DOI: http://dx.doi.org/10.26483/ijarcs.v8i7.4573

ISSN No. 0976-5697

Volume 8, No. 7, July – August 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

STUDY OF LATTICE BASED FHE FOR CLOUD DATA SECURITY

Ms.Aarti Dadheech
Assistant Prof., Computer Science Dept.
JIET College of Engineering
Jodhpur, India

Abstract: Cloud Computing is an transpiring trend in the modern world. It is a way of holding the Internet to use software or other IT services on demand. Due to its fast growth and popularity, number of users deposit their data and applications on the cloud. The impressive growth in cloud computing has proved to be promising innovation and more suitable for storing data and applications remotely. But its uses improvement is hindered by the security issue. Cloud doesn't provide more security for its services and storage purpose. The traditional security approach of encryption doesn't make cloud fully secure. So there is a need to develop such a technique which increases the security level of cloud. In order to solve the problem of data security in cloud computing system, lattice-based cryptographic schemes implements the so called "Fully Homomorphic Encryption (FHE) scheme", which allows processing directly on encrypted data and holds the promise eventually to solve the security problems with cloud computing. In this paper we survey on the existing lattice based FHE encryption techniques. Fully homomorphic encryption is a good solution to enhance security measures of cloud system that handles critical data. This makes cloud computing more stable and solid.

Keywords: Cryptography, Cloud computing, Lattice based cryptography, Fully Homomorphic Encryption, Security.

I. INTRODUCTION

In cryptography the term encryption refers to converting the original data into human unreadable form (encoding). By encoding the data only the authorized person can decode the original data and read it. Thus data confidentiality is achieved by the encryption. In this paper, the proposed algorithms for the lattice based fully homomorphic encryption (FHE) of data in cloud computing is reviewed. Cloud computing is an internet-centric computation that provides shareable computer data and processing resources on demand. It provide users and organizations with potentiality to store and handle their data in either private self-owned, or third-party data centers that may be located far from across a city to across the world. The use of cloud computing has elevated speedily in many organizations. Although cloud computing has become a mature service model, the choosing of its services by customers (businesses, consumers, etc.) is limited by concerns about the loss of privacy of their private data. There are some security issues in cloud computing such as data security, third-party control, and privacy. Encryption of data could solve this issue. If all data stored in cloud were encrypted using traditional cryptosystems, this would effectively solve the three above issues. But if the consumers want to manipulate their encoded data in the cloud, they have to share the secret key with cloud provider to decrypt it before executing the required operations. To solve this type of issue, it is necessary to use a cryptosystem based on homomorphic encryption, since these cryptosystems allow performing computations on encrypted data without sharing the secret key needed to decrypt the data.

Homomorphic encryption is a kind of encryption that enable computations to be carried out on ciphertext, thus generating an encrypted result which, when decrypted, matches with the result of operations done on the plaintext [1]. Homomorphic encryption schemes have two categories: Partially Homomorphic Encryption (PHE) and Fully Homomorphic Encryption (FHE) schemes. PHE schemes, such as RSA, ElGamal, Paillier, etc., allow performing either addition or multiplication on encrypted data i.e only one type of operation with an unlimited number of times. FHE, in this case, both operations can be carried out at same time [2]. Due to this, security mechanism for encrypted data is improved. The currently such system is a FHE system based on ideal lattices developed by Craig Gentry in 2009. If the data is always in encrypted form in the cloud, then control is not lost, and the concerns are removed. Fully homomorphic encryption schemes, although they may be limiting in efficiency at its current stage, enable many important applications, such as secured cloud searching and verifiable outsourced computing. Nonetheless, just like all other inventions at the initial stage, the fully homomorphic encryption is also young, prospective, and needs further research. The paper is organized as follows: Section 2 discusses basic of lattices, prerequisite for lattice based FHE system. The section 3 briefs the known Homomorphic Encryption methods. Section 4 focuses on related work on Fully Homomorphic Encryption methods and shows the implementation of FHE on cloud. Section 5 concludes with an utter need of efficient cryptographic methods like Fully Homomorphic Encryption in cloud data security for further study and research.

II. PRELIMINARIES

Lattice-based cryptography is a new approach towards cryptographic protection of data in computer systems. It is a counterpart of more commonly known, thoroughly tested and smoothly-working traditional algorithms (such as RSA, DSA, AES), which seem so far to fulfill their purpose more than adequately. A most appealing property of lattice based cryptography is related to the fact that there are lattice-based schemes, which are resilient to cryptanalysis, conducted with the help of quantum computers.

Lattices were first studied by mathematicians Joseph Louis Lagrange and Carl Friedrich Gauss. In 1996, Miklos Ajtai and

Micciancio discussed the use of lattices as cryptography primitive. Micciancio defined lattices as general class of cyclic lattices (ideal lattice). A lattice L is a set of points in the n dimensional Euclidean space R_n with a strong property of periodicity, as shown in Fig. 1.

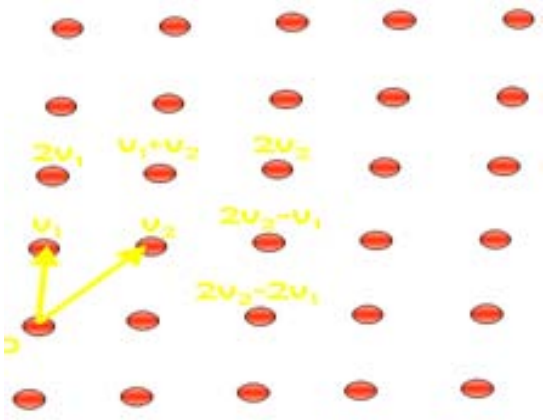

Figure 1. Lattices

We call v_1… vn (vectors) a basis of L .A basis of L is a set of vectors such that any element of L is uniquely represented as their linear combination with integer coefficients. When n is at least 2, each lattice has infinitely many different bases. All lattices over R_n have infinitely many elements, whereas in cryptography entities like the ciphertext, public key, and private key must be taken from a finite space (bit strings of some fixed length). Therefore the lattices used for cryptography are actually lattices over a finite field. There are two types of lattice based mathematical problems. They are the Shortest Vector Problem (SVP) and the Closest Vector Problem (CVP). SVP: Given a basis of a lattice, find the shortest vector in the lattice.CVP: Given a basis of a lattice and a vector not in the lattice, find the lattice vector with the least distance to the first vector. Lattice based cryptography refers to any system whose security depends on computational assumptions based on lattices [3].

III. HOMOMORPHIC ENCRYPTION (HE)

First In cloud, the foremost concern is of maintaining both confidentiality and privacy of owner's data from untrusted users. The concept of homomorphism introduced in 1978, by Rivest, can be used for securing the data stored in cloud from illegitimate users. HE permits processing of encrypted data on a remote storage without decrypting it. This method is efficient for the cloud paradigm as the users can benefit of the plus offered by the Cloud and they also protect the privacy issues of storage and processing their confidential data by an un-trusted third party.

Two messages m_1 and m_2 are encrypted by using any known encryption method E with public key or private key pk, where C_1 and C_2 are their corresponding cipher texts (i.e. $C_1 = E_{pk}(m_1)$ and $C_2 = E_{pk}(m_2)$). The HE scheme performs computation like addition and multiplication between C_1 and C_2 without decryption. The obtained result is also in encrypted form[4].

Further HE methods can be broadly classified into Partial Homomorphic Encryption (PHE) and Fully Homomorphic Encryption (FHE). PHE performs only a limited number of operations for example either addition or multiplication on encrypted data. FHE performs both addition and subtraction operations on encrypted data many times. Homomorphic encryption is distinguish, according to the operations that allows to assess on raw data, the additive Homomorphic encryption (only additions of the raw data) is the Pailler and Goldwasser-Micalli (GM) cryptosystems, and the multiplicative Homomorphic encryption (only products on raw data) is the RSA and ElGamal cryptosystems.

IV. FULLY HOMOMORPHIC ENCRYPTION (FHE)

The flexibility and potential of lattice cryptography is well exemplify by Gentry's recent discovery of a fully homomorphic encryption scheme based on lattices [5]. A FHE scheme is a public key cryptosystem such that arbitrary computations can be carried out on ciphertext, without decrypting them, and in fact without even knowing the decryption key. The potential applications scenarios for FHE are countless: consider for example submitting encrypted queries to a database, or running an encrypted program in a cloud computing framework. In all such situations, FHE would let the safe use of remote (untrusted) on-line services while keeping sensitive data of user under strict and direct control. FHE is so powerful that till recently many experts in the cryptographic research community doubted if fully homomorphic encryption schemes even existed. For all types of calculation on the data stored in the cloud, we must opt for the fully Homomorphic encryption which is able to execute all types of operations on encrypted data without decryption as shown in Fig. 2 [6].

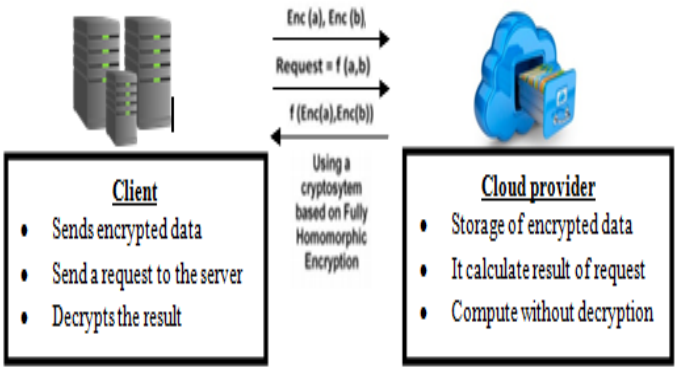

Figure 2. Fully Homomorphic Encryption applied to cloud computing

The application of FHE is an important stone in Cloud Computing security more generally; we could do the calculations on confidential data to the Cloud server, keeping the secret key that can decrypt the result of calculation.

Gentry's method can be broken into three main steps [6] and implementation is shown in Fig. 3 [7].

Step 1: Constructing an encryption scheme using ideal lattices that is somewhat homomorphic, which means it is limited to evaluating low-degree polynomials over encrypted data.

Step 2: "Squashing" the decryption circuit of the original somewhat homomorphic scheme to make it bootstrappable (a series of self-sustaining processes). While encrypting, include extra data to help decrypter for decryption.[8]

Step 3: Bootstrapping the slightly augmented original scheme of 2nd step to yield the fully homomorphic encryption scheme by refreshing.

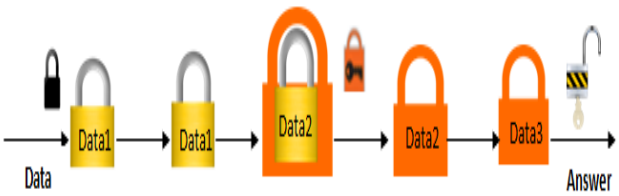

Figure 3. Craig Gentry Implementation of FHE

A. *Algorithm*

(1).Key Generation:

- Takes as input a security parameter λ and a positive integer d as the depth of circuit. For length, l = l(λ) .
- Sets (sk_i, pk_i) ← KeyGen (λ) for i ∈ [0, d]
- sk_{ij} ← EncryptE (pk_{i-1} , sk_{ij}) for i ∈ [1, d], j ∈ [1, l`], where $sk_{i1}$, . . , $sk_{il}$` is the bit representation of ski . $sk^{(d)}$ = sk0 and $pk^{(d)}$ ← (<pk_i>,<sk_{ij}>).

(2).Encryption:

- Input a public key pkd and a plaintext π ∈ P.
- Ciphertext ψ ← Encrypt (pk_d , π).

(3).Decryption:

- Input a secret key sk_d and a ciphertext ψ (which should be an encryption under pk0).
- Outputs Decrypt (sk0, ψ).

(4).Evaluation:

- Takes as input a public key pkυ, a circuit C_υ of depth at most υ gates, and a tuple of input ciphertexts Ψ_υ (where each input ciphertext should be under pk_υ).
- Check if each wire in C_υ connects gate at consecutive levels; if not add identity gates.
- If υ=0, it outputs , ψ_0 and terminates,

Else

- Sets ($C^\dagger_{\upsilon-1}$, $\psi^\dagger_{\upsilon-1}$)←Augment υ($pk_\upsilon$,C_υ, ψ_υ)
- Sets ($C_{\upsilon-1}$, $\psi_{\upsilon-1}$) ←Reduceυ-1(pk_υ, C_υ, ψ_υ)
- Evaluateυ-1 ($pk_{\upsilon-1}$, $C_{\upsilon-1}$, $\psi_{\upsilon-1}$)

(5). Augment$_\upsilon$:

- Takes as input a public key pk_υ, a circuit C_υ of depth at most υ gates, and a tuple of input ciphertexts Ψ_υ (where each input ciphertext should be under pk_υ).
- Let $\psi\dagger_{\upsilon-1}$ be the tuple of cipher texts formed by replacing each input ciphertext $\psi \in \psi_\upsilon$ by the tuple (sk_υ, ψ_υ), where ψ_j ← Encrypt (($pk_{\upsilon-1}$), ψ^j) and the ψ^j from the properly-formatted representation of ψ as elements of P. It outputs ($C^\dagger_{\upsilon-1}$,$\psi^\dagger_{\upsilon-1}$).

(6). *Reduce$_\upsilon$*:

- Takes as input a public key pkυ, a circuit C_υ of depth at most υ gates, and a tuple of input ciphertexts Ψ_υ.
- sets C_υ to be the sub-circuit of $C\dagger_\upsilon$ consisting of the first υ levels
- sets ψ_υ to be the induced input ciphertext of C_υ

This scheme is impractical, as increase in the key size results in a large increase in the ciphertext size and thus, increases computation time. After the first FHE system proposal, in 2009, by Craig Gentry, other methods to achieve fully homomorphic encryption have been discovered, but interestingly they are all based on lattices [9]. The other lattice based fully homomorphic encryption schemes are listed in Table 1[10].

Table I. Other Lattice Based FHE schemes

HE Scheme	*Type of HE*	*Operation*	*Principle*	*Concern*
BGV	Leveled FHE	Any Circuit	Modulus switching & LWE,RLWE	Large memory requirement for storing multiple keys
FV	Leveled FHE	Any Circuit	Scale invariant RLWE on BGV	Inability to stay in double CRT Form
YASHE	Leveled FHE	Any Circuit	invariant RLWE on NTRU	Not able in double CRT Form

V. CONCLUSION

The cloud data security based on FHE is a new concept of security with respect to the data confidentiality, which enables providing results of operation on encrypted data without knowing the raw data on which the operation was carried out. FHE is a promising aspect in cryptography. Inspite of its interesting properties, it is a bit limited regarding its computation abilities and practical implementations. In this paper we survey on the existing FHE encryption techniques based on lattices for cloud data security, that protect the data for the complete life cycle from the start to the end in the cloud computing. In future work, aim to propose a improved scheme that will contain the more security features while overcoming disadvantage and open issues in existing scheme.

VI. ACKNOWLEDGMENT

I would like to show my gratitude to Dr. Anil Gupta for his guidance. It gives me immense pleasure to thank my family without whose support, this work would not be possible.

VII. REFERENCES

[1] Iram Ahmad and Archana Khandekar," Homomorphic Encryption Method Applied to Cloud Computing", International Journal of Information & Computation Technology.ISSN 0974-2239 Volume 4, Number 15 (2014), pp. 1519-1530.

[2] Dr. Mohammad Miyan. "FHE Implementation of Data in Cloud Computing".

[3] Prasanna B T, C B Akki. "A Comparative Study of Homomorphic and Searchable Encryption Schemes for Cloud Computing."

[4] Feng Zhao , Chao Li , Chun Feng Liu. "A cloud computing security solution based on fully homomorphic encryption."

[5] C. Gentry. "Fully homomorphic encryption using ideal lattices". In Proceedings of STOC, pages 169–178, 2009.

[6] Kamal Benzekki, Abdeslam El Fergougui, Abdelbaki El Belrhiti El Alaoui," A Secure Cloud Computing Architecture Using

Homomorphic Encryption", International Journal of Advanced Computer Science and Applications,Vol. 7, No. 2, 2016.

[7] Shashank Bajpai and Padmija Srivastava ."A Fully Homomorphic Encryption Implementation on Cloud Computing". IJICT. ISSN 0974-2239 Volume 4, Number 8 (2014), pp. 811-816.

[8] Edlyn Teske-Wilson, Homomorphic Cryptosystems, University of Waterloo

[9] ABBAS ACAR, HIDAYET AKSU, and A. SELCUK ULUAGAC, "A Survey on Homomorphic Encryption Schemes: Theory and Implementation".

[10] Gaurav Somani, Sourabh Garg," Homomorphic Encryption Algorithms for Securing Data against Untrusted Cloud", IJARCCE,Vol 5,Issue 7, July 2016.