An Efficient Somewhat HE scheme over Integers and Its Variation

Haomiao Yang[1], Hyunsung Kim[2], Dianhua Tang[3] and Hongwei Li[1]
[1] School of Computer Science & Engineering, UESTC
Chengdu, 610054 - China
[e-mail: haomyang@uestc.edu.cn, hongwei.uestc@gmail.com]
[2] Dept. of Cyber Security, Kyungil University
Kyungsansi, 712-701 - Republic of Korea
[e-mail: kim@kiu.ac.kr]
[3] Science & Technology on Communication Security Laboratory
Chengdu, 610041 - China
[e-mail: tangdianhua86@163.com]
*Corresponding author: Hyunsung Kim

Received April 15, 2013; revised July 11, 2013; revised August 20, 2013; accepted September 21, 2013; published October 29, 2013

Abstract

In 2010, Dijk et al. demonstrated a simple somewhat homomorphic encryption (HE) scheme over the integers of which this simplicity came at the cost of a public key size in $\tilde{O}(\lambda^{10})$. Although in 2011 Coron et al. reduced the public key size to $\tilde{O}(\lambda^{7})$, it is still too large for practical applications, especially for the cloud computing. In this paper, we propose a new form of somewhat HE scheme to reduce further the public key size and a variation of the scheme to optimize the ciphertext size. First of all, we propose a new somewhat HE scheme which is built on the hardness of the approximate greatest common divisor (GCD) problem of two integers, where the public key size in the scheme is reduced to $\tilde{O}(\lambda^{3})$. Furthermore, we can reduce the length of the ciphertext of the new somewhat HE scheme by applying the modular reduction technique. Additionally, we give simulation results for evaluating ability of the proposed scheme.

Keywords: Somewhat homomorphic encryption, fully homomorphic encryption, approximate-GCD problem, cloud computing

A preliminary version of this paper appeared in CDCIEM 2012, March 5-6, Zhangjiajie, China [20]. This version includes a detailed security proof, a simulation experiment and a big variant which substantially improves the efficiency. This research was supported by the National Natural Science Foundation of China under Grants U1233108 and 61103207, the 2011 Korea-China Young Scientist Exchange Program, and the Fundamental Research Funds for Chinese Central Universities under Grant ZYGX2011J059, the Research Funds for Science &Technology Department of Sichuan Province under Grant 2012GZ0024, the Shanghai Science and Technology Committee under Grant 11511505300, and the National Research Foundation of Korea Grant funded by the Korea Government (MEST) (NRF-2010-0021575).

http://dx.doi.org/10.3837/tiis.2013.10.010

1. Introduction

Nowadays, more and more sensitive data, such as medical records, are being outsourced into the cloud to relieve the burden of data storage and maintenance. However, since the data servers in the cloud are not possibly trustworthy, sensitive data will be stored in encrypted form to protect privacy. On the other hand, encrypting data with ordinary cryptosystem seems to nullify the advantages of cloud computing since the utilization of the encrypted data such as search and operation would be extremely difficult. Are there any encryption schemes which are able to securely and effectively use the data stored in the cloud? The answer is "yes"! Fully homomorphic encryption (FHE) schemes allow anyone to compute publicly arbitrary functions on encrypted data without knowing the secret decryption key. Therefore, FHE can be used not only in the cloud computing, but also in the multitude of other scenarios where it is beneficial to keep all data encrypted and to perform computations on encrypted data. For example, in the wireless sensors network or the smart grid, FHE can be employed to achieve privacy-preserving data aggregation [1][2][3]. In the outsourcing computation, FHE can be used to construct short non-interactive zero-knowledge proofs [4][5], while in the database applications, FHE is capable of evaluating encrypted database indexing functions [6].

Early in 1978, Rivest et al. [7] presented the concept of FHE, but they failed to find a secure scheme. For more than 30 years, the functions that all known HE schemes supported were only limited, not arbitrary (full), and this restricted their applicability. During the period, the best result was given by Boneh et al. [8]. They proposed an HE scheme which can support arbitrary number of additions and one multiplication. However, constructing an FHE scheme supporting arbitrary functions was still an open problem.

In 2009, the old open problem was solved by the breakthrough work of Gentry based on ideal lattices [9]. However, Gentry's FHE scheme was not practical for many applications, because the computation time and ciphertext size were high-degree polynomials in security parameter λ. Stehlé et al. presented optimizations to Gentry's scheme that reduced its complexity from $\tilde{O}(\lambda^6)$ to $\tilde{O}(\lambda^{3.5})$ for per-gate computation [10]. Recently, Brakerski et al. offered another choice of FHE schemes based on the ring learning-with-error problem in which the per-gate computation was reduced to $\tilde{O}(\lambda^2)$ [11]. Also, Gentry gave a construction framework [12] that an FHE scheme can be easily transformed by applying the bootstrapping theorem regularly from a somewhat HE scheme, which can merely evaluate low-degree polynomials homomorphically. However, to support bootstrappable encryption, Gentry's framework has to squash the decryption circuit that results in the inefficiency of the construction. Therefore, recently a series of new works addresses methods that require no squashing. In particular, Brakerski and Vaikuntanathan [13] show how to obtain a direct construction of a bootstrappable encryption scheme without squashing. In a concurrent work, Gentry and Halevi [14] show how to get rid of squashing as well, using a completely different technique.

Originally, Gentry's somewhat HE scheme is constructed with ideal lattices over polynomial rings. In 2010, Dijk et al. proposed a very simple somewhat HE scheme using only addition and multiplication over the integers, which had merit of conceptual simplicity [15]. Its security was based on the hardness of approximate GCD of many integers - i.e., if a list of integers is provided that were near-multiples of a hidden integer, it outputs the hidden integer. In order to preserve the intractability of the problem, the public key size would be set to $\tilde{O}(\lambda^{10})$. Although Coron et al. reduced the public key size to $\tilde{O}(\lambda^7)$ by using encryption with a quadratic

form rather than with a linear form [16], it is still too large for most of practical applications. For example, in Coron et al.'s scheme, to achieve the security level of λ=72, the size of the public key becomes 802MB, while for the public key encryption over integers, usually the size of the public key should be no more than 1MB.

Our Contributions. In this paper, we propose a new form of somewhat HE scheme to reduce the public key size and a variation of the scheme to optimize the ciphertext size as follows.

- Firstly, we propose a new form of somewhat HE scheme to reduce the public key size to $\tilde{O}(\lambda^3)$. We build the scheme based on the hardness of approximate-GCD of two numbers, as opposed to that of many numbers. Consequently, our scheme can support the same level of security as Dijk et al.'s and Coron et al.'s, but with much smaller size of the public key.
- Secondly, we provide a variation of the proposed somewhat HE scheme optimized by applying modular reduction to reduce the ciphertext size, and analyze its evaluating ability.

Organization. The remainder of the paper is organized as follows. We present the related works in Section 2. Next, we introduce the notations and definitions in Section 3. Then, we propose a new somewhat HE scheme with security analysis, experiment simluation and performance comparison in Section 4. In Section 5, a variation of the new somewhat HE scheme is provided. Finally, we draw our conclusions in Section 6.

2. Related Work

After Gentry provides the construction framework of FHE [12], there are mainly two ways to construct the somewhat HE schemes. One uses ideal lattices [9] [10] [11], and the other is based on integers [15] [16] which have merit of conceptual simplicity. In this paper our work focuses on the latter, and thereby in this section we mainly review the construction techniques of the somewhat HE schemes adopted by Dijk et al. [15] and Coron et al. [16]. Furthermore, we briefly summarize our different ones in the somewhat HE schemes.

In Dijk et al.'s scheme, the message m is encrypted with a linear form in the public key elements x_i:

$$c = m + 2r + 2\sum_i x_i \bmod x_0, \tag{1}$$

where r is a random value. To thwart the attack to the underlying intractability problem, the bit-length and number of x_i are both considerably large. Hence, Coron *et al.* improve the encryption with a quadratic form instead of a linear form:

$$c = m + 2r + 2\sum_{i,j} b_{i,j} \cdot x_{i,0} \cdot x_{j,1} \bmod x_0, \tag{2}$$

where only a smaller subset of the public key needs to be stored and the full public key is then generated on the fly by combining the elements in the small subset multiplicatively. Here we encrypt a message with a new form:

$$c = m + 2r + 2r_1 \cdot x_1 \bmod x_0, \tag{3}$$

where r_1 is a new random value. Therefore, there are only two public key elements x_0 and x_1 in our scheme. Consequently, its security needs only to be based on approximate-GCD of two numbers, while both the above two schemes must build the security on that of many numbers. In order to resist the lattice attack to approximate-GCD of two numbers, ρ/γ needs only to be larger than $(\eta/\gamma)^2$, but for that of many numbers, setting $\gamma/\eta^2 = \omega(\log\lambda)$ is required. As a result, for the same security level, the size of the public key in our scheme is much smaller than that in Dijk et al.'s and Coron et al.'s.

Differences from conference version. Portions of the work presented in this paper have previously appeared as an extended abstract in [20]. This paper has improved many technical details as compared to [20]. The primary improvements are as follows: First, we provide a detailed security proof in Subsection 4.4, which is very important to public key encryption scheme. Second, we provide a big variant of the proposed somewhat HE scheme in Section 5, which substantially improves the efficiency of the original one. Finally, we provide a simulation result for the proposed somewhat HE scheme in Subsection 4.6, which shows that our scheme can achieve a good evaluating ability.

3. Preliminaries

In this section, we introduce the notations and definitions used throughout the remainder of this paper.

Notations. In the paper, we adopt the same notations as in [15]. Besides, the asymptotic notations [17], for example the big O, the big theta Θ, the small omega ω and etc., are used to analyze the algorithm efficiency. In addition, we write $f(\lambda) = \tilde{O}(g(\lambda))$, if $f(\lambda) = O(g(\lambda)\log^k g(\lambda))$ for positive integer k.The notations are shown as in **Table 1**.

Table 1. Notations

Notation	Meaning
Greek letters	Parameters (e.g., ρ, η, γ, etc.)
λ	Security parameter
Lowercase English letters	Real numbers and integers (e.g., p, x, etc.)
log	Logarithm of base-2
$[x]_p$	$-p/2 < x \bmod p \le p/2$
$x \leftarrow_R S$	Choosing an element x randomly from a set S

First, we introduce the definition of HE.

Definition 1 (Homomorphic Encryption). A homomorphic public key encryption scheme ε = (**KG**, **Enc**, **Dec**, **Eval**) has the following four algorithms.

- **KG**(λ): The algorithm is to generate keys by taking input with a security parameter λ, and output with a public/private key pair (pk, sk).
- **Enc**(pk, m): The algorithm is to encrypt a message by taking input with a public key pk and a message m, and output with a ciphertext c.
- **Dec**(sk, c): The algorithm is to decrypt a ciphertext by taking input with a private key sk and a ciphertext c, and output with a message m.

- **Eval**(pk, C, <c_1,…, c_t>): The algorithm is to evaluate a circuit by taking input with a public key pk, an evaluating circuit C with t-input, and a tuple of ciphertexts <c_1,…, c_t>, and output with another ciphertext c.

Roughly speaking, the somewhat HE scheme can only evaluate the limited circuits, while the FHE scheme can handle the arbitrarily computable ones. Here, we give the precise definition of somewhat HE scheme as follows, in which the **Eval** algorithm in **Definition 1** above can be treated as the **Add** and **Mult** algorithms to support the additive and multiplicative homomorphic operations, respectively.

Definition 2 (Somewhat Homomorphic Encryption). A somewhat homomorphic encryption scheme consists of the five algorithms (**KG**, **Enc**, **Dec**, **Add**, **Mult**).

- **KG**(λ): The same as in **Definition 1**.
- **Enc**(pk, m): The same as in **Definition 1**.
- **Dec**(sk, c): The same as in **Definition 1**.
- **Add**(c_1, c_2): The algorithm is to evaluate the additive homomorphic operation by taking input with two ciphertext c_1 and c_2, and output with another ciphertext c.
- **Mult**(c_1, c_2): The algorithm is to evaluate the multiplicative homomorphic operation by taking input with two ciphertext c_1 and c_2, and output with another ciphertext c.

Then, we give definitions of the correctness and compactness of the HE scheme $\boldsymbol{\varepsilon}$.

Definition 3 (Correctness). $\boldsymbol{\varepsilon}$ is correct for a given t-input circuit C, if it is the case that:

$$\mathbf{Dec}(sk, \mathbf{Eval}(pk, C, <c_1, ..., c_t>)) = C(m_1, ..., m_t), \tag{4}$$

for any public/private key pair (pk, sk) generated from **KG**(λ), any t plaintext bits m_1,…, m_t, and any tuple of ciphertexts <c_1,…, c_t> with c_i = **Enc**(pk, m_i).

Definition 4 (Compactness). $\boldsymbol{\varepsilon}$ is compact if there is a polynomial f such that, for every value of the security parameter λ, the decryption algorithm **Dec** can be expressed as a circuit D_{ε}, if the circuit size is at most $f(\lambda)$.

Finally, we give the definition of approximate-GCD problem of two numbers which our somewhat HE scheme is built on.

Definition 5 (Approximate-GCD). The (ρ,η,γ)-approximate-GCD problem of two numbers is: Output p, for a randomly chosen η-bit odd integer p, given two integers x_0 and x_1, where $x_i = pl_i + 2h_i$, $l_i \leftarrow Z \cap (0, 2^{\gamma}/p]$, and $h_i \leftarrow Z \cap (-2^{\rho}, 2^{\rho}]$, for $i \in \{0,1\}$.

4. Proposed Somewhat HE Scheme

In this section, we construct a somewhat HE scheme with a new encrypting form, analyze its correctness, prove its security based on the approximate-GCD problem of two numbers, and give an efficiency comparison with the other somewhat HE schemes over integers. Finally, the simulation results for the evaluating ability of the proposed scheme are provided. To begin, we give the parameter set used in our scheme.

4.1 Parameters

The parameters in our scheme are similar to those in [15], and thus firstly we recall those parameters (all polynomials in the security parameter λ):

- ρ is the bit-length of the noise
- ρ' is the bit-length of the second noise
- η is the bit-length of the private key
- γ is the bit-length of the integer in the public keys
- τ is the number of integers in the public keys

However, there are two important different aspects in our parameter set, which result in more efficient construction:

- The parameter τ is not necessary because the public key in our construction contains only two integers x_0 and x_1.
- To foil various lattice-based attacks [15], for the approximate-GCD problem of many numbers, Dijk et al. set parameter $\gamma = \omega(\eta^2 \log \lambda)$. However, for the problem of two numbers, we need only to set parameter $\gamma > O((3\eta^2)/(8\rho))$ where the lattice reduction yields nothing useful against security.

Remark 1. Howgrave-Graham attempted to solve the approximate-GCD problem of two numbers by using lattices and gave the following bound [18]:

$$\beta_0 < 1-(1/2)\alpha_0 - \sqrt{1-\alpha_0-(1/2)\alpha_0^2} - \varepsilon(h,\alpha_0). \tag{5}$$

According to the definition of Algorithm 14 in [18], we have $\eta = \gamma\alpha_0$ and $\rho = \gamma\beta_0$. Therefore, if we ignore $\varepsilon(h,\alpha_0)$, we have

$$\gamma < \left(3\eta^2\right)/(8\rho) + (\rho+\eta)/2. \tag{6}$$

As a result, in our construction, parameters of ρ, ρ', η, and γ must be set under the following constraints:

- $\rho = \omega(\log \lambda)$
- $\rho' = \rho + \omega(\log \lambda)$
- $\eta > \rho \cdot \Theta(\lambda \cdot \log \lambda^2)$
- $\gamma > O((3\eta^2)/(8\rho))$

Then, a convenient parameter set is $\rho = \lambda$, $\rho' = 2\lambda$, $\eta = \tilde{O}(\lambda^2)$, and $\gamma = \tilde{O}(\lambda^3)$.

4.2 Construction

The proposed somewhat HE scheme is consisted of four algorithms **KG**, **Enc**, **Dec** and **Eval** as shown from **Fig. 1** to **Fig. 4**.

Algorithm KG(λ)

Input: a security parameter λ.
Output: a public/private key pair (*pk*, *sk*).
Steps:

1. Choose randomly $p \leftarrow_R (2\mathbb{Z}+1) \cap [2^{\eta-1}, 2^{\eta})$.
2. Choose randomly $l_0, l_1 \leftarrow_R \mathbb{Z} \cap [0, 2^{\gamma}/p)$ and $h_0, h_1 \leftarrow_R (-2^{\rho}, 2^{\rho})$.
3. Compute $x_i = pl_i + 2h_i$ for $i = 0,1$ and assume that $|x_0| > |x_1|$.
4. Set $pk \leftarrow (x_0, x_1)$ and $sk \leftarrow p$.

Fig. 1. Key Generation Algorithm

Algorithm Enc(pk, m)

Input: a public key *pk* and a bit message $m \in \{0,1\}$.
Output: a ciphertext *c*.
Steps:

1. Choose randomly $r \leftarrow_R (-2^{\rho'}, 2^{\rho'})$ and $r_1 \leftarrow_R (-2^{\rho}, 2^{\rho})$.
2. Compute $c = m + 2r + r_1 x_1 \bmod x_0$.

Fig. 2. Encryption Algorithm

Algorithm Dec(sk, c)

Input: a private key *sk* and a ciphertext *c*.
Output: a bit message $m \in \{0,1\}$.
Steps:

1. Compute $m = (c \bmod p) \bmod 2$, where $(c \bmod p)$ gets a result of the integer in $(-p/2,\ p/2]$.

Fig. 3. Decryption Algorithm

Algorithm Eval(*pk*, *C*, <c_1,…, c_t>)

Input: a public key *pk*, an evaluating circuit *C* with *t*-input, and a tuple of ciphertexts <c_1,…, c_t>.
Output: another ciphertext *c*.
Steps:

1. Apply the integer addition and multiplication gates of the circuit *C* to the tuple of <c_1,…,c_t>, performing all the operations over the integers, and return the resulting ciphertext *c*. And the addition and multiplication on two ciphertexts are as follows:
   - **Add**(c_1, c_2): $c_1 + c_2$.
   - **Mult**(c_1, c_2): $c_1 \cdot c_2$.

Fig. 4. Evaluation Algorithm

Remark 2. In the public key $pk=(x_0,x_1)$, x_0 and x_1 cannot both be even integers. Otherwise, the message m and the ciphertext c have the same parity.

Remark 3. Note that while the **Enc** algorithm reduces the length of the ciphertext by applying modulo x_0, we cannot do the same in the **Eval** algorithm, which results in significant expansion of ciphertext size with the complexity of the evaluating circuit C. To ensure the compactness (Definition 4), the optimizing techniques of modular reduction in the variant are adopted in Section 5.

4.3 Correctness

As mentioned in Definition 3, for an HE scheme, the correctness needs to be preserved. Therefore, in this subsection, we check the correctness of the **Eval** algorithm based on the analysis of the noise of a fresh ciphertext. Then, the evaluating ability of our somewhat HE scheme is given.

- Firstly, we consider the noise of a fresh ciphertext output by the **Enc** algorithm.

According to **Enc**, $c=m+2r+r_1x_1 \bmod x_0$. And since $|x_0|>|x_1|$, we have that

$$c=m+2r+r_1x_1+kx_0,\ |k|<r_1. \tag{7}$$

Since $x_i=pl_i+2h_i$ for $i=0,1$, we have that

$$c=p\cdot(r_1l_1+kl_0)+(m+2r+2r_1h_1+2kh_0). \tag{8}$$

With regard to the noise $m+2r+2r_1h_1+2kh_0$, its parity is the same as m. According to the parameters in our scheme,

$$|m+2r+2r_1h_1+2kh_0|<2\cdot 2^{\rho'}+2\cdot 2^{\rho}\cdot 2^{\rho}+2\cdot 2^{\rho}\cdot 2^{\rho}=3\cdot 2^{\rho'+1}, \tag{9}$$

and thus the noise is at most $3\cdot 2^{\rho'+1}$ in absolute value. For the simplicity, we write $m+2r+2r_1h_1+2kh_0$ as $m+2b$ for a certain integer b.

- Secondly, let us check that the ciphertext output by the **Eval** algorithm can be decrypted correctly.

Let C' be the generalized circuit corresponding to C, which operates over the integers rather than modulo 2. Let $c=C'(c_1,\cdots,c_t)$, where the noise of c_i is m_i+2b_i for $i\in\{1,\cdots,t\}$, which has the same parity as the message m_i. We have that

$$c=C'(m_1+2b_1,\cdots,m_t+2b_t)+p\mathbb{Z}. \tag{10}$$

If the noise is small enough as

$$|C'(m_1+2b_1,\cdots,m_t+2b_t)|\le 2^{\eta-4}<p/8, \tag{11}$$

we have that

$$\left[\left[C'(m_1+2b_1,\cdots,m_t+2b_t)\right]_p\right]_2 = C(m_1,\cdots,m_t) = m\,. \tag{12}$$

Remark 4. The bound $2^{\eta-2} < p/2$ would be sufficient for correct decryption. However, in order to make the squashed decryption circuit shallower [15], here the noise may remain below $p/8$.

- Finally, we consider the degree of polynomials which our somewhat HE scheme $\boldsymbol{\varepsilon}$ can evaluate. We have the following theorem.

Theorem 1. *Let C be a boolean circuit with t inputs, and C' be the same circuit as C, but with boolean gates replaced by integer operations. Let* $f(x_1,\cdots,x_t)$ *be the multivariate polynomial corresponding to C', and d be its degree. If*

$$|f(x_1,...,x_t)| \le \left|\vec{f}\right|(3\cdot 2^{\rho'+1})^d \le 2^{\eta-4} < p/8\,, \tag{13}$$

where $\left|\vec{f}\right| = \sum_i |b_i|$, *and* b_i *is the coefficient of* f, *then ciphertexts output by* ***Eval*** *can be decrypted correctly.*

Proof. Equation (13) could be derived immediately from Equations (9) and (11): Since fresh ciphertexts output by the **Enc** algorithm have noise at most $3\cdot 2^{\rho'+1}$, the ciphertext output by the **Eval** algorithm applied to a circuit has noise at most $2^{\eta-4} < p/8$. Furthermore, by considering the multivariate polynomial $f(x_1,\cdots,x_t)$, we can get Equation (13). □

In particular, $\boldsymbol{\varepsilon}$ can handle f as long as

$$d \le \frac{\eta - 4 - \log\left|\vec{f}\right|}{\rho' + 1 + \log 3}\,. \tag{14}$$

4.4 Security

Our security proof to Theorem 2 is similar to that of Dijk et al.'s. The main difference is that our construction of the somewhat HE scheme is on the approximate-GCD of two numbers, not on that of many numbers.

Theorem 2. *Let* A *be an attacker with advantage* $\grave{o}$ *against our somewhat HE scheme with parameters (ρ,ρ',η,γ) of polynomials in the security level λ. There exists an algorithm* B *for solving the (ρ,η,γ)-approximate-GCD problem of two numbers that succeeds with probability at least* $\grave{o}/2$. *The running time of* B *is polynomial in the running time of* A, λ, *and* $1/\grave{o}$.

Proof. Given an algorithm A that breaks the above somewhat HE scheme, we show how to construct an algorithm B that solves the approximate-GCD problem. The two integers $x_i = pl_i + 2h_i$, $i = 0,1$ are given to the solver B, for a randomly chosen η-bit odd integer p,

where $l_i \leftarrow \mathbb{Z} \cap [0, 2^{\gamma}/p)$ and $h_i \leftarrow \mathbb{Z} \cap (-2^{\rho}, 2^{\rho})$; its goal is to find p. Here, for an integer z, we use $q_p(z)$ and $r_p(z)$ to denote the quotient and remainder of z with respect to p, i.e., $x = q_p(z) \cdot p + r_p(z)$, $r_p(z) \in (-p/2, p/2]$.

Step 1: Key Generation.

When x_0 and x_1 are given to the solver B, it relabels them as $|x_0| > |x_1|$. B then outputs a public key $pk = (x_0, x_1)$. Clearly, if x_0 and x_1 are not both even integers, the distribution induced on the public key in the simulation is identical to that in the original scheme.

Step 2: Prediction for Least-Significant Bit (LSB).

B produces an integer z by "encryption of zero." Then B attempts to recover p by using A to learn the LSB of $q_p(z)$, where we denote the parity of z by $\mathbf{parity}(z)$. The procedure of prediction is shown in **Fig. 5**.

Subroutine Learn-LSB(z, pk)

Input: $z \in [0, 2^{\gamma})$ with $|r_p(z)| < 2^{\rho}$ and a public key $pk = (x_0, x_1)$.

Output: LSB of $q_p(z)$.

Steps:

1. Choose randomly $r \leftarrow_R (-2^{\rho'}, 2^{\rho'})$, $r_1 \leftarrow_R (-2^{\rho}, 2^{\rho})$, and a bit $m \leftarrow_R \{0,1\}$.
2. Set $c \leftarrow z + m + 2r + r_1 x_1 \bmod x_0$.
3. Call A to get a prediction $a \leftarrow \mathrm{A}(pk, c)$.
4. Set $b \leftarrow a \oplus \mathbf{parity}(z) \oplus m$.
5. Output b.

Fig. 5. Learn-LSB Subroutine

Remark 5. Since z can be represented as $z = q_p(z) \cdot p + r_p(z)$, and $a = [[c]_p]_2$, we have $a = r_p(z) \oplus m$. Furthermore, we have $b = a \oplus \mathbf{parity}(z) \oplus m$. So, b is the parity of $q_p(z)$. Additionally, to predict b with the noticeable advantage, B could choose many random z_i's corresponding to b_i's and take majority of votes among b_i's [15].

Step 3: Binary GCD and Recovering *p*.

After obtaining the parity of $q_p(z)$ by utilizing A, we recover p by the binary GCD algorithm described as in [19].

The Success Probability of B

Let G denote the event that x_0 and x_1 cannot both be even integers. Conditioned on G in our reduction, the distribution of the public key that B creates is identical to the right distribution from our scheme. It is easy to know that the event G happens with probability $3/4$. Similarly, we define P and PK_p.

Let P denote the set of odd integers in $[2^{\eta-1}, 2^{\eta})$ for which A has over $\grave{o}/2$ advantage:

$$\mathrm{P} = \{p \in [2^{\eta-1}, 2^{\eta}) : \textbf{Advantage}(\mathrm{A}) \text{ conditioned on } sk = p \text{ is at least } ?/2\}. \quad (15)$$

Let PK_p be the set of public keys for which A has advantage at least $3\grave{o}/8$:

$$\mathrm{PK}p = \{pk \text{ for } p: \textbf{Advantage}(\mathrm{A}) \text{ conditioned on } pk \text{ is at least } 3?/8\}. \quad (16)$$

Then, we analyze the distribution of ciphertexts produced in step 2 of the subroutine **Learn-LSB**. In our scheme, $c = m + 2r + r_1 x_1 \bmod x_0$, while in the security reduction, $c' \leftarrow [x^* + m + 2r + r_1 x_1]_{x_0}$. According to parameter choices, $r_1 \leftarrow_R (-2^{\rho}, 2^{\rho})$ and $x_1 \in [0, 2^{\gamma})$. Therefore, whatever the binary GCD algorithm operations, $x^* \in [0, 2^{\gamma})$, and thus the distribution of $r_1 x_1$ is close to that of $x^* + r_1 x_1$. Hence, the ciphertext in the reduction produces a distribution which is statistically close to that of our scheme. As a result, with the probability $3\grave{o}/8 - \textbf{negl}$ (we denotes the amount neglectable in λ by **negl**) the ciphertext-generation in the reduction "works" for this public key.

Finally, if x_0 and x_1 are given, B recovers the hidden secret $p \in \mathrm{P}$ in a single run with probability at least $\frac{1}{2} \cdot (3\grave{o}/8 - \textbf{negl})$. By repeating it for $16/(3\grave{o}) \cdot \omega(\log \lambda)$ times, we will recover such p with overwhelming probability. Hence, the overall success probability of B is at least $\grave{o}/2$ with complexity which is polynomial in λ and $1/\grave{o}$. □

4.5 Comparision

As shown in **Table 2**, considering the evaluating degree of the polynomial, our scheme is similar to Dijk et al.'s [15] and Coron et al.'s [16]. However, the public key size in our scheme is much smaller than theirs. The public key size in our scheme is $\tilde{O}(\lambda^3)$, while it is $\tilde{O}(\lambda^{10})$ and $\tilde{O}(\lambda^7)$ in Dijk et al.'s and Coron et al.'s, respectively. This is mainly due to the based hardness problem. We construct our somewhat HE scheme on the approximate-GCD of two numbers as opposed to that of many numbers in other schemes.

Table 2. Comparison of somewhat HE schemes

Scheme	*Hardness problem*	*Public keys size*	*Evaluating degree*
Dijk *et al.*'	approximate-GCD of many Numbers	$\tilde{O}(\lambda^{10})$	$\frac{\eta - 4 - \log\lvert\vec{f}\rvert}{\rho' + 2}$
Coron *et al.*'	approximate-GCD of many Numbers	$\tilde{O}(\lambda^{7})$	$\frac{\eta - 3 - \log\lvert\vec{f}\rvert}{\rho' + 2 + 2\log\beta}$
Ours	approximate-GCD of two Numbers	$\tilde{O}(\lambda^{3})$	$\frac{\eta - 4 - \log\lvert\vec{f}\rvert}{\rho' + 1 + \log 3}$

4.6 Simulation Experiments

In this subsection, we give the simulation results for the proposed somewhat HE scheme. We run the experiments on a Thinkpad Notebook, featuring an Intel CPU P8400 (2.26GHz), with 3GB of random access memories. Our implementation uses Shoup's NTL library [21] version 5.5.2 for high-level numeric algorithms. We consider the evaluating degree d of the polynomial, which represents the evaluating ability of the somewhat HE scheme. As shown in **Table 3**, on one hand, the degree d increases with the security parameter λ. It means the evaluating ability of our scheme can grow normally with λ. On the other hand, d is almost the same even if the different message length l is considered. It means that our scheme can achieve steady evaluating ability with respect to l. Consequently, our scheme can provide a good evaluating ability.

Table 3. Evaluating ability of our scheme
(λ : security parameter, l: message length, d: evaluating degree)

d	l = 64	l = 112	l = 160	l = 208	l = 256
λ =42	21	21	22	21	21
λ =52	28	28	26	26	26
λ =62	31	31	33	31	34
λ =72	39	36	37	37	36

5. Variation and Optimization

Dijk et al. also considered a variation of somewhat HE scheme by applying modular reduction into the **Eval** algorithm to reduce the ciphertext size [15]. However, they did not give the evaluating ability of the variation. Obviously, the degree d of the variation is lowered due to the introduction of the additional noises. In this section, we analyze the evaluating ability of the variation of our scheme. Here, we denote the optimized **Eval** algorithm by **Opt-Eval**.

5.1 Opt-Eval

We apply the modular-reduction into **Eval** of our basic scheme. Firstly, the $\gamma+1$ random elements $x_i' = pl_i' + 2h_i'$, $i = 0,1,...,\gamma$ are added to the public key, in which $l_i' \leftarrow_R Z \cap [2^{\gamma+i-1}/p, 2^{\gamma+i}/p)$ and $h_i' \leftarrow_R Z \cap (-2^{\rho}, 2^{\rho})$. Then, we have $x_i' \in [2^{\gamma+i-1}, 2^{\gamma+i})$.

In the basic **Eval** algorithm, $\mathbf{Add}(c_1, c_2)$ increases the magnitude of the integers by at most a factor of 2. However, $\mathbf{Mult}(c_1, c_2)$ may square the magnitude of the integers. Therefore, we optimize them differently in the **Opt-Eval** algorithm as shown in **Fig. 6**.

Algorithm Opt-Eval(pk, C, $<c_1,\ldots, c_t>$)

Input: a public key pk, an evaluating circuit C with t-input, and a tuple of ciphertexts $<c_1,\ldots, c_t>$.
Output: another ciphertext c.
Steps:
1. The optimized addition and multiplication on two ciphertexts are as follows:
- $\mathbf{Opt}(\mathbf{Add}(c_1, c_2)) = (c_1 + c_2) \bmod x_0$.
- $\mathbf{Opt}(\mathbf{Mult}(c_1, c_2)) = \left(\cdots\left(\left((c_1 \cdot c_2) \bmod x_\gamma'\right) \bmod x_{\gamma-1}'\right)\cdots\right) \bmod x_0'$.

Fig. 6. Optimized Evaluation Algorithm

5.2 Additional Noises in Opt-Eval

Let c_1 and c_2 be ciphertexts of m_1 and m_2, respectively:

$$c_1 = m_1 + 2r_{01} + r_{11}x_1 \bmod x_0, \qquad c_2 = m_2 + 2r_{02} + r_{12}x_1 \bmod x_0 . \tag{17}$$

We have

$$c_1 = m_1 + 2r_{01} + r_{11}x_1 + k_1x_0, \qquad c_2 = m_2 + 2r_{02} + r_{12}x_1 + k_2x_0 . \tag{18}$$

Let k_{Add} be a multiple of x_0, and $k_{Mult,i}$ be a multiple of x_i', $i = 0,1,...,\gamma$. Let n_1 and n_2 be noises of c_1 and c_2, respectively. Let n_{Add} and n_{Mult} be additional noises introduced in $\mathbf{Opt}(\mathbf{Add}(c_1,c_2))$ and $\mathbf{Opt}(\mathbf{Mult}(c_1,c_2))$, respectively. We analyze n_{Add} and n_{Mult} as below.

- For $\mathbf{Opt}(\mathbf{Add}(c_1,c_2))$:

Firstly, n_{Add} can be represented as $n_{Add} = n_1 + n_2 + 2k_{Add}x_0$. Then, since $c_1 + c_2 \bmod x_0 = c_1 + c_2 + k_{Add}x_0 \Rightarrow |k_{Add}x_0| \le x_0$, we have $|k_{Add}| \le 1$. Thus we have the condition

$$|n_{Add}| \le |n_1| + |n_2| + 2^{\rho+1}. \tag{19}$$

- For $\mathbf{Opt}(\mathbf{Mult}(c_1,c_2))$:

To analyze the noise in $\mathbf{Opt}(\mathbf{Mult}(c_1,c_2))$, firstly we consider modulo x_γ'. We have $(c_1 \times c_2) \bmod x_\gamma' = c_1 \times c_2 + k_{Mult,\gamma}x_\gamma'$. Since one multiplication operation on two ciphertexts at most doubles the bit length of one ciphertext, we have $\left| k_{Mult,\gamma}x_\gamma' \right| = \left| (c_1 \times c_2) \bmod x_\gamma' - (c_1 + c_2) \right| \le 2x_\gamma'$, which means that $|k_{Mult,\gamma}| \le 2$. Secondly, we consider modulo $x_{\gamma-1}'$. We have $\left(c_1 \times c_2 \bmod x_\gamma'\right) \bmod x_{\gamma-1}' = c_1 \times c_2 + k_{Mult,\gamma}x_\gamma' + k_{Mult,\gamma-1}x_{\gamma-1}'$. Since $\left| (c_1 \times c_2) \bmod x_\gamma' \right| \le x_\gamma'/2 < 2^{2\gamma-1} < 2x_{\gamma-1}'$ and $\left|\left((c_1 \times c_2) \bmod x_\gamma'\right) \bmod x_{\gamma-1}'\right| \le x_{\gamma-1}'/2$, we have $|k_{Mult,\gamma-1}| \le 2$. Similarly, we have $|k_{Mult,i}| \le 2, i = 0,1,...,\gamma-2$. And since n_{Mult} can be represented as $n_{Mult} = n_1 \cdot n_2 + 2\sum_{i=0}^{\gamma} k_{Mult,i}r_i'$, we have

$$|n_{Mult}| < |n_1| \cdot |n_2| + (\gamma+1) \cdot 2^{\rho+2}. \tag{20}$$

As analyzed above, the additional noises are introduced from $\mathbf{Opt}(\mathbf{Add}(c_1,c_2))$ and $\mathbf{Opt}(\mathbf{Mult}(c_1,c_2))$, which could lower the evaluating ability of our variation. In the following, we further consider the degree *d* of the evaluated polynomial.

5.3 Evaluating Ability of Opt-Eval

For the sake of the convenience, let $X = 3 \cdot 2^{\rho'+1}$ be the noise in the fresh ciphertext, $A = 2^{\rho+1}$ be the additional noise in $\mathbf{Opt}(\mathbf{Add}(c_1, c_2))$, and $B = (\gamma+1) \cdot 2^{\rho+2}$ be the additional noise in $\mathbf{Opt}(\mathbf{Mult}(c_1, c_2))$.

Firstly, we analyze the noise in the monomial of degree *d*: $x_{i_1} x_{i_2} x_{i_3} \cdots x_{i_d}$. For $\mathbf{Opt}(\mathbf{Mult}(c_1, c_2))$, we have $|n_{Mult}| \lhd |n_1| \cdot |n_2| + B$. Hence, the noise of $x_{i_1} x_{i_2} x_{i_3} \cdots x_{i_d}$ is $(\cdots(((X \cdot X + B) \cdot X + B) \cdot X + B) \cdots) X + B$. By "Mathematical Induction" method, we have

$$X^d + BX^{d-1} + \cdots + BX + B = X^d + B \cdot \frac{X^{d-1} - 1}{X - 1}. \tag{21}$$

Then, we consider any polynomial of degree *d*: $f(x_1, ..., x_t)$. There are at most $\left|\vec{f}\right|$ monomials of degree *d* in $f(x_1, ..., x_t)$. Thus, we have

$$\begin{aligned} &\left|\vec{f}\right|(X^d + B \cdot \frac{X^{d-1} - 1}{X - 1}) + \left(\left|\vec{f}\right| - 1\right) A \le 2^{\eta-4} < p/8 \\ &\Rightarrow \left|\vec{f}\right|(X^d + B \cdot \frac{X^{d-1} - 1}{X - 1}) + \left|\vec{f}\right| A \le 2^{\eta-4} + A \\ &\Rightarrow X^d (1 + \frac{B}{X(X-1)}) \le \frac{2^{\eta-4} + A}{\left|\vec{f}\right|} - A + \frac{B}{X-1} \quad . \end{aligned} \tag{22}$$

Furthermore, we can get

$$d \le \frac{\log(\frac{2^{\eta-4} + A}{\left|\vec{f}\right|} - A + \frac{B}{X-1}) - \log(1 + \frac{B}{X(X-1)})}{\log X}. \tag{23}$$

Since $X = 3 \cdot 2^{\rho'+1}$, $A = 2^{\rho+1}$, $B = (\gamma+1) \cdot 2^{\rho+2}$, we have

$$\frac{B}{X-1} \approx \frac{\gamma+1}{3 \cdot 2^{\rho-1}} \text{ and } \frac{B}{X(X-1)} \approx \frac{\gamma+1}{9 \cdot 2^{3\rho}}, \tag{24}$$

which can be both omitted according to the parameter setting of our scheme.

Finally, we have

$$d \le \frac{\log(\frac{2^{\eta-4} + 2^{\rho+1}}{\left|\vec{f}\right|} - 2^{\rho+1})}{\rho' + 1 - \log 3}. \tag{25}$$

Remark 6. As analyzed, the proposed variation is effective and efficient. With respect to the effectiveness, we have focused on the correctness of the proposed variation. As shown in Section 5, by analyzing the evaluating ability of the variation, we have checked the correctness of the variation, based on the analysis of additional noises of ciphertexts. Therefore, the variation of our scheme can achieve the effectiveness. On the other hand, we have considered the efficiency. As we know, if the technique of the modular-reduction is not used, the operations on ciphertexts would increase significantly the size of the ciphertext as shown in original scheme, since just one multiplication the ciphertext doubles the bit-length. While in the proposed variation, the size of the ciphertext is no more than that of the modulus due to the modular-reduction. As a result, the variation of our scheme could optimize the ciphertext size greatly.

6. Conclusion

In this paper, we proposed a new somewhat HE scheme over integers with small public key size $\tilde{O}(\lambda^3)$. Its security is based on approximate-GCD of two integers. Besides, we gave a variation of our scheme and analyzed its evaluating ability. In the future work, we will further improve the efficiency of the FHE scheme to be practical for cloud computing and other related applications.

References

[1] J. Shi, R. Zhang, Y. Liu, and Y. Zhang, "Prisense: privacy-preserving data aggregation in people-centric urban sensing systems," in *Proc. of INFOCOM*, 2010, pp. 1–9. Article (CrossRef Link)

[2] H. Li, X. Lin, H. Yang, X. Liang, R. Lu, and X. Shen, "EPPDR: An Efficient Privacy-Preserving Demand Response Scheme with Adaptive Key Evolution in Smart Grid, " *IEEE Transactions on Parallel and Distributed Systems*, 2013, to appear. Article (CrossRef Link)

[3] R. Lu, X. Liang, X. Li, X. Lin, and X. Shen, "Eppa: An efficient and privacy preserving aggregation scheme for secure smart grid communications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 9, pp. 1621–1631, 2012. Article (CrossRef Link)

[4] R. Gennaro, C. Gentry, and B. Parno, "Non-interactive verifiable computing: Outsourcing computation to untrusted workers," in *Proc. of Advances in Cryptology - CRYPTO 2010*, Springer Berlin / Heidelberg, LNCS 6223, 2010, pp. 465–482. Article (CrossRef Link)

[5] K.-M. Chung, Y. T. Kalai and S. P. Vadhan, "Improved delegation of computation using fully homomorphic encryption," in *Proc. of Advances in Cryptology - CRYPTO 2010*, Springer Berlin / Heidelberg, LNCS 6223, 2010, pp. 483–501. Article (CrossRef Link)

[6] Y. Gahi, M. Guennoun and K. El-Khatib, "A secure database system using homomorphic encryption schemes" in *Proc. of The Third International Conference on Advances in Databases, Knowledge, and Data Applications*. 2011, pp. 54-58. http://www.thinkmind.org/index.php?view=article&articleid=dbkda_2011_3_20_30074

[7] R. L. Rivest, L. M.Adleman, and M. L.Dertouzos, "On data banks and privacy homomorphisms," in *Proc. of Foundations of Sec. Comp.*, 1978, pp.169-180. http://www.citeulike.org/user/deitosrafael/article/3877157

[8] D. Boneh, E.-J Goh, and K. Nissim, "Evaluating 2-DNF Formulas on Ciphertexts," in *Proc. of Theory of Cryptography Conference*, Springer Berlin / Heidelberg, LNCS 3378, 2005, pp.325-341. Article (CrossRef Link)

[9] C. Gentry, "Fully homomorphic encryption using ideal lattices," *STOC 2009*, ACM, 2009, pp.169-178. Article (CrossRef Link)

[10] D. Stehlé, and R. Steinfeld, "Faster Fully Homomorphic Encryption," in *Proc. of Advances in Cryptology - ASIACRYPT 2010*, Springer Berlin / Heidelberg, LNCS 6477, 2010, pp.377-394.

Article (CrossRef Link)
[11] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "Fully Homomorphic Encryption without Bootstrapping," http://eprint.iacr.org/2011/
[12] C. Gentry, "A fully homomorphic encryption scheme," Stanford University, PhD Thesis, 2009.
[13] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) LWE," in *Proc. of IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS)*, 2011, pp. 97-106. Article (CrossRef Link)
[14] C. Gentry and S. Halevi, "Fully homomorphic encryption without squashing using depth-3 arithmetic circuits," in *Proc. of IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS)*, 2011, pp. 107-109. Article (CrossRef Link)
[15] M. Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully Homomorphic Encryption over the Integers," in *Proc. of Advances in Cryptology – EUROCRYPT 2010*, Springer Berlin / Heidelberg, LNCS 6110, 2010, pp. 24-43. Article (CrossRef Link)
[16] J.-S. Coron, A. Mandal, D. Naccache, and M. Tibouchi, "Fully Homomorphic Encryption over the Integers with Shorter Public Keys," in *Proc. of Advances in Cryptology - CRYPTO 2011*, Springer Berlin / Heidelberg, LNCS 6841, 2011, pp.487-504. Article (CrossRef Link)
[17] D. E. Knuth. Asymptotic Representations, volume 1 of *The Art of Computer Programming*, Addison-Wesley, 3rd edition, 1997. http://www.amazon.com/Art-Computer-Programming-Volume-Fundamental/dp/0201896834
[18] N. Howgrave-Graham, "Approximate integer common divisors," in *Proc. of CaLC' 01*, Springer, LNCS 2146, 2001, pp.51-66. http://dl.acm.org/citation.cfm?id=753508
[19] D. E. Knuth. Seminumerical Algorithms, volume 2 of *The Art of Computer Programming*, Addison-Wesley, 3rd edition, 1997. http://www.amazon.com/Art-Computer-Programming-Volume-Seminumerical/dp/0201896842
[20] H. Yang, D. Tang, Q. Xia, and X. Wang, "A New Somewhat Homomorphic Encryption Scheme over Integers," in *Proc. of 2012 International Conference on Computer Distributed Control and Intelligent Environmental Monitoring*, CDCIEM 2012, 2012, pp. 61-64. Article (CrossRef Link)
[21] V. Shoup, NTL: A Library for doing Number Theory. http://shoup.net/ntl/

Haomiao Yang, received his M.S. and Ph.D. degrees in Computer Applied Technology from University of Electronic Science and Technology of China (UESTC) in 2004 and 2008 respectively. He is an associate professor in Network Security Technology Laboratory in UESTC. Currently, he is doing post-doctoral research in the Research Center of Information Cross-over Security, Kyungil University, Republic of Korea. His research interests include cryptography, cloud security, and the cyber-security for aviation communications.

Hyunsung Kim received his M.S. and Ph.D. degrees in Computer Engineering from Kyungpook National University, Republic of Korea, in 1998 and 2002, respectively. From 2000 to 2002, he worked as a senior researcher at Ditto Technology. He had been an associate professor from 2002 to 2011 with the Department of Computer Engineering, Kyungil University. Currently, he is an associate professor at the Department of Cyber Security, Kyungil University, Republic of Korea. His current research interests are cryptography, VLSI, authentication technologies, network security and ubiquitous computing security.

Dianhua Tang received his B.S. degree in Applied Mathematics from Xidian University, China and the M.S. degree in Communication & Information System from Science & Technology on Communication Security Laboratory, China in 2009 and 2012, respectively. Currently, he is an engineer in Science & Technology on Communication Security Laboratory, China. His research interests include cryptography and network security.

Hongwei Li received his M.S. degree in Computer Application from Southwest Jiaotong University (SWJTU) and Ph.D. degree in Computer Software and Theory from University of Electronic Science and Technology of China (UESTC) in 2004 and 2008 respectively. From 2011 to 2012, he worked as a Postdoctoral Fellow at University of Waterloo, Canada. Currently, he is an associate professor at the School of Computer Science and Engineering, UESTC, China. His research interests include cryptography, and the secure smart grid.