ISSN:2320-0790

An International Journal of Advanced Computer Technology

SECURING DATA USING FULLY HOMOMORPHIC ENCRYPTION SCHEMES OVER CLOUD COMPUTING

V. Satish Kumar[1], V.Trilik Kumar[2]

[1]Student, Computer Science & Engineering Dept., Dr. KVSR Institute of Technology, Andhra Pradesh, INDIA
[2]Assoc. Professor, Information Technology Dept., Dr. KVSR Institute of Technology, Andhra Pradesh, INDIA

Abstract: There is a problem for business organizations to move towards cloud computing regarding the safety and security issues associated with cloud computing. Different technologies have been used to relate these types of issues including various control methods and cryptographic techniques. When the data transmitted to the Cloud we use standard encryption methods to secure the operations and the storage of the data. Fully homomorphic encryption has cloud computing is to perform computations on encrypted data without previous decryption. The first fully homomorphic schemes have been proposed and developed to improve the performance, reduce the complexity and the cost of the scheme. Mainly two important schemes are refreshed and discussed in this paper. The first scheme discoursed in this paper is “Encryption over Integers using fully Homomorphism”. The second one is about “Encryption without Bootstrapping by using fully Homomorphism”. These two are basically concentrated on the security, performance and complexity factors of the mentioned schemes.

Key Words: Fully Homomorphic Encryption, homomorphic scheme, Security, Partial Homomorphic Encryption

1. INTRODUCTION

The distributed systems and especially cloud computing are developing according to the technology. The organization has benefits through information sharing and the greater degree of flexibility in scaling resources has pushed the cloud into mainstream computing.

However, the cloud comes into most information security problems from traditional computing domains. With this, the distributed nature of the cloud alters many new types of problems. There are several major problems that the cloud faces:

- **The cloud may not be trusted.** The cloud service provider (CSP) is not necessarily trusted. For example, a malicious company employee may be able to setup back doors and circumferential all the protection over the company cloud services. In addition, some systems in the cloud may be mismanaged, making them tender to attacks. Even further, some machines may belong to intruders.
- **Implementation of bugs can be overworked.** Even if the Cloud Service Providers are trusted and they provide easy mechanisms such as sandboxing and virtualization. Bugs in the machine are declared every day, may be exploited to circumvent any protection, e.g. [1]. As an example, in [2] the authors show that an attacker could take control of the VMware and Xen virtualization software when moving a virtual machine from one physical computer to another.
- **Side channel attacks can bypass security.** Although the system is fully protected and the code is executing in a confidential platform, the side channel attacks may still compromise the security. For example, an attacker using the cold boot attack [3] is able to retrieve sensitive data from the unrepressed DRAM after using a cold reboot to restart the machine. An intruder using the Branch prediction attacks [4] can gather information about the encryption keys by simply monitoring the system time. These types of attacks typically require physical access to the systems, which is not an easy way to perform. However it is possible that your code will be executed in a system belongs to the intruder in cloud computing settings. In such cases, the intruder will be able to gain active access to the system easily.

2. BACK GROUND

2.1 Homomorphic Encryption Schemes:

The development of homomorphic encryption provides yet another distinct approach to build SFE protocols. Conversationally, a homomorphic encryption scheme allows computation directly on encrypted data. It is easy to build the SFE protocol freely by using Homomorphic Encryption. Alice send the input x in the form of cipher texts to Bob. Bob will calculate f(x) directly on the cipher text which the Alice was sent and get back the encrypted result. And that result was decrypt only by Alice. As long as the security of the homomorphic encryption scheme holds, Bob will not be able to learn anything about x. The Homomorphic properties of public key encryption schemes, e.g. RSA and ElGamal encryption, were accepted. However they were largely viewed as a weakness rather than a plus. Applications where data is inactive typically require uninfluenced encryption.

However, the community of people has grown to believe the confidentiality of these schemes and recently, the work of Gentry and others exhibit that, when carefully employed, such homomorphic properties can be quite useful. Indeed, a number of recent specific applications such as data aggregation in distributed networks, electronic voting, and biometrics and Financial transactions.

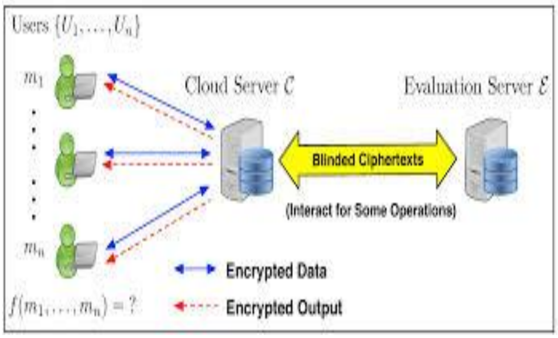
Users $\{U_1,\ldots,U_n\}$
m_1
m_n
Cloud Server $\mathcal{C}$
Evaluation Server $\mathcal{E}$
Blinded Ciphertexts
(Interact for Some Operations)
Encrypted Data
Encrypted Output
$f(m_1,\ldots,m_n) = ?$

Fig - 1 : Homomorphic encryption scheme

Restriction of Partial HE Scheme: Without any doubt, partial HE schemes are valuable in certain applications. In accession, for particular applications some partial HE schemes are very adequate regarding the efficiency. E.g. the Paillier scheme can execute ratings in milliseconds level. Yet, the loopholes of this category of schemes are also clear.

The main problem of the partial HE schemes is the range of the circuits that they provide. Almost partial HE schemes only support one kind of process, e.g. additions for Paillier and multiplications for RSA. This draws a heavy restriction on the circuits that the HE schemes can evaluate homomorphically. Some partial HE schemes supports more than one process, however, limitations even survive. The Boneh-Goh-Nissim scheme supports one level of multiplications via bilinear maps. The characteristic of this scheme make changes by measuring 2-DNF expressions and which may not be measured by using single-operation partial HE schemes. However, only one level of multiplication operation is supported. The Boneh-Goh-Nissim scheme cannot hold additional difficult circuits.

Somewhat Homomorphic Encryption Scheme: FHE schemes without freeing the disturbance can also be utilized as partial HE schemes. These schemes commonly have a large number of additions and bounded levels of multiplications. With this property HE schemes are usually referred as Somewhat Homomorphic Encryption Schemes (SWHE). Although this type of partial HE schemes can support much more complicated circuits than the single-operation ones, it is even hardly limited because the restrictions on levels of multiplications will finally be achieved. Only some partial HE schemes have additional constrictions that keep them from existing applied only for exceptional applications. For example, the Boneh-Goh-Nissim scheme needs a small message length to reach responsive decryption ratio, which enforces additional restriction to the scheme.

2.2 GENTRY'S Fully Homomorphic Encryption Schemes

The idea of fully homomorphic encryption was raised by Rivest, Adleman and Dertouzos [7], shortly after the invention of RSA [8]. A fully homomorphic encryption scheme consists of following four algorithms:

- KeyGen (λ) **-** Generates the encryption keys. As an input, it exacts λ for the security parameter and generates the secret key sk and the public key pk.
- Enc (pk, m) – Encrypts the plaintext m with the public key pk to create ciphertext c.
- Dec (sk,c) – Decrypts the ciphertext c using the secret key sk to retrieve the plaintext m.
- Eval (pk,C,c1,c2 … ct) **-** Uses a Boolean circuit C to outputs a ciphertext of f(m) such that Decrypt (sk, m) = f(m).

Gentry's construction consists of three main elements: a somewhat homomorphic encryption scheme that can evaluate low degree polynomials, a technique to "squash the decryption circuit" to get a "bootstrappable" scheme and finally a method of transferring "bootstrapping" the scheme into fully homomorphic encryption scheme. The significant point in this process is to obtain a scheme that can evaluate high degree polynomials while the decryption procedure can still be expressed as low degree polynomial. Once the scheme can evaluate its own decryption function plus an additional operation then it is called "bootstrappable" scheme and can be converted into a fully homomorphic scheme.

Although Gentry scheme proved the possibility of implementing fully homomorphic encryption, the scheme complexity, efficiency and performance needed to be improved. For example Gentry has estimated that building a circuit to execute an encrypted Google search with encrypted keywords would multiply the current computing time by one trillion. However the scheme has cheered many researchers to suggest many forms to Gentry's scheme to better the performance and lighten the complexity and ciphertext length. Two interesting approaches were discussed in the following sections.

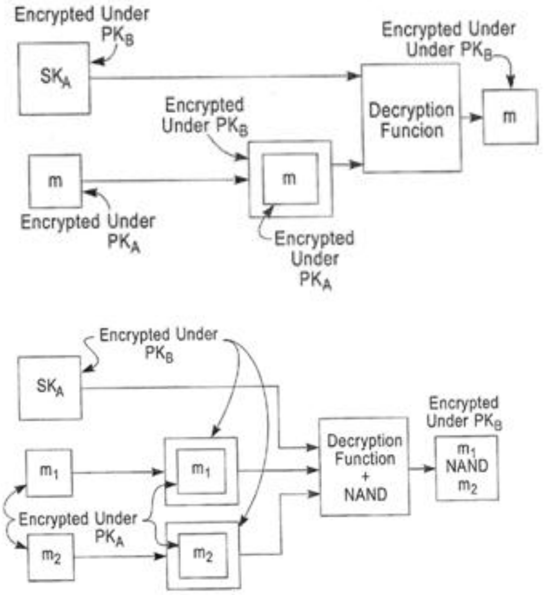

Fig 2 & 3 : FHE scheme

3. DIFFERENTIATING

In this part we compare and analyze Dijk's and Brakerski's schemes for the security, efficiency and complexity factors.

A. Encryption over Integers using Fully Homomorphism

Security: The security of this scheme is based on the hardness of the approximate-gcd problem. With appropriate selection of parameters the scheme has proved to resist different types of attacks to recover the secret key including brute-force attack with atleast 2λ time. Nevertheless by using lattice reduction algorithm, it has been examine that the scheme can be attacked to retrieve the plaintext from ciphertext. The parameters settings which have been used in the attack were considered to be appropriate by the Dijk's scheme.

Performance: The noise factor grows large when addition and multiplication operations are performed, it doubles on addition and squares on multiplication. By multiplying ciphertexts yields a noise factor with a size equals ~2dn where n is the number of performances. The ciphertext original message cannot be amended, if the noise factor gets higher than q/2. On the other hand, in order to better the security of the scheme the cipher text was choose to have a prominent value n6, this value gain with multiplication which also outcomes on degraded efficiency.

Complexity: Reducing the Gentry's scheme complexity was the main purpose of developing this scheme. The complexity of the somewhat scheme was reduced by using additions and multiplications over integers instead of ideal lattices. Although one of the meaning of this scheme is that it examined out that different mathematical functions and theories can be used to build a fully homomorphic encryption scheme using Gentry's blueprint.

B. Encryption without Bootstrapping using Fully Homomorphism

Security: The security of this scheme is based on the hardness of lattice problems with quasi-polynomial estimation components. The attained level of security has not improved from original FHE scheme as it remains 2λ time against known lattice attacks. Because the scheme is comparatively fresh, it is likely too early to sustain its security strength against different types of attacks with high assurance.

Efficiency: Brakerski has developed a novel noise management technique that controlled the noise level so that it increases linearly with multiplication instead of exponential function. In a theoretical manner, this scheme measures former bootstrapping-based FHE schemes operation-wise. This scheme also permitted for L-level arithmetic circuit to be measured with Õ(λ.L3) per-gate computation or rather than of Ω(λ4) which is a large polynomial in the security parameter quantity. The removal of the bootstrapping technique has also resulted on real cost reduction as the cost of bootstrapping in only θ(λ) time was Ω(λ4). From this, it provides us for measuring deeper circuits with a less cost. Applying batching and bootstrapping as optimization techniques can achieve a better per-gate computation of Õ(λ2) independent of number of levels.

Complexity: When compared to FHE over Integers, Brakerski's scheme applies most difficult mathematical algorithms and notations as a result of using Ring LWE instead of working with integers. However the removal of bootstrapping technique has reduced decryption function and calculations.

4. CONCLUSION

Even though Dijk's scheme has succeeded to reduce the complexity of Gentry's original scheme, his scheme has transmitted the efficiency limitations of the original scheme in term of disturbance, size of cipher text and encryption keys, as well the time needed for encoding, decoding and evaluation functions. On the other hand Brakerski's scheme introduced new novel technique for noise management which permitted for measuring deeper circuits with the same cost as earlier, this method is used in later schemes to improve FHE schemes performance. Both schemes have inspired many researchers to search for new mathematical approaches and techniques to improve the performance and efficiency while meeting the security requirements. At the moment the available schemes provide a great potential for cloud computing but they still have lots of scope for improvement and enhancement before they can be ready for practical use in the cloud computing.

REFERENCES

[1] M. Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, ―Fully homomorphic encryption over the integers, in EUROCRYPT, 2010, pp. 24–43.

[2] N. Howgrave-Graham, ―Approximate integer common divisors, in CaLC, 2001, pp. 51–66.

[3] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, ―Fully homomorphic encryption without bootstrapping,Cryptology ePrint Archive, Report 2011/277, 2011.

[4] Vadim Lyubashevsky, Chris Peikert, and Oded Regev, ―On ideal lattices and learning with errors over rings, in EUROCRYPT, volume 6110 of Lecture Notes in Computer Science, pages 1–23, 2010.

[5] Oded Regev, ―On lattices, learning with errors, random linear codes, and cryptography, in Harold N. Gabow and Ronald Fagin, editors, STOC, pages 84–93. ACM, 2005.

[6] Gu Chunsheng, ―Attack on Fully Homomorphic Encryption over the Integers, Cryptology ePrint Archive, Report 2012/157, 2012.

[7] National Institute of Standards and Technology - Computer Security Resource Center - www.csrc.nist.gov.

[8] R. Rivest, L. Adleman, and M. Dertouzos, ―On data banks and privacy homomorphisms, in Foundations of Secure Computation. Academic Press, 1978, pp. 169–177.

[9] T. E. Gamal, ―A public key cryptosystem and a signature scheme based on discrete logarithms, in CRYPTO, 1984, pp. 10–18.

[10] Gu Chunsheng, ―Attack on Fully Homomorphic Encryption over the Integers, Cryptology ePrint Archive, Report 2012/157, 2012.

BIOGRAPHIES

V.SATISH KUMAR received his B.Tech degree in Computer Science & Engineering from SK University, Anantapur, India, in 2006. Currently pursuing M.Tech in Computer Science and Engineering at Dr.KVSR Institute of Technology, Kurnool, India.

V.TRILIK KUMAR received his M.Tech in Computer Science & Engineering from Jawaharlal Nehru Technological University, Anantapur, India in 2008. He is working as an Assoc.Professor at DR.K.V.S.R.I.T, Kurnool, India