International Journal on Future Revolution in Computer Science & Communication Engineering
Volume: 3 Issue: 12
ISSN: 2454-4248
131 – 134

Study of Fully Homomorphic Encryption over Integers

ZHOU Gang Qiang
College of Electronic and Computer
Zhejiang Wanli University
NingBo 315100, China
e-mail: 2665680080@qq.com

CHEN Shu Hang
College of Electronic and Computer
Zhejiang Wanli University
NingBo 315100, China
e-mail: 296326173@qq.com

MA Jia Min
College of Electronic and Computer
Zhejiang Wanli University
NingBo 315100, China
e-mail: 913937747@qq.com

Abstract—Fully homomorphic encryption has long been regarded as an open problem of cryptography. The method of constructing first fully homomorphic encryption scheme by Gentry is complicate so that it has been considered difficult to understand. This paper explains the idea of constructing fully homomorphic encryption and presents a general framework from various scheme of fully homomorphic encryption. Specially, this general framework can show some possible ways to construct fully homomorphic encryption. We then analyze the procedure how to obtaining fully homomorphic encryption over the integers. The analysis of recrypt procedure show the growth of noise, and the bound of noise in recrypt procedure is given. Finally, we describe the steps of implementation..

Keywords*-Fully homomorphic encryption; encryption; approximate-GCD problem*

*****

I. INTRODUCTION

Encryption has traditionally been viewed as a mechanism that enables secure communication [1]. For someone who have the secret decryption key can learn the entire message, but without the decryption key, the ciphertext is completely useless. Can we do arbitrary computations on data while it remains encrypted, without ever decrypting it? This question from Rivest et al. and then become a open problem in cryptography [2]. In a breakthrough work Gentry described in 2009 the first construction of a fully homomorphic scheme [3]. At 2010 van Dijk et al. described a fully homomorphic encryption scheme over the integers [4]. In recent years DGHV scheme was improved continuously by Coron et al. [5][6]. Asymptotics of FHE over integers are much better now.

II. THE IDEA OF FULLY HOMOMORPHIC ENCRYPTION

The first step in framework of Gentry’s scheme is to describe a somewhat homomorphic scheme that supports a limited number of additions and multiplications on ciphertexts rather than do arbitrary computation on ciphertexts [7]. This is because every ciphertext has a noise component and any homomorphic operation applied to ciphertexts increases the noise in the resulting ciphertext. Once this noise reaches a certain bound the resulting ciphertext does not decrypt correctly anymore. For example, the bit-size of noise is ρ and the threshold of noise size is k • ρ , then the bound is reached after only log2k levels of multiplication.

If one wants to achieve fully homomorphic encryption, the problem of noise growth must be solved. The Gentry’s idea of solving it is to decrypt ciphertext, but homomorphically! The decryption circuit is inputted in the bits of a secret key and the bits of a ciphertext, each encrypted under another public key. As long as the decryption circuit can be handle by this somewhat homomorphic scheme, the output is another ciphertext for the same plaintext. If the degree of the decryption polynomial is small enough, the new ciphertext's noise would less than the old cipertext's noise; this is called the "ciphertext refresh" procedure. One can refresh ciphertexts before every homomorphic operation. So, the decryption circuit is augmented by some gate- e.g., Add; call this augmented circuit. Since AND gate and XOR gate have the property of functional completeness [8]. That is, any other logic function can be implemented by combining AND gate and XOR gate. If the somewhat homomorphic scheme can handle augmented circuits, it means it do arbitrary computation on ciphertexts. This somewhat homomorphic scheme can obviously handle AND gate and XOR gate. So, it is critical whether the decryption circuit can be handled. If it can be, this calls “bootstrapping”. Unfortunately, it is not. Hence one needs to squash the decryption circuit so that the degree of the decryption polynomial is small enough. Then the somewhat homomorphic scheme can do arbitrary computation on ciphertexts. It means we get fully homomorphic encryption scheme. This framework has been instantiated with a number of cryptographic assumptions, yielding progressively simpler and more efficient schemes [9][10][11][12][13][14].

III. THE DGHV SCHEME

At 2010 van Dijk et al. described the first fully homomorphic encryption scheme over the integers. The DGHV scheme was got from a symmetric encryption

 ISSN: 2454-4248
131 – 134

scheme. Encrypt: c←m+2r+pq ; Decrypt: m←(c mod p) mod 2= (c–p·*⌈c/p⌋) mod 2 = Lsb(c) XOR Lsb(⌈c/p⌋). The key is a random odd integer p. Choosing at random large q and small r. A bit m∈ {0,1}. The plaintext space is a set of binary, and the Ciphertext space is a set of integers. The system as described above is obviously a somewhat homomorphic encryption scheme. Ciphertexts from above scheme are near-multiples of p. We call (c mod p) the noise associated to the ciphertext c. It is the distance to the nearest multiple of p. In order to turn it into a public key system, we take advantage of the homomorphic nature of the system. That is zero and one are also valid encryptions of themselves.

The public key is pk={ x_i ; $x_i=r_i+pq_i$ }. Encryption of a plaintext m is then performed by selecting a random subset S from pk, a random noise r, and outputting the ciphertext c according to: $c\leftarrow m+2r+2\sum_{i\in s}x_i \bmod x_0$. Decryption is as same as above. This is a somewhat homomorphic encryption scheme in DGHV scheme. The security of the scheme reduces to the hardness of the approximate-GCD problem. The bound of noise is set to be p/8 in DGHV scheme. In fact, p/4 or p/16 is also correct, but p/2 is not. Because computing ⌈c/p⌋ is complicated, E cannot handle the function f(p, c)= ⌈c/p⌋ (Does it surprise you?). Hence it makes the decryption circuit become shallow by squashing the decryption circuit.

The main idea of the transformation is to replace c/p, which multiplies two long numbers, with the summation of a fairly small set of numbers. In terms of the bits of the addends, this summation corresponds to a polynomial of fairly low degree that the somewhat homomorphic scheme can handle. Since c is ciphertext and shouldn't be changed, we only transfer p to the summation of a fairly small set of numbers. However, p is secret key and cannot be open; p can only be hide in the summation. Hence we need a trapdoor to ensure the security of the hidden p. This trapdoor is the sparse subset sum problem (SSSP). Generate a set $y = \langle y_1,\ldots\ldots,y_t\rangle$ of rational numbers in [0,2) such that there is a sparse subset S {1,……, t} of size α with $\sum s_i\cdot y_i \approx 1/p \bmod 2$ ($i\in S$). Set sk* to be the sparse subset S, encoded as a vector s $\{0,1\}^t$ with Hamming weight α. Set pk*←(pk, y). Since $\sum s_i\cdot c\cdot y_i \approx \sum s_i\cdot z_i \approx c/p \bmod 2$, we transfer ⌈c/p⌋ to $\lceil \sum s_i\cdot z_i \rfloor$. Now the decryption circuit is converted to Lsb(c) XOR Lsb($\lceil \sum s_i\cdot z_i \rfloor$). The ciphertext c* consists of c and $\mathbf{z} = \langle z_1,\ldots\ldots, z_t \rangle$. To see that the somewhat homomorphic scheme can handle the decryption function plus an additional gate when α is set small enough, let us consider the computation of the sum $\sum s_i\cdot z_i$. In this sum, we have numbers $\{a_i \mid 1<=i<=\alpha\}$, each ai expressed in binary$\{a_{i,j} \mid 1<=i<=\alpha,\ 0<=j<=-l)$ with l=(log α), where at most α of the ai's are nonzero. We want to express each bit of the output as a polynomial of the input bits, while minimizing the degree of the polynomial and the number of monomials. It can be achieved by some calculating skills. Finally, the somewhat homomorphic become bootstrappable. Then we get a fully homomorphic encryption scheme over integers.

The public key size is $\tilde{O}(\lambda^{10})$. In practice the size of the x_i's should be at least 2^{23} bits to prevent lattice attacks. The public key size is then at least 2^{46} bits, which is too large for any practical system.

The somewhat-homomorphic scheme relies on hardness of approximate-GCD. Resulting scheme relies also on hardness of sparse-subset-sum and circular security. There are two types attacks. The first consider attacks on the approximate-gcd problem for two numbers including brute-forcing the remainders, continued fractions, and Howgrave-Graham's approximate gcd algorithm[15]. The second consider attacks for arbitrarily large values of t including lattice-based algorithms for simultaneous Diophantine approximation[16], Nguyen and Stern's orthogonal lattice[17], and extensions of Coppersmith's method to multivariate polynomials[18]. Since the sparse subset sum problem is not well study[19], it need to study deeply.

IV. VARIANT OF THE DGHV SCHEME—CMNT SCHEME

At 2011 Coron et al. proposed some techniques in [5] to reduce the public key size and increase the efficiency of the DGHV scheme, the most important of which is to use a quadratic form instead of a linear form for masking the message when computing a ciphertext. The idea consists in storing only a smaller subset of the public key and then generating the full public key on the fly by combining the elements in the small subset multiplicatively. More precisely, ciphertexts are computed as: $c\leftarrow m+2r+\Sigma\,(b_{i,j}\cdot x_{i,0}\cdot x_{j,0}) \bmod x_0$ for $1\le i, j \le \beta$,where β is a new parameter. Then only 2β integers need to be stored in the public key in order to generate the β^2 integers used for encryption. Evaluate and decrypt is same as in the original scheme, except that ciphertexts are reduced modulo x_0 after addition and multiplication.

The idea of Squashing decryption circuit is same as DGHV scheme in order to get a fully homomorphic encryption scheme. However public key is be compressed in CMNT scheme by some techniques. Generate the y_i's using a pseudo-random generator f(se) instead of storing the y_i's in the public key as in [4]. Then only the seed se and y_1 need to be stored in the public key, and the other y_i's can be recovered during ciphertext expansion by applying f (se) again. In addition, encryptions of the secret key bits should also be made available publicly so that the Recrypt procedure is simply obtained by applying the decryption circuit to the ciphertext bits and the encrypted secret key

 ISSN: 2454-4248
131 – 134

bits. Therefore one uses two bit vectors **s**(0) and **s**(1) of length $\Theta^{1/2}$ Instead of generating the secret key as a single bit vector s=(s_1, s_2,……, s_Θ). Then s is recovered on the fly during decryption as the following matrix with Θ entries: $s_{i,j} = s_i(0) \cdot s_j(1)$.The encryption of the bits of s(0) and s(1) is $\sigma(0)$ and $\sigma(1)$. Finally, the secret key as sk = (s(0) ,s(1)), and the public key as pk = (pk*, se, $u_{1,1}$, $\sigma(0)$, $\sigma(1)$) . This brings down public key size to about $\tilde{O}(\lambda^7)$.

CMNT scheme is still semantically secure, but under the (stronger) error-free approximate GCD assumption. The know attacks include brute force attack on the noise, approximate-GCD attack on the public key and lattice attack on the sparse subset-sum problem. Corresponding 72 bits of security, encryption and recryption take 3 minutes and 14 minutes respectively, with a public key size of 800 MBytes.

V. VARIANT OF THE DGHV SCHEME—CNT SCHEME

At 2012 Coron et al. describe three technologies to improve DGHV schme.

The first is a technique to reduce the public key size of DGHV-like schemes [5] by several orders of magnitude. The technology is first to generate the secret-key p. Then, use a pseudo-random number generator f with public random seed se to generate a set of γ bit integers χ_i (i.e. the -χ_i's are of the same bit-size as the x_i's). Finally, compute small corrections δ_i to the χ_i's such that $x_i = \chi_i - \delta_i$ is small modulo p, and store only the small corrections δ_i in the public key, instead of the full x_i's. Knowing the PRNG seed se and the δ_i's is sufficient to recover the x_i's. Therefore instead of storing a set of large x_i's in the public key we only store the much smaller δ_i's. The new public key for the somewhat homomorphic scheme has size about $\tilde{O}(\lambda^5)$. Corresponding 72 bits of security, encryption and recryption take 7 minutes and 12 minutes respectively, with a public key size of 10.1MBytes. This encryption scheme is still semantically secure under the error-free approximate GCD assumption, albeit in the random oracle model. The know attacks is same as CMNT scheme.

The second is Coron et al. prove that the natural extension of this quadratic encryption technique to to cubic forms, and more generally forms of arbitrary fixed degree d, remains secure, making it possible to further reduce the public key size. However, this quadratic encryption technique doesn't use in the implementation. This is because the "squashed decryption" procedure adds an incompressible additional term u1 of size $\gamma = \tilde{O}(\lambda^5)$ to the public-key, so it is unnecessary to reduce the number of public-key elements x_i or the number of encrypted secret-key bits.

The third is Coron et al. show how to adapt Brakerski, Gentry and Vaikuntanathan's (BGV) new FHE framework [20] to the DGHV scheme over the integers. The new BGV framework is described in [20] with Brakerski and Vaikuntanathan's scheme [21], and the key technical tool is the modulus-switching technique of [21] that transforms a ciphertext c modulo p into a ciphertext c' modulo p' simply by scaling by p'/p and rounding appropriately. This allows to reduce the ciphertext noise by a factor close to p'/p without knowing the secret-key and without bootstrapping. Under the BGV framework the noise ceiling increases only linearly with multiplicative depth, instead of exponentially. However the modulus switching technique cannot directly apply to DGHV since in DGHV the moduli p and p' are secret. It need to use some skill to achieve. First given as input a DGHV ciphertext c, and a "virtual" ciphertext of the form $c' = 2k \cdot q' + r'$ with $[q'] = [q]_2$ can be obtained . This is done by first "expanding" the initial ciphertext c using the yi's, as in the "squashed decryption" procedure in [4], and then "collapsing" the expanded ciphertext into c', using the secret-key vector **s** = (s_i). However we cannot reveal s in clear, so instead we provide a DGHV encryption under p' of the secret-key bits si, as in the bootstrapped procedure. Then the expanded ciphertext can be collapsed into a new ciphertext $c'' = 2 < \sigma, c' > + [c]_2$ under p' instead of p, for the same underlying plaintext. As in [20] a leveled fully homomorphic scheme can be constructed without bootstrapping. The security of this scheme is semantically secure under the decisional approximate GCD. Corresponding 72 bits of security, encryption and recryption take 3 seconds and 2 hours 27 minutes respectively, with a public key size of 18MBytes. The running time of the Recrypt operation is disappointing compared to the non-leveled implementation; however we think that there is room for improvement. A leveled fully homomorphic scheme include the bootstrapping operation; although not strictly necessary, this enables to get a FHE that can perform homomorphic evaluations indefinitely without needing to specify at setup time a bound on the multiplicative level.

VI. CONCLUDE

We describe the progress of fully homomorphic encryption over integers. The result show fully homomorphic encryption over integers is not only conceptual simplicity but also asymptotics are much better now. We think the high efficient method is similar to the modulus-switching technique should be to study in fully homomorphic scheme over integers. Moreover, we think, the running time of the Recrypt operation in the leveled implementation, there is room for improvement.

ACKNOWLEDGMENT

This research was supported by the National Innovation and Entrepreneurship Training Program of Zhejiang Wanli University(201610876008, 201710876016) and Zhejiang Province College Students Science and Technology

 ISSN: 2454-4248
131 – 134

Innovation Program of Zhejiang Wanli University(2017R420014).

REFERENCES

[1] W. Diffie and M. Hellman, "New directions in cryptography" IEEE Transactions on Information Theory, vol.IT-22, pp. 644-654, 1976.

[2] R. L. Rivest, A. Shamir, and L. M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems (reprint)," Commun. ACM, vol. 26, no. 1, pp. 96-99, 1983.

[3] C. Gentry.Fully homomorphic encryption using ideal lattices. STOC, 2009, 169-178.

[4] M. Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, Fully homomorphic encryption over the integers, EUROCRYPT, 2010, 24-43.

[5] J.S.Coron, A.Mandal,D.Naccache and M.Tibouchi, Fully Homomorphic Encryption over the Integers with Shorter Public Keys. CRYPTO, 2011, 6841:487-504.

[6] J.S. Coron, D. Naccache and M. Tibouchi. Public key compression and modulus switching for fully homomorphic encryption over the integers. EUROCRYPT, 2012, 7237: 446 -464.

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

[8] Enderton, Herbert. A mathematical introduction to logic (2nd ed.), Boston, Academic Press, 2001.

[9] N.P. Smart, F. Vercauteren, Fully homomorphic encryption with relatively small key and ciphertext sizes.PKC 2010, 6056:420-443.

[10] D. Stehle, R. Steinfeld, Faster fully homomorphic encryption. ASIACRYPT 2010, 2010, 6477:377-394.

[11] C. Gentry, S. Halevi, Implementing Gentry's fully homomorphic encryption scheme. EUROCRYPT 2011, 6632:129-148.

[12] Z. Brakerski, V. Vaikuntanathan. Fully Homomorphic Encryption from Ring-LWE and security for Key Dependent Messages, CRYPTO 2011, 6841:505-524.

[13] C.Gentry, S.Halevi, N.P.Smart. Better Bootstrapping in Fully Homomorphic Encryption, Cryptology ePrint Archive, 2011, Report 2011/680.

[14] C. Gentry, S. Halevi. Fully Homomorphic Encryption without Squashing using depth-3 arithmetic circuits, FOCS 2011,107-109.

[15] N. Howgrave-Graham. Approximate integer common divisors. CaLC' 01, 2001, 2146:51-66.

[16] J. C. Lagarias. The computational complexity of simultaneous diophantine approximation problems. SIAM J. Comput., 1985,14(1):196-209.

[17] P. Q. Nguyen, J. Stern. The two faces of lattices in cryptology. Cryptography and Lattices, CaLC' 01, 2001, 2146:146-180.

[18] D. Coppersmith. Small solutions to polynomial equations, and low exponent RSA vulnerabilities. J. Cryptology, 1997, 10(4):233-260.

[19] V. Vaikuntanathan. Computing blindfolded: New developments in Fully Homomorphic Encryption, FOCS 2011,5-16.

[20] Z. Brakerski, C. Gentry and V. Vaikuntanathan, Fully Homomorphic Encryption without Bootstrapping. Cryptology ePrint Archive, Report 2011/277.

[21] Z. Brakerski and V. Vaikuntanathan, Efficient Fully Homomorphic Encryption from (Standard) LWE. FOCS 2011, 97-106.