Journal of Computer and Communications, **2014, 2, 109-111**
Published Online January 2014 (http://www.scirp.org/journal/jcc)
http://dx.doi.org/10.4236/jcc.2014.22019

Simulation Study Based on Somewhat Homomorphic Encryption

Jing Yang[1], Mingyu Fan[1], Guangwei Wang[1], Zhiyin Kong[2]

[1]School of Computer Science and Engineering, University Electronic Science and Technology of China, Chengdu, China; [2]Science and Technology on Information Assurance Laboratory, Beijing, China.
Email: ay4922@163.com

Received October 2013

ABSTRACT

At present study, homomorphic encryption scheme is most focusing on algorithm efficiency and security and the rare for homomorphic encryption simulation research. This paper for the Gentry's Somewhat Homomorphic Encryption scheme for the simulation research, in clear text size within a certain range simulation was Somewhat Homomorphic Encryption scheme, and presents the relationship between the length of plaintext and ciphertext size.

KEYWORDS

Somewhat Homomorphic Encryption; Simulation; Clear Text Length; Ciphertext Length

1. Introduction

Rivest *et al*. [1], in 1978, put forward the concept of a fully homomorphic encryption, namely under the condition of not unlocking to various operations of cryptograph, the results from the decrypted plaintext and operation results of the same accordingly. All the ideas of the homomorphic encryption put forward, scholars both from home and abroad do a lot of research on fully homomorphic encryption; however, their proposed solutions are only for limited time homomorphism cryptograph computing, which cannot do homomorphism calculation of arbitrary depth processing circuit or any more as to time, it did not achieve the entire real homomorphism.

Until graduating from Stanford University in 2009, IBM researcher Gentry [2] based on ideal case the first fully homomorphic encryption scheme is proposed, and in his doctoral thesis [3], fully homomorphic encryption scheme is further discussed. Gentry's fully homomorphic encryption scheme is then proposed on the research and development of the homomorphic encryption provides a powerful support and motivation. Later, domestic and foreign scholars put forward many improved fully homomorphic encryption schemes. Dr Dijk *et al*. [4,5] proposed integer on the homomorphism scheme based on modular arithmetic, but the execution efficiency is low; Smart *et al*. [6] such as using Gentry fully homomorphic encryption idea, put forward the key and the cipher text with relatively small size of the solution, to improve the efficiency; Stehle and others optimized Gentry's scheme, put forward a fast fully homomorphic encryption scheme to reduce the computational complexity allowing the negligible probability decryption error on the weakened condition.

On the Gentry's paper for integer Somewhat Homomorphic Encryption scheme for the simulation research, under the condition of the weakening of certain parameters for the Gentry of the homomorphic encryption scheme, the simulation experiment on the premise of meeting the homogeneity of cipher text is obtained under different plaintext length sizes, and the linear relationship between them is discussed.

2. To The Knowlodge

2.1. Homomorphic Encryption

A homomorphic encryption scheme is composed of the fowling four algorithms:

Keygen: According to the security para-meter, we can produce the public key pk and private key sk of the scheme.

Encryption: We give a plaintext m and $m \in \{0,1\}^*$, so that we can get ciphertext c using public key pk to encrypt plaintext.

Decryption: Inputting private key sk and ciphertext c then decrypting them, we can get the output plaintext

m .

Evaluate: Inputting public key pk, t input circuits C and a set of ciphertext $c = (c_{1,} c_{2,} ... c_t)$,we can get the output result $c^* = Evaluate(pk, C, \bar{c})$ .What's the most important, the result must meet the condition $Dec(sk, c^*) = C(m_{1,} m_{2,} ... m_t)$.

2.2. Somewhat Homomorphic Encryption

Gentry structure of Somewhat homorphic encryption scheme is composed of the following four algorithms:

Parameter selection: Let's set them blow

$$\rho = \lambda\ ,\ \rho' = 2\lambda\ ,\ \eta = \tilde{O}(\lambda^2)\ ,\ \eta = \tilde{O}(\lambda^2)\ ,$$

$\gamma = \tilde{O}(\lambda^5)$. λ is safety parameter. Security parameters associated with the security of scheme, usually take dozens to hundreds of bits.

keygen: We choose η bit odd prime numbers p and θ bits odd prime numbers q randomly and order $N = pq$. Then choose two random integers $l \in [0, 2^{\gamma} / p]$, $h \in (-2^{\rho}, 2^{\rho})$, and calculate (1). Set public key $pk = (N, x)$, private key $sk = p$.

$$x = pl + 2h \qquad (1)$$

Encryption(pk, m): Given a plaintext $m \in \{0,1\}^*$, we choose two random integers $r_1 \in (-2^{\rho'}, 2^{\rho'})$ and $r_2 \in (-2^{\rho}, 2^{\rho})$. According to the public key $pk = (N, x)$, we can calculate (2), c serve as the result of ciphertext.

$$c = m + 2r_1 + r_2 x \bmod N \qquad (2)$$

Decryption(sk,c) : According to the given ciphertext, make use of private key sk to calculate (3).

$$m' = (c \bmod p) \bmod 2 \qquad (3)$$

$Evaluate(pk, C, c_1, c_2 ... c_t)$: Given a boolean circuit C with t inputs and t ciphertexts c_i . Let's put T ciphertext into extended circuits to perform all its operations, then verify the result of the circuit's output is (4) to see if it conform (5).

$$c^* = Evaluate(pk, C, c) \qquad (4)$$

$$Dec(sk, c^*) = C(m_{1,} m_{2,} ... m_t) \qquad (5)$$

3. Somewhat Homomorphic Encryption Simulation Study

3.1. Simulation Environment

In the experimental environment, we set safe parameter λ to the length of the 19 bits, the order of magnitude as the 10^5; so orders of magnitude for $\rho = \lambda$ also as the 10^5; order of magnitude for $\rho' = 2\lambda$ as the 10^5; order of magnitude for $\eta = O(\lambda^2)$ as the 10^{10}; order of magnitude for $\theta = O(\lambda^4)$ as the 10^{20}; order of magnitude for $\gamma = \tilde{O}(\lambda^5)$ as the 10^{25}. Algorithm *keygen* of the data are determined by the above parameters. Experiments on a Unix environment using the Compiler GCC (the GNU Compiler Collection).

3.2. Simulation Results

In order to facilitate analysis of data, size and ciphertext expressly to bit size, the simulation time in seconds. Because the simulation environment and algorithm limits, the size of the plaintext is only 10 bit, here we are in clear text size respectively for 1 bit, 2 bit, 5 bit and 10 bit to experiment. Although the cipher text size is very big, but in order to guarantee the consistency of the order of magnitude of the proceeds of the cipher text results by bit for the unit.

After the simulation，we get clear the relationship between the size and cipher text size shown in the **Figure 1**.

We can find the result that, along with the rising of the clear size, the size of the cipher text also increased, but the growth rate is not high. Visible proclaimed in writing to the size of the changes for the size of the cipher text is having a certain influence.

Than we consider the efficiency problem, we compare the size of plaintexts with time that the experiment costs, we can get their relation as **Figure 2**.

Through the **Figure 2** we can find that, along with the change of plaintext size, encrypt the consumed time is changing, and increases the time that time costs are increasing with the size of plaintext has also been gradually increased.

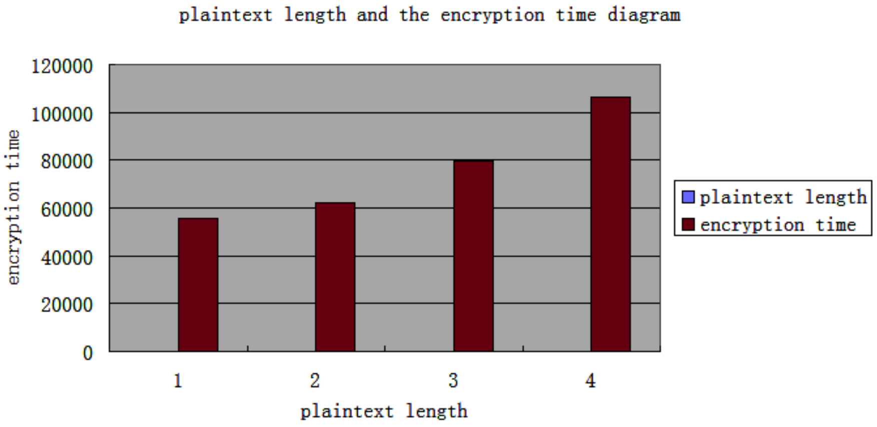
4. Epilogue

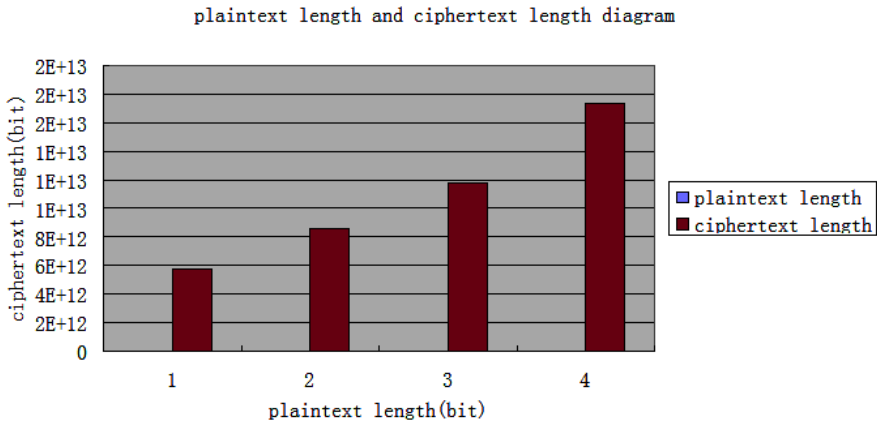
An Under Unix system for Somewhat homomorphic encryption scheme, simulation experiment and the simulation results to get the relationship between the size of the size and the ciphertext expressly as well as the time needed for different plaintext size case simulation analysis, we get the following conclusions:

1) In the security parameters are the same, to the cases of different input plaintext, along with the rising of the clear size, cipher text size is also on the increase, but growth is not high. In the security parameters are the same, for different input plaintexts, cost will gradually increase the time needed for simulation.

2) By the simulation results it can be seen that now the homomorphic encryption scheme can produce huge amounts of ciphertext, cost a lot of time at the same time. After study of homomorphic encryption we may need to start from the efficiency in order to improve the above problems.

3) As for the more research on our study, we want to reduce the cost of time and increase the length of the





keys to extend our research to more and more applications.

REFERENCES

[1] R. Rivest, A. Shamir and L. Adleman, "A Method of Obtaining Digital Signatures and Public-Key Cryptosystems," *Communications of the ACM*, Vol. 21, No. 2, 1978, pp. 120-126. http://dx.doi.org/10.1145/359340.359342

[2] C. Gentry, "Fully Homomorphic Encryption Using Ideal Lattices," *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, ACM Press, New York, 2009, pp. 169-178.

[3] D. Boneh and C. Gentry, "A Fully Homomorphic Encryption Scheme," Stanford University, Stanford, 2009.

[4] M. Van Dijk, C. Gentry and I. Halev, "Fully Homomorphic Encryption over the Integers," *Proceedings of the* 29*th Annual International Conference on Theory and Applications of Cryptograhic Techniques*. Springer-Verlag, Berlin, 2010, pp. 24-43.

[5] C. Gentry, "Computing Arbitrary Function of Encrypted Data," *Communications of the ACM*, Vol. 53, No. 3, 2010, pp. 97-105. http://dx.doi.org/10.1145/1666420.1666444

[6] N. P. Smart and F. Vercauteren, "Fully Homomorphic Encryption with Relatively Small Key and Ciphertext Sizes," *Proceedings of the* 13*th International Conference on Practices and Theory in Public Key Cryptography*, Springer-Verlag, Berlin, 2010, pp. 420-443.

[7] D. Stehle and R. Steinfeld, "Faster Fully Homomorphic Encryption," *Proceedings of International Conference on the Theory and Application of Cryptology and Information Security*, Springer, Berlin, 2010, pp. 377-394.