Fully Homomorphic Encryption over the Binary Polynomials

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 binary polynomials. 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 $O(n)$ in our scheme.
The security of our scheme is based on the hardness of finding an approximate-GCD problem over the binary polynomials, which is given a list of binary polynomials perturbed by the error polynomials with the smaller degree.
Keywords: Fully Homomorphic Encryption, Approximate-GCD, Approximate Binary Polynomial 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 designing a simple fully homomorphic encryption scheme over the binary polynomial. Our scheme is based on the trapdoor function of an approximate binary polynomial GCD. For a binary polynomial $s \in F_2[x]$ with $s = 1 \bmod x$ and $\deg(s) = 2n+1$, the public key is a list of approximate multiples for s: $\{b_i = (a_i s + x e_i) \bmod 2\}_{i=1}^{\tau}, \tau = O(n)$, where a_i, e_i is the uniformly random elements over $F_2[x]$ such that $\deg(e_i) \le n-1$. The secret key is s. To encrypt a message bit m, the ciphertext is computed as $c = \sum_{i \in T, T \subseteq \{1,\ldots,\tau\}} b_i + xe + m$, where $\deg(e) \le n-1$. To decrypt, we compute the message bit $m = c \bmod s \bmod x$.

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

1.1 Our Contribution

In high level, our scheme is similar to the fully encryption scheme over the integers [vDGHV10]. Their scheme is to work over the integers, whereas our scheme is to work over the binary polynomials. The main difference between two schemes is efficiency and security.

The size of the public key in our scheme is $O(n^2)$ bits, the expansion factor of ciphertext is $O(n)$. The security of our scheme depends on the hardness of finding an approximate-GCD over the binary polynomials, given a list of approximate multiples of s.

1.2 Related work

Rivest, Adleman, and Dertouzos [RAD78] first investigated a privacy homomorphism, which now is called the fully homomorphic encryption (FHE). Many researchers [BGN05, ACG08, SYY99, Yao82] have worked at this open problem. Until 2009, Gentry [Gen09] constructed the first fully homomorphic encryption using ideal lattice. In Gentry's scheme, the public key is approximately n^7 bits, the computation per gate costs $O(n^6)$ operations. Smart and Vercauteren [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. Dijk, Gentry, Halevi, and Vaikuntanathan [vDGHV10] proposed a simple fully homomorphic encryption scheme over the integers, whose security depends on the hardness of finding an approximate integer gcd. Stehle and Steinfeld [SS10] improved Gentry's fully homomorphic scheme and leaded to a faster fully homomorphic scheme, with $O(n^{3.5})$ bits complexity per elementary binary addition/multiplication gate, but the assumption of security in [SS10] is stronger than that in [Gen09].

1.3 Outline

In the next section we recall the definition of the homomorphic encryption. In Section 3, we present a somewhat homomorphic encryption. In Section 4, we transform the somewhat homomorphic encryption into a fully homomorphic encryption by using self-loop bootstrappable technique. In Section 5, we reduce the security of our scheme to the approximate-GCD problem over the binary polynomials. In Section 6, we conclude this paper and give some open problems.

2. Preliminaries

Notations. To simplify the notation, all computation is over the field F_2 throughout this paper. Let n be a security parameter, $\tau = n-1$, $[n] = \{0,1,...,n\}$.

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 first give a somewhat homomorphic encryption, then prove the correctness of scheme, and finally simply analyze its performace.

3.1 Construction

Key Generating Algorithm (GenKey).

(1) Select a random irreducible binary polynomial $s \in F_2[x]$ with $\deg(s) = 2n+1$ and $s = 1 \bmod x$.

(2) Pick uniformly random elements $a_i, e_i \in F_2[x], i \in [\tau]$ subject to $\deg(a_i) \le 2n+1$, $\deg(a_i) = O(n)$, $\deg(a_0) = 2n+1$ and $\deg(e_i) \le n-1$. Then compute $b_0 = a_0 \times s + xe_0$ and $b_i = (a_i s + xe_i) \bmod b_0$.

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

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

$\deg(e) \le n-1$. Compute the ciphertext $c = (\sum_{i\in T} b_i + xe + m) \bmod b_0$.

Add Operation (Add). Given the public key pk, and the ciphertexts c_1, c_2, evaluate the ciphertext $c = c_1 + c_2$.

Multiplication Operation (Mul). Given the public key pk, and the ciphertexts c_1, c_2, evaluate the ciphertext $c = c_1 \times c_2$.

Decryption Algorithm (Dec). Given the secret key sk, and a ciphertext c, decipher $m = c \bmod s \bmod x$.

It is easy to see that the above scheme is correct for **Enc**, **Dec** algorithm.

Theorem 3.1. The above scheme can compute arbitrary additions and one multiplication.

Proof. Assume c_1, c_2 are two ciphertexts generated by **Enc**. The degree of perturbed error polynomial in the addition is not rising, whereas the degree of perturbed error polynomial in the multiplication is linearly increasing. So, we only prove that the scheme can correctly carry out one multiplication.

$$\begin{aligned} c &= c_1 \times c_2 \\ &= (\sum_{i\in T_1} b_i + xe_1 + m_1) \times (\sum_{i\in T_2} b_i + xe_2 + m_2). \\ &= a \times s + xe + m_1 m_2 \end{aligned}$$

where $a = (\sum_{i\in T_1} b_i + xe_1 + m_1) \times (\sum_{i\in T_2} a_i) + x\sum_{i\in T_2} e_i + xe_2 + m_2$,

$e = ((\sum_{i\in T_1} e_i + e_1) \times (\sum_{i\in T_2} xe_i + xe_2 + m_2) + m_1 \times (\sum_{i\in T_2} e_i + e_2)$.

So,

$$\begin{aligned} \deg(xe) &= \deg((\sum_{i\in T_1} e_i + e_1) \times (\sum_{i\in T_2} xe_i + xe_2 + m_2) + m_1 \times (\sum_{i\in T_2} xe_i + xe_2)) + 1 \\ &\le \deg(\sum_{i\in T_1} e_i + e_1) + \deg(\sum_{i\in T_2} xe_i + xe_2 + m_2)) + 1 \\ &\le 2n \end{aligned} .$$

Hence, $c \bmod s \bmod x = (as + xe + m_1 m_2) \bmod s \bmod x = m_1 m_2$. ■

Notice that **MUL** increases the length of ciphertext. We will solve this problem in Section 4.

3.2 Performance of SHE

The size of the public key $pk = (n, \{b_i\}_{i=0}^{\tau})$ is $O(n^2)$, the size of secret key $sk = (s)$ is $O(n)$. The running times of **Enc**, **Dec**, **Add**, **Mul** are $O(n^2)$, $O(n^2)$, $O(n)$, and

$O(n\log n)$, respectively. The expansion factor of ciphertext is $O(n)$.

4. Fully Homomorphic Encryption (FHE)

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

4.1 Scheme of FHE

KeyGen Algorithm.

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

(2) Assume $s=\sum_{j=0}^{2n+1} s_j x^j$. Compute the inverse polynomial of the secret key $s^{-1}=\sum_{j=0}^{2n+1} s_j^{-1} x^{-(j+2n+1)}$ s.t. $s\times s^{-1}=1+O(x^{-2n-2})$.

(3) Choose uniformly random polynomials $a_j \in F_2[x], j=[2n+1]$ with $\deg(a_j)\le 2n+1$, and perturbed error polynomials $e_j \in F_2[x]$ with $\deg(e_j)\le n-1$, and then compute $\vec{s}_j^{-1}=(a_j s+xe_j+s_j^{-1}) \bmod b_0$, $j=[2n+1]$.

(4) Choose uniformly random polynomials $a_j \in F_2[x], j=[4n-1]$ with $\deg(a_j)=j+2n+2$, and perturbed error polynomials $e_j \in F_2[x]$ with $\deg(e_j)\le n-1$, and then compute $t_j=a_j s+xe_j$.

(5) Output the public key $pk=(n,r,\{b_i\}_{i=0}^{\tau},\{t_j\}_{j=0}^{4n-1},\vec{s}^{-1}=\sum_{j=0}^{2n+1}\vec{s}_j^{-1}x^{-(j+2n+1)})$, and the secret key $sk=(s)$.

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

Optimization of MUL algorithm (MOpt). Assume $c=c1\times c2=\sum_{i=0}^{8n+2} c_i x^i$. For $i=8n+2$ to $4n+3$, $c=c \bmod t_{i-4n-3}$. Finally, output $c=c \bmod b_0$.

Recrypting algorithm (Recrypt). Evaluate a new ciphertext $c_{new} = \left\lceil (c \times \vec{s}^{-1}) \right\rfloor \bmod x + c_0$.

Recall that $\lceil b \rfloor$ is removed all negative exponential terms over the variable x for the polynomial b.

It is easy to verify that **MOpt** is correct and reduces the length of ciphertext to the fixed length.

Theorem. Recrypt correctly generates a 'fresh' ciphertext c_{new} with the same message of c and the perturbed error term e subject to $\max D(xe) \le n$.

Proof: We know the general form of ciphertext $c = as + xe + m$ subject to $\max D(xe) \le 2n$. So,

$$\begin{aligned} & \left\lceil c \times s^{-1} \right\rfloor \bmod x \\ &= \left\lceil (as + xe + m) \times s^{-1} \right\rfloor \bmod x \\ &= \left\lceil as \times s^{-1} + (xe + m) \times s^{-1} \right\rfloor \bmod x \\ &= \left\lceil a \times (1 + O(x^{-2n-2})) + O(x^{-1}) \right\rfloor \bmod x \\ &= a \bmod x \\ &= a_0 \end{aligned}.$$

By using $c_0 = (as + xe + m) \bmod x = a_0 + m$, we obtain the message $m = c_0 + a_0$. Thus, **Recrypt** only substitute s^{-1} with $\vec{s}^{-1}$, which is the form of the ciphertexts of bits in s^{-1}. Moreover, we merely use the addition of ciphertexts in **Recrypt**, which does not increase the degree of the perturbed error polynomial, and the degree of the perturbed error terms in ciphertext polynomial $\vec{s}^{-1}$ is at most n. ■

4.2 Performance of FHE

The size of the public key $pk = (n, r, \{b_i\}_{i=0}^{\tau}, \{t_j\}_{j=0}^{4n-1}, \vec{s}^{-1} = \sum_{j=0}^{2n+1} \vec{s}_j^{-1} x^{-(j+2n+1)})$ is $O(n^2)$, the size of secret key $sk = (s)$ is $O(n)$. The running times of **Enc**, **Dec**, **Add**, **Mul, MOpt, Recrypt** are also $O(n^2)$, $O(n^2)$, $O(n)$, and $O(n \log n)$, $O(n^2)$, $O(n^2)$, respectively. Thus, the running time of **Mul** is increased to $O(n^2) + O(n \log n) = O(n^2)$ to remain the fixed length of ciphertext. The expansion factor of ciphertext is $O(n)$.

5. Security

5.1 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 Binary Polynomials (BP-AGCD)) Given a list of polynomials $\{b_i = a_i s + e_i\}_{i=0}^{\tau}$ for a randomly selected polynomial $s \in F_2[x]$ with degree $2n+1$ and $s = 1 \bmod x$, where a_i, e_i with $\max D(a_i) = O(n)$ and $\max D(e_i) < n$ are the uniformly random elements over $F_2[x]$, find s.

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 **BP-AGCD** with advantage at least $\varepsilon/2$. The running time of D is polynomial in the running time of A, and $1/\varepsilon$.

Proof. The proof of Theorem 5.1 is similar to that in [vDGHV10] except for the difference between polynomial and integer. ■

Remark 1: In some sense, the security of our scheme is higher than theirs since the error noise in the ciphertext is bigger for our scheme.

Remark 2: Although the public key of **FHE** includes the ciphertexts of the secret key, it is not difficult to prove the semantic security of **FHE** by using Theorem 5.1.

5.2 Known Attack

After the first manuscript posted, Martin Albrecht presented an attack by using the public key. His attack works if a sufficient number of the public key is given. Thus, we change the scheme only to include $b_0, b_1, ..., b_{n-1}$ as the public key in **SHE**. Certainly, this scheme can not open the ciphertext and its corresponding plaintext to guarantee the security of scheme.

6. Conclusion and Open Problem

We constructed a new fully homomorphic encryption scheme based on the binary polynomials. The security of our scheme relies on the hardness of solving approximate-GCD problem over the binary polynomial. As far as I know, this problem does not appear in the related paper, but the corresponding problem over the integers is studied in [HG01, vDGHV10]. The interesting problem is whether the hardness of solving BP-AGCD is equivalent to that over the integers. Another problem is to extend the scheme to reduce the expansion of ciphertext from $O(n)$ to $O(1)$.

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.