homomorphic encryption is simple: given ciphertexts that encryption of $m_1,...,m_t$, one can evaluate any desired (computation efficiently)function f on the ciphertexts, and outputs an evaluated ciphertext that encryption of the $m_f = f(m_1,...,m_t)$. No information about plaintexts $m_1,...,m_t$ or $m_f = f(m_1,...,m_t)$, or other intermediate plaintexts leak; that is to say, the input and output, and intermediate values are transparent.

For more than 30 years, it was unclear whether fully homomorphic encryption was achievable or not. During this period, a novelty encryption system was proposed by the Boneh- Goh-Nissim [4], which supported an unlimited number of addition operations but one multiplication at the most. In 2009 Gentry [5] [6] came up with the first construction of such a scheme based on ideal lattices. Soon after Gentry's initial paper appeared, Smart and Vercauteren [7] improved Gentry's scheme by shorten the key size and ciphertext size. In Asiacrypt'2010, Stehle and Steinfeld [8] proposed an improved scheme of Gentry's scheme, which introduced decryption errors to reduce computation cost. In Eurocrypt'2010, van Dijk et al. [9] proposed a simply homomorphic public-key encryption scheme, named as DGHV. The innovation in the scheme was the construction of a weakly homomorphic encryption scheme over integers instead of ideal lattices. Building on the work of DGHV, Coron et al. [10] [11] [12] proposed three new schemes, each of them was quite potential in improving the performance and thus suitable for applications. Brakerski and Vaikuntanathan [13] [14] proposed two homomorphic encryption schemes based on the ring LWE problem [15], in which security can be reduced to the worst-case hardness problem on ideal lattices. Xiao [16] and Ramaiah [17] also proposed two efficient homomorphic encryption schemes. In [18], a simply homomorphic encryption scheme was proposed, which was useful for cloud computing. The previous works are described in table I.

TABLE I. THE PREVIOUS WORK

scheme	Algebraic structure	Hard problem	Message space	homomorphic
[5]	Ideal lattices	GapSvp	{0,1}	Fully homomorphic
[7]	Polynomial ring	SPIP	{0,1}	Fully homomorphic
[8]	Ideal lattices	SSSP	{0,1}	Fully homomorphic
[13]	Polynomial ring	LWE	{0,1}	Fully homomorphic
[14]	Polynomial ring	LWE	$\{0,1\}^n$	Fully homomorphic
[9]	Finite field $\mathbb{Z}_q$	AGCD	{0,1}	Somewhat homomorphic
Scheme in this paper	Finite field $\mathbb{Z}_q$	LDN	{0,1}	Somewhat homomorphic

doi:10.4304/jnw.9.6.1464-1470

A. Our Results

Informally, one wants the homomorphicaly generated encryption of m_f to be statistically close to the ciphertext corresponding to m_f. We call the schemes that have this property strongly homomorphic. For some applications, it seems that strongly homomorphic encryption is overkill. We need weaker notion of homomorphic encryption that might be easier to construct and still suffice for these applications. The minimal requirement is that a homomorphically generated encryption can be decrypted correctly to the corresponding message. Unfortunately, the minimal requirement does not seem to be useful as is, because it captures schemes that we do not really consider to be homomorphic. Specifically, we call an encryption scheme weakly homomorphic if homomorphically generated ciphertexts can be decrypted correctly to the corresponding message, as well as their lengths only depend on the security parameters and the message length instead of the number of input ciphertexts. Hence, in this paper we mainly focus on the weakly homomorphic encryption.

Firstly, we propose a weakly homomorphic encryption scheme whose security is based solely on the hardness of learning divisor with noise (LDN). Secondly, inspired by [19], we propose a technique, which can refresh the rough ciphertext and gets a new ciphertext with smaller noise. Thirdly, the private key homomorphic encryption scheme is transferred to its public key version. Compared with the previous works, the construction only use elementary modular arithmetic over the integers rather than working with ideal lattices. And it is simpler in pedagogical and theoretical. Note that, the paper concentrates on the weakly homomorphic scheme, as we think that the *making it fully homomorphic* procedure in [9] is straightforward.

B. Roadmap

The paper is structured as follows. Section II presents the notations and some definitions used throughout the paper. Section III proposes a weakly homomorphic encryption scheme over integers. Section IV proposes a clean procedure that refreshes the rough ciphertext. This technique significantly reduces the noise of ciphertext. Section V describes a conversion from the private key to public key. Section VI concludes this paper and gives some open problems.

Circular-Security. Unfortunately, during the *Clean* procedure, the security of rough scheme cannot guarantee. This notion is called circular security and it is a subclass of key dependent message (KDM) security. Using a common cryptographic technique - Sparse subset sum assumption (SSSP), the modified scheme in this paper is circular security.

II. Preliminary

Notations. In this paper, the parameters of the scheme are denoted by Greek letters (e.g., p,q,n etc.), particularly, n is the security parameter. The real number and integer are denoted by lowercase letters, and the set is denoted by capital letters. For a real number x, the upper and lower bound, and its nearest integer are denoted by $\lceil x \rceil$, $\lfloor x \rfloor$, $\lfloor x \rceil$ respectively. Namely, these are the unique integers in the half open intervals $(x-1,x]$, $[x,x+1)$ and $(x-\frac{1}{2},x+\frac{1}{2}]$, respectively. For two integers z and p, we denote the quotient and remainder of z with respect to p by $q_p(z)$ and $r_p(z)$, respectively. For modular arithmetic, we can denote by $r_p(z)$ the reduction of $z \bmod p$ into the interval $(-p/2, p/2]$.

A. Hard Problem

Definition (Learning divisor with noise). The hardness assumption can be described as follows: let p a random n bit prime, R a random n^3 bit prime, and $N = pR$. Given a set of integers $x_1, x_2, ... x_\tau$, which are chosen randomly close to multiples of a large integer p, and e is the "noise", output the "common near divisor" p.

Definition (Sparse subset sum assumption). Given m n-bit numbers $\alpha_1,...\alpha_m$ and a target number β, whether there exists a subset $T \subseteq \{1,2,...,m\}$ such that $\sum_{i\in T}\alpha_i = \beta$. When m is sufficiently as large as a polynomial of n, it is a hard problem.

According to SSSP, for any $n,\kappa > 0$ and $\beta \in [-2^n, +2^n]$, the following two distributions over $\alpha_1,...,\alpha_m$ are computationally indistinguishable: (I) $\alpha_1,...,\alpha_m$ are chosen randomly and independently in $[-2^n, +2^n]$; (II) a subset is chosen as follows: the size of subset is n^κ, if $i \notin T$, α_i are chosen randomly and independently in $[-2^n, +2^n]$, otherwise $\sum_{i\in T}\alpha_i = \beta$.

B. Weakly Homomorphic Encryption

Definition (Correctness homomorphic encryption). A homomorphic private-key encryption scheme $\varepsilon = (KegGen, Enc, Eval, Dec)$ is correct for a given t-input function f if, for any private key sk, any t plaintext bits $m_1,...,m_t$ and any ciphertexts $\vec{c} = \langle c_1,...,c_t \rangle$ with $c_i = Enc(sk, m_i)$, it is the case that:

$$Dec(sk, Eval(sk, f, \vec{c})) = f(m_1,...,m_t) \qquad (1)$$

The scheme $\varepsilon = (KegGen, Enc, Eval, Dec)$ is homomorphic for a function set F if it is correct for all functions $f \in F$.

Definition (Discrete Statistical distance). It is a measure of distance between two discrete probability distributions and a useful tool in the analysis of randomized algorithms. X and Y are two discrete random variables over a set A. Define the Discrete Statistical Distance between X and Y as

$$\Delta(X,Y)=\frac{1}{2}\sum_{a\in A}\left|\Pr\{X=a\}-\Pr\{Y=a\}\right| \qquad (2)$$

Definition (Weakly homomorphic encryption). Let $c_1,...,c_t$ be the ciphertext corresponding to $m_1,...,m_t$ respectively, $c_{Eval}=Eval(f,(c_1,...,c_t))$ and $c_{Enc}=Enc(f(m_1,...,m_t))$. During the evaluate phase, the noise is chosen randomly, so c_{Eval} and c_{Enc} are two random variables. The homomorphic encryption is weakly, for any ciphertext when $Dec(c_{Eval})=f(m_1,...,m_t)$, $c_{Eval}\neq c_{Enc}$, as well as $\Delta(c_{Eavl},c_{Enc})<negl(n)$. In which $negl(n)$ is a neglect quantity.

III. CONSTRUCTION

A. The Basic Scheme

Let n be the security parameter, and the message space is $\mathbb{Z}_2$. The parameter is similar to the DGHV's convenient parameter setting [9].

$KeyGen_{sk}(1^n)$. We choose p to be a random n-bit prime, and R to be a random n^3 bit prime, $N=pR$. We keep p secret, and publish N as a public parameter.

$Enc_{sk}(sk,m)$. Choose $q\leftarrow_R[N/p]$, choose a $\sqrt{n}$-bit random integers e . Output the ciphertext $c=qp+2e+m(\bmod N)$.

$Dec_{sk}(sk,c)$. Compute $m=(c\bmod p)\bmod 2$, output the ciphertext c.

$Eval(C,(c_1,c_2,...,c_k))$. The arithmetic circuit C corresponds to a multivariate polynomial f with k variables. For any two ciphertexts c_1 and c_2, addition and multiplication operations are described as follows:

Addition: $c_{add}=(c_1+c_2)\bmod N$

Multiplication: $c_{mult}=(c_1\cdot c_2)\bmod N$

B. Correctness

Considering the fresh ciphertext $c=qp+2e+m(\bmod N)$ output by $Enc_{sk}(sk,m)$. In order to correctly decrypt the fresh ciphertext, the term $2e+m$ should be less than $p/2$ (for an integer a, $a\bmod p\in(-p/2,p/2]$). According to the parameter setting, the absolute value of the e is less than $2^{\sqrt{n}}$, p is a n-bit prime. That is to say, $2e\ll p/2$, $m\in\{0,1\}$, so $2e+m\ll p/2$. We have $c\bmod p=2e+m$, and then take $m=2e+m(\bmod 2)$. Hence, $Dec_{sk}(sk,c)$ works properly for the fresh ciphertext.

$Eval(C,(c_1,c_2,...,c_t))$. Notice that the scheme supports two universal operations addition and multiplication homomorphicly. The addition and multiplication of ciphertext can be defined as follows: $Add(c_1,c_2)=c_1+c_2\bmod N$, $Mult(c_1,c_2)=c_1\cdot c_2\bmod N$.

Addition: $c_1=q_1p+2e_1+m$, $c_2=q_2p+2e_2+m$, then

$$c_{add}=c_1+c_2=(q_1+q_2)p+2(e_1+e_2)+(m_1+m_2) \qquad (3)$$

Multiplication:

$$\begin{aligned}c_{mult}&=c_1\cdot c_2=(q_1p+2e_1+m_1)\cdot(q_2p+2e_2+m_2)\\&=(q_1q_2p+2q_1e_2+2q_2e_1+q_1m_2+q_2m_1)p\\&+4e_1e_2+2e_1m_2+2e_2m_1+m_1m_2\end{aligned} \qquad (4)$$

Following the Eq. (3) and Eq. (4), it can be concluded that the noise (the distance from ciphertext c to the multiply of private key p) of c_{add} is $c_{add}-(q_1+q_2)p=2(e_1+e_2)+(m_1+m_2)$. According to the parameter setting, the absolute value of the e for encryption is less than $2^{\sqrt{n}}$, p is a n-bit prime; we have $2(e_1+e_2)\ll p/2$, then $2(e_1+e_2)+(m_1+m_2)\ll p/2$, $c_{add}\bmod p=2(e_1+e_2)+(m_1+m_2)$, and $2(e_1+e_2)+(m_1+m_2)(\bmod 2)=m_1+m_2$. Similarly, the noise of c_{mult} is

$$\begin{aligned}&c_1\cdot c_2-(q_1q_2p+2q_1e_2+2q_2e_1+q_1m_2+q_2m_1)p\\&=4e_1e_2+2e_1m_2+2e_2m_1+m_1m_2\end{aligned} \qquad (5)$$

The absolute value of e for encryption is less than $2^{\sqrt{n}}$, p is a n-bit prime; we have $4e_1e_2+2e_1m_2+2e_2m_1+m_1m_2\leq 4\cdot 2^{2\sqrt{n}}+4\cdot 2^{\sqrt{n}}+1\ll p/2$. It is easy to verify $c_{mult}\bmod p=4e_1e_2+2e_1m_2+2e_2m_1+m_1m_2$ and $(4e_1e_2+2e_1m_2+2e_2m_1+m_1m_2)\bmod 2=m_1m_2$. We can conclude that $Dec_{sk}(sk,c)$ and $Eval(C,(c_1,c_2,...,c_t))$ work properly for the fresh ciphertext and evaluated ciphertext.

C. Homomorphic

In this subsection, we formally discuss the homomorphic property of the proposed scheme. Clearly, since addition and multiplication form a complete set of operations, such a scheme enables performing any polynomial-time computation on encrypted data. Note that, for any two ciphertexts c_1 and c_2, addition and multiplication operations are described as follows: addition: $c_{add}=(c_1+c_2)\bmod N$, and multiplication: $c_{mult}=(c_1\cdot c_2)\bmod N$.

By the triangle inequality, a d additive operations on ciphertexts increases the magnitude of the integers by a factor of d at most. However, one multiplicative operation on ciphertexts may square the magnitude of the integers, that's to say, double their bit-lengths. Clearly, the main influence of the homomorphic capacity is the multiplicative depth of the arithmetic circuit C . In addition, the scheme supports multiplications as long as the nosie of ciphertext is sufficiently smaller than $p/2$. According to the parameter setting, we can conclude that the scheme supports approximately $\sqrt{n}$ multiplications on ciphertexts.

D. Security

The security of the proposed scheme including two parts: CPA-security and key-security. For the private-key construction, key-security can be guaranteed. In this section, we mainly discuss the CPA-security, and the key-security of public-key construction will be discussed in section V.

Definition (CPA-security). A scheme HE is IND-CPA secure for any polynomial time adversary atk , Adv is the probability of atk guess the plaintext from ciphertexts, it holds that

$$\begin{aligned}\text{Adv}[atk] &\overset{\Delta}{=} \left|\Pr[atk(Enc_{sk}(1) = 1 - \Pr[atk(Enc_{sk}(0) = 1\right| \\ &= \text{negl}(n)\end{aligned} \tag{6}$$

Corollary. The scheme proposed in section III is CPA-security.

Proof. For any adversary atk , give plaintext 0, then $Enc_{sk}(0) = q_0 p + 2e_0$, $q_0 \leftarrow_R [N/p]$ and $e_0 \leftarrow_R \left[-2^{\sqrt{n}}, 2^{\sqrt{n}}\right]$. The set consists of $Enc_{sk}(0) = q_0 p + 2e_0$ has $2^{n^4+\sqrt{n}+1}$ elements. For any adversary atk , the private-key is transparent. For any adversary atk , the probability of correctly guess the plaintext corresponding to ciphertext $Enc_{sk}(0)$ is $\frac{1}{2^{n^4+\sqrt{n}+1}} = negl(n)$, which means that

$$\begin{aligned}\text{Adv}[atk] &\overset{\Delta}{=} \left|\Pr[atk(Enc_{sk}(1) = 1 - \Pr[atk(Enc_{sk}(0) = 1\right| \\ &= \text{negl}(n)\end{aligned}.$$

We can conclude that the scheme proposed in section III is CPA-security.

IV. Reducing the Noisy

When two ciphertexts c_1 and c_2 are added ($c_{add} = c_1 + c_2$), the noise term in c_{add} essentially doubles, and when they are multiplied ($c_{mult} = c_1 \cdot c_2$), the noise term is squared. If the noise gets too large, the plaintext message becomes impossible to recover. In this section, we propose a technique that refreshes the rough ciphertext to obtain a refreshed ciphertext with small noise, named as "*Clean*". The technique does not change the previous encryption and decryption algorithms, and we just add a "hint" into the public parameters. Putting the hint $W_1,...W_m$ in *Clean* procedure induces another computational assumption, related to the SSSP. We can easily avoid some known attacks on the problem by choosing T large enough to avoid brute-force attacks. Thus, all the properties of the scheme (e.g. correctness, homomorphism) will be preserved. The implementation of *Clean* procedure is described as below.

A. The Implementation of Clean

Given a ciphertext $c = qp + 2e + m$, the clean subroutine come up with a new ciphertext $c_{refresh} = q' p + 2e' + m$ with smaller noise, for Y_i to be an encryption with noise parameter $2^{n^{0.1}}$. The subroutine can be described as follows:

Algorithm. *Clean*

Step1: Let $a = c(\text{mod}\,2)$. Since p is odd, we have $m = a \oplus \lfloor c/p \rceil(\text{mod}\,2)$. The goal is to come up with an intermediate value c_{inter} , which is the encryption of $a' = \lfloor c/p \rceil(\text{mod}\,2)$. Set $c_{refresh} = c_{inter} + a$.

Step2: Choose m n -bit numbers $\alpha_1,...\alpha_m$, there exists a subset $T \subseteq \{1,2,...m\}$ such that $\sum_{i\in T} \alpha_i = 1/p$. Compute $W_1,...W_m$ where $W_i = c \cdot \alpha_i$, $\sum_{i\in T} W_i = c/p$. Denote the truncation of number W_i by $\tilde{W}_i + 2$, throw away all the binary digits corresponding to 2^i for $i > 0$ and $i \le -2\log k$, then $\lfloor \sum_{i\in T} \tilde{W}_i \rceil(\text{mod}\,2) = \lfloor c/p \rceil(\text{mod}\,2)$.

Step3: Let y_i be 1 if $i \in T$, otherwise 0. Let $w_{i,1},...,w_{i,2\log k}$ denote the binary digits of the $y_i \tilde{w}_i$. Then, $y_i = 0$, $w_{i,j} = 0$ for all j , and otherwise $w_{i,j}$ is the j digit of $\tilde{W}_i$.

Step4: Compute $W_{i,j} = Y_i \cdot w_{i,j}$, we have $c_{inter} = \sum_{i\in[t]} \sum_{-2\log k+1}^{0} W_{i,j} 2^{\ell}$, output $c_{refresh} = c_{inter} + a$. We can conclude that the $c_{refresh}$ with smaller noise, for Y_i to be an encryption with noise parameter $2^{n^{0.1}}$.

Theorem. Denote the truncation of W_i by $\tilde{W}_i$, throw away all the binary digits corresponding to 2^i for $i > 0$ and $i \le -2\log k$, then

$$\lfloor \sum_{i\in T} \tilde{W}_i \rceil(\text{mod}\,2) = \lfloor c/p \rceil(\text{mod}\,2)$$

Proof. Throwing the digits corresponding to $2^{-2\log k}$ or smaller can introduce at most $1/2$ difference to the sum of W_i . Since c/p is very close to an integer, after changing it by adding a number in $[-1/2, 1/2]$, round to the same integer with high probability. Changing digits corresponding to 2 or larger will only add or subtract an even number from c/p , that also not change its value modulo 2.

Theorem. If $c_{inter} = \sum_{i\in[t]} \sum_{-2\log k+1}^{0} W_{i,j} 2^{\ell}$, then $c_{inter} = Enc_{sk}(a')$.

Proof. Give $c_{inter} = \sum_{i\in[t]} \sum_{-2\log k+1}^{0} W_{i,j} 2^{\ell}$, we can conclude that

$$c_{inter} = \sum_{i\in[t]} \sum_{-2\log k+1}^{0} W_{i,j} 2^{\ell} = \sum_{i\in[t]} Y_i \sum_{-2\log k+1}^{0} w_{i,j} 2^{\ell} = \sum_{i\in[t]} Y_i W_i \tag{7}$$

Y_i to be an encryption (with noise parameter $2^{n^{0.1}}$) of 1 if $i \in T$ and of 0 if $i \notin T$, we have $Y_i \equiv w_{i,j} \bmod 2$. Then, it is easy to verify that

$$\begin{aligned} Dec_{sk}(c_{\mathrm{int}er}) &= c_{\mathrm{int}er} \bmod p \bmod 2 = (\sum_{i\in[t]} Y_i W_i) \bmod p \bmod 2 \\ &= (\sum_{i\in[t]} w_{i,j} W_i) \bmod p \bmod 2 = (\sum_{i\in T} W_i) \bmod p \bmod 2 \\ &= \lfloor (\sum_{i\in T} W_i) \rceil \bmod p \bmod 2 = a' \end{aligned} .$$

We can conclude that, when $c_{\mathrm{int}er} = \sum_{i\in[t]} \sum_{-2\log k+1}^{0} W_{i,j} 2^{\ell}$, $c_{\mathrm{int}er} = Enc_{sk}(a')$.

V. Transform Private Key to Public Key Homomorphic Encryption

From a high-level point of view, the weakly public-key homomorphic scheme is constructed by first proposing a simple weakly private-key homomorphic scheme. The intuition for how to move from private to public key can be seen in a straightforward manner in the case of weakly homomorphic schemes. The following construction was suggested implicitly in [9].

Let Enc_{sk} and Dec_{sk} be the respective encryption and decryption algorithm of a weakly private-key encryption scheme. Suppose that this encryption scheme is weakly homomorphic w.r.t the identity function. That is, it is possible to re-randomize ciphertexts. Such a scheme can be used to construct a weakly public-key homomorphic scheme as follows. The secret key is the key of primal private key scheme, and the public key consists of an encryption of 0 (i.e. $Enc_{sk}(0)$). To encrypt a bit m, just randomize the ciphertext corresponding to m. To decrypt the ciphertext, apply the private-key decryption algorithm using sk. There is an additional parameter r_i, which is the distance between the public key elements and the nearest multiples of the secret key p.

$KeyGen_{pk}(1^n)$. We choose p to be a random n bit prime, and R to be a random n^3 bit prime, $N = pR$. Keep p secret, and publish N as a public parameter. The secret key is p. Then choose uniformly from the interval $q_i \leftarrow_R Z \cap [0, N/p)$, and $r_i \leftarrow_R Z \cap [-2^{\sqrt{n}}, 2^{\sqrt{n}})$, compute $x_i = pq_i + 2r_i$, for $i = 0, \ldots, \tau$. $\tau = n$. Relabel so that x_0 is the largest, and x_0 is odd. The public key is $pk = \langle x_0, x_1, \ldots x_\tau \rangle$.

$Enc_{pk}(pk, m)$. Choose a random subset $S \subseteq \{1, 2, \ldots, \tau\}$, choose a $\sqrt{n}$-bit random integers e, then output a ciphertext $c = [2\sum_{i\in S} x_i + 2e + m](\bmod x_0)$.

$Eval(pk, C, c_1, \ldots, c_t)$. Given the circuit C with t ciphertexts c_i, apply the (integer) addition and multiplication gates of C to the ciphertexts, perform all the operations over the integers, and output the resulting integer.

$Dec_{pk}(sk, c)$. Output $m = c \bmod p (\bmod 2)$.

A. Correctness

Considering the fresh ciphertext $c = [2\sum_{i\in S} x_i + 2e + m](\bmod x_0)$ output by $Enc_{pk}(pk, m)$. We have $c = 2\sum_{i\in S} x_i + 2e + m - a \cdot x_0$ for an integer a. Note that, $x_i = pq_i + 2r_i$, for $i = 0, \ldots, \tau$, $\tau = n$, we can conclude that $c = b \cdot p + 2e + m$ for an integer b, $b \neq a$. In order to correctly decrypt the fresh ciphertext, the term $4\sum_{i\in S} r_i + 2e + m$ should be less than $p/2$ (for an integer a, $a \bmod p \in (-p/2, p/2]$). According to the parameter setting, it can be seen that the absolute value of the e is less than $2^{\sqrt{n}}$, $r_i \leftarrow_R Z \cap [-2^{\sqrt{n}}, 2^{\sqrt{n}})$, p is a n-bit prime. That is to say, $4\sum_{i\in S} r_i + 2e \ll p/2$, $m \in \{0,1\}$, so $4\sum_{i\in S} r_i + 2e + m \ll p/2$, $c \bmod p = 4\sum_{i\in S} r_i + 2e + m$, and compute $m = 4\sum_{i\in S} r_i + 2e + m(\bmod 2)$. Hence, the decryption procedure $Dec_{pk}(sk, c)$ works properly for the fresh ciphertext.

B. Homomorphic

Notice that the scheme can supports two universal operations *Add* and *Mult* homomorphicly. Given two ciphertexts $c_1 = [2\sum_{i\in S_1} x_i + 2e_1 + m_1](\bmod x_0)$, and $c_2 = [2\sum_{i\in S_2} x_i + 2e_2 + m_2](\bmod x_0)$. The addition and multiplication of ciphertext can be defined as follows: $Add(c_1, c_2) = c_1 + c_2 \bmod N$, $Mult(c_1, c_2) = c_1 \cdot c_2 \bmod N$.

Addition: according to the Enc_{pk} procedure, assume that $c_1 = 2\sum_{i\in S_1} x_i + 2e_1 + m_1 - c \cdot x_0$, $c_2 = 2\sum_{i\in S_2} x_i + 2e_2 + m_2 - d \cdot x_0$ for two integers c and d, then we can compute

$$c_1 + c_2 = 2\sum_{i\in S_1 \cup S_2} x_i + 2(e_1 + e_1) + m_1 + m_1 - (c+d) \cdot x_0 \quad (8)$$

Since, $2\sum_{i\in S_1 \cup S_2} r_i + 2e_1 + 2e_2 + m_1 + m_2 \ll p/2$, then

$$c_1 + c_2 (\bmod p)(\bmod 2) = m_1 + m_2 \quad (9)$$

Multiplication: the multiplication of ciphertext is

$$\begin{aligned} c_1 \cdot c_2 &= (2\sum_{i\in S_1} x_i + 2e_1 + m_1 - c \cdot x_0)(2\sum_{i\in S_2} x_i + 2e_2 + m_2 - d \cdot x_0) \\ &= p \cdot f + 2e_{mult} + m_1 \cdot m_2 \end{aligned} \quad (10)$$

for an integer f. The noise of c_{mult} is $2e_{mult} + m_1 \cdot m_2$, where

$$\begin{aligned} e_{mult} &= 4\sum_{i\in S_1} r_i \sum_{i\in S_2} r_i + 4e_2 \sum_{i\in S_1} r_i + 4e_1 \sum_{i\in S_2} r_i \\ &+ 2m_2 \sum_{i\in S_1} r_i + 2m_1 \sum_{i\in S_2} r_i + 2e_1 m_2 + 2e_2 m_1 + m_1 \cdot m_2 \end{aligned} \quad (11)$$

According to the parameters setting, $2e_{mult} + m_1 \cdot m_2 \ll p/2$, it is easy to verify that

$$\begin{aligned} Dec_{pk}(c_1 \cdot c_2) &= c_1 \cdot c_2 (\bmod p)(\bmod 2) \\ &= 2e_{mult} + m_1 m_2 (\bmod 2) = m_1 m_2 \end{aligned} \qquad (12)$$

We can conclude that $Eval(C,(c_1,c_2,...,c_t))$ work properly for the evaluated ciphertext.

C. Security

For the public-key construction, the key-security is the main issue.

Key-security. The key-security of the scheme can be reduced to the hardness of the *LDN* problem. Namely, given a set of integers $x_0, x_1, x_2, ...x_\tau$, all randomly chosen close to multiples of a large integer p, e is the "noise", and find this "common near divisor" p. That is to say, the key of the weakly public-key homomorphic encryption is an instance of the *LDN* problem. Due to the similarities in the construction and ciphertexts, the proposed scheme and the DGHV's scheme are identical with only differences in the plaintexts they encrypt and the public key size. Hence, the same strategy employed in DGHV's [9] and CMNT's scheme [10] can be applied in reducing the security of the proposed scheme to solving the *LDN* problem. The proof of Theorem is similar with the proof of theorem 4.2 in [9].

Theorem (DGHV, Theorem 4.2). An adversary *atk* (attacker) with a non-negligible advantage σ against the scheme leads to a ppt -algorithm Ψ for solving the *LDN* problem with a success probability of σ.

Proof: We assume that an attacker *atk* can breaks the semantic security of the scheme with advantage σ. We use *atk* to construct a solver Ψ for *LDN*. For the randomly chosen n -bit prime p, the solver Ψ has access to as many pair (c_i, pk) as it needs, and the goal is to find the secret key p. The challenger generates an instance (pk, sk) of the *LDN* assumption, and delivers the public key to the attacker atk. The attacker atk accesses the encryption procedure, and requests a set of ciphertexts $c_1, c_2, ..., c_\vartheta$ of the encryption of 0, where $c_i = p\eta_i + 2\upsilon_i$ for two rand integers η_i and υ_i. Then the attacker atk delivers the ciphertexts $c_1, c_2, ..., c_\vartheta$ and the public key pk to the solver Ψ. Next, the solver Ψ calls the attacker atk to get a prediction $\upsilon_i^* = atk(c_i, pk)$, that is to say $c_i = p\eta_i + 2\upsilon_i^*$. Choose two random integers c_1 (with a prediction υ_1^*) and c_2 (with a prediction υ_2^*) from the set of ciphertexts $c_1, c_2, ..., c_\vartheta$, we can compute $c_1^* = c_1 - 2\upsilon_1^*$ and $c_2^* = c_2 - 2\upsilon_2^*$. In the end, compute $p = \gcd(c_1^*, c_2^*)$. As a result, we can conclude that a ppt - algorithm Ψ can solves the *LDN* assumption with a success probability of σ.

D. Complexity

The detailed complexity analysis is given as follows.

Size of private key: the private key is an n bits integer p, and the size of public key is $O(n^6)$ bits.

Length of ciphertext: a 1-bit plaintext is encrypted into a ciphertext $c = [2\sum_{i \in S} x_i + 2e + m](\bmod x_0)$ of $O(n^4)$ bits.

Complexity of encryption: It needs to compute a rough integer multiply, $n-1$ additions and one module, the total computation cost of encryption is $O(n^5)$.

Computation cost of decryption: It needs to compute division and rounding, the total computation cost of decryption is $O(n^7)$.

Homomorphic addition: The addition of two ciphertexts is simply an integer addition. After an addition, the length of ciphertext is not increased, and accordingly the computation cost of decryption remains the same. And the total computation cost of homomorphic addition is $O(n^7)$.

Homomorphic multiplication: When multiplying two ciphertexts, it needs to directly compute multiplication on two ciphertexts in the finite field $\mathbb{Z}_{x_0}$. After one multiplication, the length of ciphertext is not increased, and the computation cost of decryption remains the same, the total computation cost of homomorphic addition is $O(n^8)$. Table II shows that the proposed scheme is more effective than other schemes, e.g., DGHV [9] and CMNT [10].

TABLE II. COMPARED WITH RELATED WORK

Item	DGHV[9]	CMNT[10]	Scheme proposed in this paper
Underling algebraic structure	Integer	integer	integer
Size of pk	$O(n^{10})$	$O(n^7)$	$O(n^6)$
Message expansion	$O(n^5)$	$O(n^5)$	$O(n^4)$
Overall complexity	$O(n^{12})$	$O(n^{15})$	$O(n^8)$
Error-reducing	No	No	Yes
Hardness problem	AGCD	Error-free AGCD	LDN

VI. CONCLUSIONS

In this paper, we propose a weakly homomorphic encryption based on *LDN* problem. We reduce the CPA-security of the weakly homomorphic scheme to *LDN* problem. One contribution of our scheme is the small key size and ciphertext size. It is expected that, the scheme proposed in this paper makes the encrypted data processing practically for suitable applications. It would be interesting how to apply the public key compression technique of [11] to the weakly homomorphic scheme. Also, it is an open problem to improve the efficiency of the scheme, while preserving the hardness of the *LDN* problem.

ACKNOWLEDGMENTS

I would like to express my thanks and appreciation to my Doc advisor Yixian Yang, for his encouragement and guidance in completing this work. In particular, I would like to thank Prof. Shoushan Luo for many helpful guidance and constructive comments that helped present this work in a more coherent way. This work is supported by the National Natural Science Foundation of China, (Grant No. 61121061, 61161140320).

REFERENCES

[1] W. Diffie and M. Hellman, “New directions in cryptography”, *IEEE Transactions on Information Theory,* vol. 22(6), pp. 644-654, 1976.

[2] R. L. Rivest, A. Shamir and L. Adleman, “A method for obtaining digital signatures and public key cryptosystems”, *Communication of the ACM,* vol. 21(2), pp. 120-126, 1978.

[3] R. L. Rivest, L. Adleman, and M. L. Dertouzos, “On data banks and privacy homomorphisms”, In Foundations of Secure Computation, Academic Press, pp. 169-180, 1978.

[4] D. Boneh, E. J. Goh, and K. Nissim, Evaluating 2-DNF formulas on ciphertexts. *Proceedings of the second Theory of Cryptography Conference, Cambridge, MA, USA,* pp. 325-341, 2005.

[5] C. Gentry, Fully homomorphic encryption using ideal lattices, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, Bethesda, MD, USA,* pp. 169-178, 2009.

[6] C. Gentry, A fully homomorphic encryption scheme. *Ph. D. thesis, Stanford University,* 2009.

[7] N. P. Smart, F. Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. *Proceedings of the 13th International Conference on Practice and Theory in Public Key Cryptography, Paris, France,* pp. 420–443, 2010.

[8] D. Stehle and R. Steinfeld, Faster fully homomorphic encryption. *Proceedings of the 16th International Conference on the Theory and Application of Cryptology and Information Security,* December 5-9, 2010, Singapore, pp. 377-394, 2010.

[9] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, Fully homomorphic encryption over the integers. *Proceedings of the 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera,* pp. 24–43, 2010.

[10] J. S. Coron, A. Mandal, D. Naccache, and M. Tibouchi, Fully homomorphic encryption over the integers with shorter public keys. *Proceedings of the 31st Annual Cryptology Conference, Santa Barbara, CA, USA,* pp. 487-504, 2011.

[11] J. S. Coron, D. Naccache, and M. Tibouchi, Public key compression and modulus switching for fully homomorphic encryption over the integers. *Proceedings of the 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK,* pp. 446-464. Springer, 2012.

[12] J. S. Coron, D. Naccache, and M. Tibouchi, Batch Fully Homomorphic Encryption over the Integers. *Proceedings of the 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques,* Athens, Greece, pp. 315-335, 2013.

[13] Z. Brakerski and V. Vaikuntanathan, Efficient fully homomorphic encryption from (standard) LWE, *Proceedings of the 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science, Palm Springs, CA. USA,* pp. 97-106, 2011.

[14] Z. Brakerski, Fully homomorphic encryption without modulus switching from classical gapsvp. *Proceedings of the 32nd Annual Cryptology Conference, Santa Barbara, CA, USA,* pp. 868-886, 2012.

[15] Vadim Lyubashevsky, Chris Peikert, and Oded Regev, On ideal lattices and learning with errors over rings. *Proceedings of the 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera,* pp. 1-23, 2010.

[16] L. L. Xiao, O. Bastani, I. L. Yen, An Efficient Homomorphic Encryption Protocol for Multi-User Systems. *IACR Cryptology ePrint Archive* 2012: 193 (2012). Available at http://eprint. iacr. org/2012/193.

[17] Y. Govinda. Ramaiah, G. VijayaKumari, “Efficient public key generation for homomorphic encryption over the integers”, *3rd International Conference on Advances in Communication, Network and Computing, Janahan Lal Stephen (Ed.), LNICST,* pp. 262–268, Springer, 2012.

[18] A. Kipnis and E. Hibshoosh, Efficient Methods for Practical Fully Homomorphic Symmetric-key Encrypton, *Randomization and Verification.* IACR Cryptology ePrint Archive 2012: 637 (2012). Available at http://eprint. iacr. org/2012/637.

[19] B. Barak. Cryptography course - Lecture notes, COS 433. *Princeton University, Computer Science Department.* Available at http://www. cs. princeton. edu/ courses/archive/spring10/cos433.

Chao Feng was born in 1984. He received his B.S. degree in Shan Dong Normal University (SDNU), China in June 2007, and his M.S. degree in Shan Dong Normal University (SDNU), China in June 2010. He is currently working towards his Ph.D. degree in information security at Shan Dong University (SDU), China. His current research interest includes cryptography, homomorphic encryption and remote data checking.

Yang Xin was born in 1977; he received the B.S. degree in Communication Engineering from Shan Dong University (SDU), China in 2001. He received his PhD degree in electrical engineering and communication systems from Beijing University of Posts and Telecommunications (BUPT) in 2006. His current research interest includes cryptography, network security, and information security.

Hongliang Zhu was born in 1982; he received his M.S. degree in communication systems from Beijing University of Posts and Telecommunications (BUPT) in 2007. He received his PhD degree in information security from Beijing University of Posts and Telecommunications (BUPT) in 2010. His current research interest includes cryptography, information security, remote data checking, and network security.

Yixian Yang was born in 1961, is currently the Department Chairman of Information Security Center (ISC), Beijing University of Posts and Telecommunications (BUPT), dean of National Engineering Laboratory for Disaster Backup and Recovery, National Key Laboratory for Network and Information Defense. He received his PhD degree in electrical engineering and communication systems from BUPT in 1988. His research areas include network coding, coding theory, cryptography, and information security.