New Fully Homomorphic Encryption over the Integers

Gu Chunsheng
School of Computer Engineering
Jiangsu Teachers University of Technology
Changzhou, China, 213001
guchunsheng@gmail.com

Abstract: This paper presents a new fully homomorphic encryption scheme over the integers,which is different from the fully homomorphic encryption scheme in [vDGHV10], but the somewhat homomorphic encryption is similar to that in [vDGHV10]. By using the self-loop bootstrappable technique, a ciphertext is refreshed to a new ciphertext with same message of an original ciphertext and smaller error terms. The size of ciphertext is remained fixed and the expansion of ciphertext is $O(n^2)$ in our scheme.
The security of our scheme is based on the hardness of finding an approximate-GCD problem over the integers, which is given a list of integers perturbed by the small error noises.
Keywords: Fully Homomorphic Encryption, Approximate-GCD, GCD.

1. Introduction

The conclusion in [vDGHV10] said, “The primary open problem is to improve the efficiency of the scheme, to the extent that it is possible while preserving the hardness of the approximate-GCD problem”. In this paper, we affirmatively solve this open problem by the self-loop bootstrappable technique. The public key in our scheme is a list of approximate multiples $\{b_i = a_i p + 2e_i\}_{i=1}^{\tau}, \tau = O(n)$ for an odd integer p, where a_i, e_i is the uniform random integers over Z such that $|e_i| < 2^{n-1}$. The secret key is p. To encrypt a message bit m, the ciphertext is computed as $c = \sum_{i \in T, T \subseteq [\tau]} b_i + 2e + m$, where $|e| < 2^{n-1}$. To decrypt, we compute the message bit $m = [c]_p \bmod 2$. Recall that $[z]_p$ is an integer in $(-p/2, p/2)$ throughout this paper.

It is not difficult to verify that the above scheme has additively and multiplicatively homomorphic. Furthermore, we can use the self-loop bootstrappable technique to get a new fully homomorphic encryption scheme from this simple scheme.

1.1 Our Contribution

Our somewhat homomorphic encryption (SHE) is similar to their SHE [vDGHV10], but our fully homomorphic encryption (FHE) is complete different from theirs. The main difference between two schemes is efficiency and security. The size of the public key in our scheme is $O(n^3)$ bits, the expansion factor of ciphertext is $O(n^2)$. As the scheme in [vDGHV10], the security of our scheme also depends on the hardness of finding an approximate-GCD over the integers, given a list of approximate multiples of p, but the noise of ciphertext in our scheme is bigger than theirs.

1.2 Related work

Since [RAD78] introduced a privacy homomorphism, many researchers [BGN05, ACG08, SYY99, Yao82] attempted to solve this open problem. [Gen09] constructed the first fully homomorphic encryption based on ideal lattice. In Gentry's scheme, the public key is approximately n^7 bits, the computation per gate costs $O(n^6)$ operations. [SV10] presented a fully homomorphic encryption scheme with both relatively small key $O(n^3)$ bits , ciphertext size $O(n^{1.5})$ bits and computation per gate at least $O(n^3)$ operations, which is in some sense a specialization and optimization of Gentry's scheme. [vDGHV10] proposed a simple fully homomorphic encryption scheme over the integers whose security is based on the hardness of finding an approximate integer gcd. [SS10] improved Gentry's fully homomorphic scheme based on ideal lattices and leaded to a faster fully homomorphic scheme, with $O(n^{3.5})$ bits complexity per elementary binary addition/multiplication gate.

1.3 Outline

Section 2 recalls some notations, and defines the homomorphic encryption. Section 3 presents a somewhat homomorphic encryption. Section 4 transforms the somewhat homomorphic encryption into a fully homomorphic encryption. Section 5 gives the security of scheme. Section 6 concludes this paper.

2. Preliminaries

Notations. To simplify the notation, all computation is over the field F_2 throughout this paper. Let n be a security parameter, $\tau = O(n)$. $[n] = \{0,1,...,n\}$. $w \leftarrow_U S$ chooses an

uniformly random element w in the set S.

Homomorphic Encryption. We adapt the definitions of [Gen09] for our homomorphic encryption scheme. We only consider the boolean circuit including gates for addition and multiplication mod 2. A homomorphic public key encryption scheme consists of four algorithms: the key generating algorithm **KeyGen**, the encrypting algorithm **Enc** and the decryption algorithm **Dec**, and an additional algorithm **Evaluate**. The **Evaluate** takes as input a public key pk, a t-input circuit C with the gates of addition and multiplication, and any ciphertext $\vec{c} =< c_1,...,c_t >$ with $c_i = Enc(pk, m_i)$, and outputs a new ciphertext $c = Evaluate(pk, C, \vec{c})$ such that $Dec(sk, c) = C(m_1,...,m_t)$.

Our fully homomorphic encryption scheme (FHE) computes arbitrary circuits of any depth by applying the self-loop bootstrapple encryption technique.

3. Somewhat Homomorphic Encryption (SHE)

We present a somewhat homomorphic encryption and simply analyze its performace.

3.1 Construction

Key Generating Algorithm (KeyGen).

(1) Select an odd integer $p > 2^{n^2+3}$ such that $s \approx 1/p$, $sp = 1 + O(2^{-n^2-7})$, and $h(s) = O(\log n)$, where $h(s)$ is the number of 1 in the binary representation of s.

(2) Pick random integers $a_i \in (2^{O(n)}, 2^{n^2})$ subject to the largest a_0 is an odd integer, $e_i \in Z, i \in [\tau]$ with $|e_i| < 2^{n-1}$. Then compute $b_0 = a_0 p + 2e_0$, and $[b_i = a_i p + 2e_i]_{b_0}$.

The public key is $pk = (n, \{b_i\}_{i=0}^{\tau})$, the secret key is $sk = (p)$.

Encryption Algorithm (Enc). Given the public key pk and an message bit $m \in \{0,1\}$, choose a random subset $T \subseteq [\tau]$ and an independent perturbed error polynomial e with $|e| < 2^{n-1}$. Compute the ciphertext $c = \left[\sum_{i \in T} b_i + 2e + m\right]_{b_0}$.

Add Operation (Add). Given the public key pk, and the ciphertexts c_1, c_2, evaluate the

ciphertext $c = [c_1 + c_2]_{b_0}$.

Multiplication Operation (Mul). Given the public key pk , and the ciphertexts c_1, c_2 , evaluate the ciphertext $c = [Opt(c_1 \times c_2)]_{b_0}$, where Opt is same as that in [vDGHV10].

Decryption Algorithm (Dec). Given the secret key sk , and a ciphertext c , decipher $m = [c]_p \bmod 2$.

3.2 Performance of SHE

The size of the public key $pk = (n, \{b_i\}_{i=0}^{\tau})$ is $O(n^3)$ bits, the size of secret key $sk = (s)$ is $O(n^2)$. The running times of **Enc**, **Dec**, **Add**, **Mul** are $O(n^3)$, $O(n^2)$, $O(n^2)$, and $O(n^2 \log n)$, respectively. The expansion factor of ciphertext is $O(n^2)$. In addition, we need to add $O(n^4)$ bits in the public key to use optimization algorithm Opt .

4. Fully Homomorphic Encryption (FHE)

We now construct a new fully homomorphic shceme by using **SHE**. Since the multiplication operation increase the degree of perturbed error noise, we require to reduce it to obtain fully homomorphic encryption. We apply self-loop Gentry's bootstrappable technique by freshing a ciphertext c to get a new ciphertext c_{new} with the smaller error noise. To implement this function, we encrypt the secret key s generated by **KeyGen** and add the ciphertexts of s to the public key.

4.1 FHE Scheme

KeyGen Algorithm.

(1) First, generate pk and sk as **SHE**.

(2) Assume $s = \sum_{j=n^2+3}^{2n^2+6} s_j 2^{-j}$. Choose random integers $a_j \in (2^{O(n)}, 2^{n^2})$, $e_j \in Z$ with $|e_j| < 2^{n-1}$, $j \in [n^2+3]$, and compute $\left[\vec{s}_{j+n^2+3} = a_j p + 2e_j + s_{j+n^2+3}\right]_{b_0}$.

(3) Output the public key $pk = (n, r, \{b_i\}_{i=0}^{\tau}, \vec{s} = \sum_{j=0}^{n^2+3} \vec{s}_j 2^{-(j+n^2+3)})$, and the secret key

$sk = (p)$.

The **Enc**, **Dec**, **Add**, **Mul** algorithms are identical to ones in the above **SHE**.

Recrypting algorithm (Recrypt). Evaluate a new ciphertext

$$c_n = \lfloor c \times \vec{s} + 0.5 \rfloor \bmod 2 \oplus c \bmod 2 .$$

Theorem 4.1. Recrypt correctly generates a 'fresh' ciphertext c_{new} with the same message of c and the perturbed error noise e subject to $|2e| < (p/8)^{1/2}$.

Proof: We know the general form of ciphertext $c = ap + 2e + m$ subject to $|2e| \le p/8$. So,

$$\lfloor c \times s + 0.5 \rfloor \bmod 2 = \lfloor (ap + 2e + m) \times s + 0.5 \rfloor \bmod 2 = a \bmod 2 .$$

By using $c_0 = c \bmod 2 = (ap + 2e + m) \bmod 2 = a \bmod 2 + m$, we obtain the message $m = c_0 + a \bmod 2$. Thus, **Recrypt** only substitutes s with $\vec{s}$, which is the form of the ciphertexts of bits in s. It is not difficult to verify that **Recrypt** algorithm correctly computes a new ciphertext c_{new} of m in c by using the ciphertext arithmetic circuit and the fact $h(s) = O(\log n)$, and c_{new} has the error noise less than $|2e| < (p/8)^{1/2}$. Notice that **Recrypt** uses the methods of the hamming weights, the symmetric polynomials and the three-for-two, all of which are explained in [Gen09, vDGHV10].■

Now we only require to prove the scheme can compute the circuit depth of **Recrypt**.

Lemma 4.1. The **Dec** algorithm from the above scheme is correct, if the error noise of ciphertext is less than $p/8$ when decrypted.

Lemma 4.2. The above scheme is correct for arbitrary arithmetic circuit C with addition and multiplication gates, and circuit depth $d = \log_2 n$.

Proof. Assume $c_j = a_j p + 2e_j + m_j$, $j = 1, 2$ are the ciphertexts of arbitrary two bits of s generated by **KeyGen** in FHE. To correctly decrypt, the perturbed error noise of ciphertext output by arithmetic circuit can not be too large. The error noise in addition gate is linearly rising, whereas the error noise in multiplication gate is exponentially increasing. So, the multiplication operation dominates the depth of arithmetic circuit. Now, we estimate the bound of the perturbed error term in the ciphertext generated by one multiplication operation.

$$\begin{aligned} c &= c_1 \times c_2 \\ &= (a_1 p + 2e_1 + m_1) \times (a_2 p + 2e_2 + m_2) . \\ &= (a \times f + 2e + m_1 m_2) \end{aligned}$$

where $a = (a_1 p + 2e_1 + m_1) \times a_2 + 2a_1 e_2 + a_1 m_2$, $e = e_1 \times (2e_2 + m_2) + m_1 e_2$.

So, $|2e| = |2e_1 \times (2e_2 + m_2) + 2m_1 e_2| < 2^{2n}$.

Since the perturbed error noise in the ciphertexts c_1, c_2 are less than 2^n. So, the error term for one multiplication operation is less than $(2^n)^2$. Thus, To correctly decrypt, the depth d of arithmetic circuit must be satisfied inequality $(2^n)^{2^d} \le p/8$, namely,

$d = \log(\log(p/8)/n) = \log_2 n$. ■

4.2 Performance of FHE

The size of the public key $pk = (n, r, \{b_i\}_{i=0}^{\tau}, \vec{s} = \sum_{j=0}^{n^2+3} \vec{s}_j 2^{-(j+n^2+3)})$ is $O(n^4)$, the size of secret key $sk = (p)$ is $O(n^2)$. The expansion factor of ciphertext is $O(n^2)$.

5. Security analysis

To reduce the security of our scheme to the hardness of the approximate-GCD over the binary polynomials, we first define the approximate-GCD problem.

Definition 5.1. (Approximate-GCD over the Integers (AGCD)) Given a list of approximate multiples of p: $\{b_i = a_i p + e_i : a_i \in (2^{O(n)}, 2^{O(n^2)}), e_i \in Z, s.t. |e_i| < 2^{n-1}\}_{i=0}^{\tau}$, find p.

The following theorem is proved in [vDGHV10].

Theorem 5.1. Suppose there is an algorithm A which breaks the semantic security of our SHE with advantage ε. Then there is an algorithm D for solving **AGCD** with advantage at least $\varepsilon/2$. The running time of D is polynomial in the running time of A, and $1/\varepsilon$.

6. Conclusion

We have designed a new fully homomorphic encryption scheme over the integers. The security of our scheme relies on the hardness of solving approximate-GCD problem over the integers.

References

[Ajt96] M. Ajtai. Generating hard instances of lattice problems (extended abstract). In Proc. of STOC 1996, pages 99-108, 1996.

[ACG08] C. Aguilar Melchor, G. Castagnos, and G. Gaborit. Lattice-based homomorphic encryption of vector spaces. In IEEE International Symposium on Information Theory,

ISIT'2008, pages 1858-1862, 2008.
[BGN05] Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-DNF formulas on ciphertexts. Lecture Notes in Computer Science, 2005, Volume 3378, pages 325-341, 2005.
[vDGHV10] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. Fully homomorphic encryption over the integers. In Proc. of Eurocrypt, volume 6110 of LNCS, pages 24-43. Springer, 2010.
[Gen09] C. Gentry. Fully homomorphic encryption using ideal lattices. In Proc. of STOC, pages 169-178, 2009.
[GHV10] C. Gentry and S. Halevi and V. Vaikuntanathan. A Simple BGN-type Cryptosystem from LWE. In Proc. of Eurocrypt, volume 6110, pages 506-522, 2010.
[GPV08] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Proc. of STOC, pages 197-206, 2008.
[HG01] N. Howgrave-Graham. Approximate integer common divisors. In Cryptography and Lattices, International Conference (CaLC)'01, LNCS 2146, pages 51-66. Springer, 2001.
[LPR10] V. Lyubashevsky and C. Peikert and O. Regev. On Ideal Lattices and Learning with Errors over Rings. In Proc. of Eurocrypt, volume 6110, pages 1–23, 2010.
[Mic07] D. Micciancio Generalized compact knapsaks, cyclic lattices, and efficient one-way functions. Computational Complexity, 16(4):365-411.
[MR07] D. Micciancio and O. Regev. Worst-case to average-case reductions based on Gaussion measures. SIAM Journal Computing, 37(1):267-302, 2007.
[Reg09] O. Regev, On lattices, learning with errors, random linear codes, and cryptography, Journal of the ACM (JACM), v.56 n.6, pages1-40, 2009.
[SS10] D. Stehle and R. Steinfeld. Faster Fully Homomorphic Encryption. Cryptology ePrint Archive: Report 2010/299: http://eprint.iacr.org/2010/299.
[SV10] N. P. Smart and F. Vercauteren. Fully Homomorphic Encryption with Relatively Small Key and Ciphertext Sizes. Lecture Notes in Computer Science, 2010, Volume 6056/2010, 420-443.
[SYY99] T. Sander, A. Young, and M. Yung. Non-interactive CryptoComputing for NC1. In 40th Annual Symposium on Foundations of Computer Science, pages 554{567. IEEE, 1999.
[RAD78] R. Rivest, L. Adleman, and M. Dertouzos. On data banks and privacy homomorphisms. In Foundations of Secure Computation, pages 169-180, 1978.
[Yao82] A. C. Yao. Protocols for secure computations (extended abstract). In 23rd Annual Symposium on Foundations of Computer Science (FOCS '82), pages 160-164. IEEE, 1982.