Chinese Remainder Theorem based Fully Homomorphic Encryption over Integers

Vinod Kumar[1]
[1]Department of Electronics and Communication, University of Allahabad,
Allahabad, UP, India

Neelam Srivastava[2]
[2]Department of Electronics Engineering, REC Kannauj,
Kannauj UP, India

Abstract- **Fully homomorphic encryption has attracted the attention of researcher in the field of cryptography. Fully Homomorphic encryption allows to perform arbitrary operations on encrypted data and it produce result in encrypted form. It produces the same result as well that the operations performed on the data before encryption. The first fully homomorphic encryption scheme has given by Craig Gentry in his Phd thesis in 2009. Though it was only a theoretical approach but later on much progress has been made up on it. This paper proposes a fully Homomorphic Encryption Scheme based on Chinese remainder theorem with probabilistic encryption over Integers for better security. The proposed scheme supports both additive and multiplicative homomorphism property. Our scheme provides better performance in terms of computational overhead and response time.**

Key words: Encryption, Homomorphism, Fully Homomorphic encryption, security.

I. INTRODUCTION

With the tremendous advancement and progress in computing technology, especially when outsourcing the computation to a third-party server like cloud has become the norm, then the security standards also need to be updated. Cryptography is one such branch which ensures the integrity, confidentiality and availability of the data but we want it to preserve the privacy of the data also. Traditional cryptographic algorithms cannot openly work on encrypted data, but homomorphic encryption algorithms can work, and the result can encrypt automatically.

The homomorphic cryptosystems are classified into two main category namely partially homomorphic systems and fully homomorphic systems. The partial homomorphic encryption systems support only single homomorphism property either additive homomorphic or multiplicative homomorphic. The fully homomorphic encryption systems support both homomorphism property. In last 30 years so many partially homomorphic encryption schemes have been proposed. Those partially homomorphic systems divided into two main categories namely additive homomorphic systems and multiplicative homomorphic system based on operations they support. The partially homomorphic systems defined over group because group supports only single operation while on the other hand fully homomorphic systems defined over ring because ring supports two operations. The partially homomorphic systems are RSA and P. Paillier cryptosystems etc. while Craig Gentry has given first fully homomorphic system based on lattice in 2009.

The search for a Fully Homomorphic Encryption (FHE) system had started about 40 years ago, soon after RSA (Rivest, Adleman and Shamir) have published in 1978[1]. RSA was a ground-breaking algorithm at that time as it introduced the notion of asymmetric cryptography or Public key cryptography for the first time. In such encryption schemes, the secret decryption key enables to decrypt the encrypted message and one can read the entire message, but without this secret decryption key, the encrypted message is totally useless. This situation of the case raised an intriguing question, first introduced by Rivest, Adleman and Dertouzos in 1978: Can we perform arbitrary operations on encrypted data without decrypting and the result of operations remains in encrypted form. Since then, several attempts have been made to develop such a system. But most of the researches that have been received are only partially homomorphic. i.e. the algorithms proposed during this period have achieved only restricted homomorphic capacities. Either they are multiplicatively homomorphic like RSA [1] or additively homomorphic like Goldwasser–Micali [2] but not have both properties.

The scheme which could support both additive and multiplicative homomorphic properties could be considered fully homomorphic as these both operations are sufficient for any arbitrary computation. A homomorphic cryptosystem is a technique with additional properties in which an efficient algorithm exists to perform arbitrary operations on the cipher-

text of input messages. No information about msg_1, ..., msg_t or f(msg_1, ..., msg_t), or any intermediate plaintext does not leak. The inputs, outputs and intermediate values are always encrypted and therefore are useless for an adversary.

Fully Homomorphic cryptosystem is a special kind of cryptographic technique, which allows to execute arbitrary operations on cipher-text without decrypting in advanced. Thus, producing an encrypted result, which, when decrypting, matches the result of the operations done on the data before encryption. Nowadays cloud computing techniques are increasing rapidly; a key challenge is to create assurance that the cloud can maintain very sensitive data safely. After encryption the data is uploaded to the cloud. However, the encrypted data uploaded must decrypt before performing arbitrary operations. Fully Homomorphic encryption eliminates this problem by allowing computations on encrypted data and produce results in encrypted form as well. This useful feature of the homomorphic encryption scheme has been known for over 30 years.

This paper proposes an efficient fully homomorphic cryptosystem over integers based on Chinese remainder theorem for better security. The proposed scheme greatly reduces the computation complexity. The rest of the paper is organized as follows: Section 2 provides brief survey of related work. Section 3 provides brief introduction of Chinese remainder theorem cryptosystem and Euler's theorem. Section 4 presents the proposed fully homomorphic cryptosystem. To verify the proposed cryptosystem, the numerical example is presented in section 5. The theoretical analysis is presented in section 6. Section 7 provides the experimental results. Finally, conclusion is presented in section 8.

II. RELATED WORK

The fully homomorphic encryption has a lot of attention in the field of cryptography. A variety of fully homomorphic encryption schemes have been proposed [1-16]. The key challenges of various existing schemes are higher computation overhead and security. In order to deal with these issues, it is necessary to analyse the various fully homomorphic cryptosystems available in literature [1-16]. The review of some well-known fully homomorphic cryptosystem is presented as follows.

Craig Gentry [4] has proposed Ideal lattices based fully homomorphic encryption scheme. The scheme has decryption algorithm with less computation complexity. Craig Gentry [5] has proposed firrst fully homomorphic encryption algorithm to solve central open problem of homomorphism in cryptography. The scheme allows to perform arbitrary operations over encrypted data without decrypting. The scheme uses difficult problems on ideal lattices. The proposed solution is somewhat Homomorphic Boostrable Encryption Scheme, which works when the scheme has its own decryption function.

Guangli, Xiang et al. [6] have proposed an algebra homomorphism encryption scheme which is based on the hardness of division of very large integer. To incorporate the property of probabilistic encryption, the scheme uses random number in the encryption function. Due to having probabilistic encryption property, the encryption function of the scheme has different results for two different encryptions for the same plaintext but has same results for decryptions. Chen Liang et al.[7] have proposed a RSA based exponential homomorphic encryption scheme. The scheme is also an algebraic homomorphic encryption scheme. The scheme has additive, multiplicative and exponential homomorphism properties.

M. van Dijk et al. [8] have given a somewhat homomorphic scheme using elementary integer arithmetic. To convert the proposed scheme into a fully homomorphic encryption, the scheme uses Gentry's algorithms. Brakerski Zvika et al.[9] have presented learning with errors(LWE) assumption based a fully homomorphic encryption scheme. The scheme reduced the size of cipher-texts and greatly minimized the computational complexity of decryption function. Darko Hrestak et al.[11] have discussed about the merit and demerits homomorphic encryption and give lot of attention to fully homomorphic encryption cryptosystems for cloud based systems. The authors also discussed about open source library by IBM for homomorphic encryption.

Reem Alattas [13] has proposed a Fermat's Little Theorem based Algebraic Homomorphic Encryption Scheme for better security on cloud computing. Kangavalli, R. et al. [13] have proposed a Mixed Homomorphic Encryption Scheme for Secure Data Storage in Cloud uses the concept of Byte Encryption that can be adopted in any scheme for additional security and compressing the size of ciphertext. But as the basis for this, it uses BGV scheme which is lattice based and uses several expensive operations like modulus switch which are simply too costly in terms of both time and cost.

III. PRELIMINARIES

A. Chinese remainder theorem

Let $m_1, m_2 \ldots\ldots\ldots m_n \in \mathbb{N}$ and for $i \neq j$, $\gcd(m_i, m_j) = 1$ then from Chinese remainder theorem, we have

$$\mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \cdots\cdots\cdots \times \mathbb{Z}_{m_n} \cong \mathbb{Z}_{m_1 \cdots\cdots m_n}$$

This implies that, given positive integers $k_1, k_2, k_3 \ldots\ldots\ldots k_n$, there exists an unique solution X modulo $N = m_1 \times m_2 \times \cdots\cdots \times m_n$ such that X satisfies the following congruences

$$X \equiv k_1 \bmod m_1$$

$$X \equiv k_2 \bmod m_1$$

$$\vdots$$

$$X \equiv k_n \bmod k_n$$

To compute the unique solution X, the member computes $N_i = \frac{N}{m_i}$ for $1 \leq i \leq n$

Then for each i, $\gcd(m_i, N_i) = 1$ and there are integers u_i and v_i with $u_i N_i + v_i m_i = 1$.

then $X \equiv \sum_{i=1}^{n} k_i u_i N_i \pmod N$ is the desired solution.

B. Euler's Theorem

Euler's theorem has two versions which are as follows:

1. $a^{\emptyset(n)} \equiv 1 \pmod n$, If a and n are relatively prime to each other.
2. $a^{k \times \emptyset(n)+1} \equiv a \pmod n$ Where k is an integer, a < n and $n = p \times q$. The second version removes the condition of co-prime for a and n.

The Euler's theorem can be used to compute the exponentiations very quickly.

IV. PROPOSED FULLY HOMOMORPHIC ENCRYPTION SCHEME

The proposed fully homomorphic encryption scheme has three phases known as key generation phase message encryption phase and message decryption phase. The model of proposed scheme is given in figure 1.

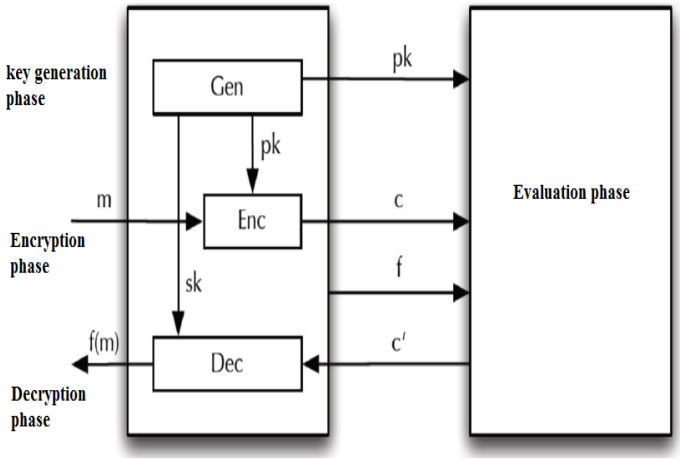

Figure 1: Model of Proposed FHE scheme

Our proposed scheme evaluates an arbitrary number of additions and multiplications. The proposed Fully Homomorphic Encryption scheme is described as follows:

Essential conditions for selection of different parameters

1. The randomly selected constant k does not divide $\varphi(z)$ and $\varphi(n)$
2. The $\varphi(x)$ does not divide $k \times \varphi(z)$.
3. The randomly selected k should be prime and $k >$ max (p, q, r) and $k <$ n

Key Generation

1. Generate two distinct and large primes p and q
2. Compute n = $p \times q$ and $\emptyset(n)$
3. Generate another large prime z for which Greatest Common Divisor (GCD) with n should be 1 i.e., $\gcd(n, z) = 1$
4. Compute $x = n \times z$
5. Compute $\mu = ((p^{-1} \bmod q) \times p) + ((q^{-1} \bmod p) \times q)$

Messages Encryption

The messages M_1 & M_2 should be less than n. the messages addition ($M_1 + M_2$) and multiplication ($M_1 * M_2$) should also be less than < *n.* To maintain probabilistic encryption, the scheme uses two random large integers.
$C_1 = \mu \times (M^{k_1 \times \emptyset(n)+1} \bmod x)$ and $C_2 = \mu \times (M^{k_2 \times \emptyset(n)+1} \bmod x)$; where k_1 and k_2 are random integers and C_1 and C_2 are cipher texts. Compute result C_3 by performing arbitrary functions on cipher-texts C_1 &C_2

Message Decryption
$M = C_3 \bmod n$, Where C_3 is cipher-text, n is decryption key and M is resultant message.

Proof of Correctness of proposed fully homomorphic encryption scheme

$$C = \mu \times \left(M^{k \times \emptyset(n)+1} \, mod \, x\right)$$

$D = C \, mod \, n$

$$= \left(\mu \times \left(M^{k \times \emptyset(n)+1} \, mod \, x\right)\right) mod \, n$$

$$= \left((p^{-1} \, mod \, q) \times p\right) + \left((q^{-1} \, mod \, p) \times q\right) \times \left(\left(M^{k \times \emptyset(n)+1} \, mod \, x\right)\right) mod \, n$$

$$= \left(M^{k \times \emptyset(n)+1} mod \, n\right) mod \, x;$$

Since, $\left((p^{-1} \, mod \, q) \times p\right) + \left((q^{-1} \, mod \, p) \times q\right) mod n \equiv 1 \, mod \, n$

$= (M \,) mod \, x$

Since, $\left(a^{k \times \emptyset(n)+1}\right) \equiv a(mod \, n)$ by Euler's Theorem

$= (M \,) mod \, x$

$= M, \quad M < x$ *(Hence proved)*

Homomorphism

To prove the fully homomorphism let us considered two cipher text C_1 and C_2 of messages M_1 and M_2 respectively. To maintain probabilistic encryption property, the proposed scheme used two large random integers k_1 and k_2 for decryption. For fully homomorphism, the scheme should completely fulfill the additive and multiplicative Homomorphic property. The proof of both homomorphic properties for proposed solution is presented below.

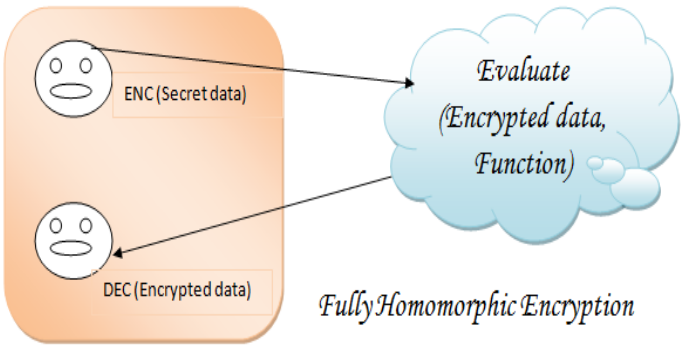

Figure 2 Fully Homomorphic encryption

Multiplicative homomorphism

The encryption algorithm will be multiplicative homomorphic if it satisfies the following property.

$$M_1 \times M_2 = DEC[ENC(M_1) \times ENC(M_2)]$$

Where, *ENC* is the encryption function and *DEC* is the decryption function.

Proof:

$C_1 = \mu \times (M_1{}^{k_1 \times \emptyset(n)+1} \, mod \, x)$, $C_2 = \mu \times (M_2{}^{k_2 \times \emptyset(n)+1} \, mod \, x)$

$$C_1 \times C_2 = \mu \times (M_1{}^{k_1 \times \emptyset(n)+1} \, mod \, x) \times \mu \times (M_2{}^{k_2 \times \emptyset(n)+1} \, mod \, x)$$

$$D(C_1 \times C_2) = (C_1 \times C_2) \, mod \, n$$

$$= \left[\mu \times \left(M_1{}^{k_1 \times \emptyset(n)+1} \, mod \, x\right) \times \mu \times \left(M_2{}^{k_2 \times \emptyset(n)+1} \, mod \, x\right)\right] mod \, n$$

$$= \left[\mu \times \left(M_1{}^{k_1 \times \emptyset(n)+1} \, mod \, x\right) mod \, n \times \mu \times \left(M_2{}^{k_2 \times \emptyset(n)+1} \, mod \, x\right) mod \, n\right]$$

$$= \left[\mu \times \left(M_1{}^{k_1 \times \emptyset(n)+1} \, mod \, n\right) mod \, x \times \mu \times \left(M_2{}^{k_2 \times \emptyset(n)+1} \, mod \, n\right) mod \, x\right]$$

$$= \left[\left(M_1{}^{k_1 \times \emptyset(n)+1} \, mod \, n\right) mod \, x \times \left(M_2{}^{k_2 \times \emptyset(n)+1} \, mod \, n\right) mod \, x\right]$$

Since; $\mu \, mod \, n \equiv 1 \, mod \, n$

$= [(M_1 \, mod \, x) \times (M_2 \, mod \, x)]$

Since; $\left(a^{k \times \emptyset(n)+1}\right) \equiv a(mod \, n)$

$= M_1 \times M_2$

Additive Homomorphism:

The encryption algorithm will be additive homomorphic if it satisfies the following property.

$$M_1 + M_2 = DEC[ENC(M_1) + ENC(M_2)]$$

Where, *ENC* is the encryption function and *DEC* is the decryption function.

Proof:

$C_1 = \mu \times (M_1{}^{k_1 \times \emptyset(n)+1} \, mod \, x)$, $C_2 = \mu \times (M_2{}^{k_2 \times \emptyset(n)+1} \, mod \, x)$

$$C_1 + C_2 = \mu \times \left({M_1}^{k_1 \times \emptyset(n)+1} \bmod x\right) + \mu \times \left({M_2}^{k_2 \times \emptyset(n)+1} \bmod x\right)$$

$$D(C_1 + C_2) = (C_1 + C_2) \bmod n$$

$$= \left[\mu \times \left({M_1}^{k_1 \times \emptyset(n)+1} \bmod x\right) + \mu \times \left({M_2}^{k_2 \times \emptyset(n)+1} \bmod x\right)\right] \bmod n$$

$$= \left[\ \mu \times \left({M_1}^{k_1 \times \emptyset(n)+1} \bmod x\right) \bmod n + \mu \times \left({M_2}^{k_2 \times \emptyset(n)+1} \bmod x\right) \bmod n\right]$$

$$= \left[\ \mu \times \left({M_1}^{k_1 \times \emptyset(n)+1} \bmod n\right) \bmod x + \mu \times \left({M_2}^{k_2 \times \emptyset(n)+1} \bmod n\right) \bmod x\right]$$

$$= \left[\left({M_1}^{k_1 \times \emptyset(n)+1} \bmod n\right) \bmod x + \left({M_2}^{k_2 \times \emptyset(n)+1} \bmod n\right) \bmod x\right]$$

Since; $\mu \bmod n \equiv 1 \bmod n$

$= [(M_1) \bmod x + (M_2) \bmod x]$

Since; $\left(a^{k \times \emptyset(n)+1}\right) \equiv a(\bmod n)$

$= M_1 + M_2$

V. Working Example

Example: Consider two messages M1 & M2 are as follows; M1=4 and M2= 6.

Generate two prime numbers p and q

p =13 and q =23 and compute $n = p \times q$ i.e. n =299

Computes $\emptyset(n)$ using Euler Totient function

$\emptyset(299) = 264;$

Next generates another prime number z where *gcd (n, z) =1*

z = 11, and computes *gcd(299,131);*

gcd(299,11)=1 which satisfies the condition for n and z

Next, computes x= $n \times z$

x= 299×11; $x = 3289$

Next, generates two random integers, k_1 and k_2.

k_1*=3* and k_2*=4*, and two messages $m_1 = 4$ and $m_2 = 6$, which fulfill the condition that $(m_1 + m_2)$ & $(m_1 * m_2)$ less than n

Now $c_1 = \mu \times {m_1}^{k_1 \times \emptyset(n)+1} \bmod x$

$c_1 = 300 \times 4^{3 \times \emptyset(299)+1} \bmod 3289$

$c_1 = 300 \times 4^{3 \times 264+1} \bmod 3289$

$c_1 = 1798$

And $c_2 = \mu \times {m_2}^{k_2 \times \emptyset(n)+1} \bmod x$

$c_2 = 300 \times 6^{4 \times \emptyset(299)+1} \bmod 3289$

$c_2 = 300 \times 6^{4 \times 264+1} \bmod 3289$

c_2 *=2697*

Additive Homomorphic property;

Let c_3 is the result of computation of additive function on two cipher texts c_1 and c_2

$c_3 = c_1 + c_2$; $c_3 = 1798 + 2697$; $c_3 = 4495$

The decryption of c_3 produces message m_3 which is same as addition of two messages m_1 and m_2.

$m_3 = c_3 \bmod n$

$=4495 \bmod 299$

$= 10$, which is same to $m_1 + m_2$ *(i.e. 4+ 6=10)*

Multiplicative homomorphism:

Let c_4 is the result of computation of multiplicative function on two cipher texts c_1 and c_2

$c_4 = c_1 \times c_2$

$c_4 = 1798 \times 2697$

$= 4849206$

The decryption of c_4 produces message m_4 which is same as multiplication of two messages m_1 and m_2.

$m_4 = c_4 \bmod n$

$= 4849206 \bmod 299$

$= 24$, which is same to $m_1 \times m_2$ *(i.e. 4× 6=24)*

VI. RESULTS AND COMPARATIVE ANALYSIS

The proposed fully homomorphic encryption system has been implemented in JAVA technology and tested on the computer system with configuration of 2 GB RAM, 200 GB HDD, Intel Dual Core processor and Window-7 operating system. To handle large integers the BigInteger class of JAVA is used. The computation time of proposed scheme and other related schemes has been computed for various key sizes. For testing,

the key sizes for various related schemes and proposed FHE scheme has been taken from 64 bits to 2048 bits. From Table 1, it is clear that when the key size is 512 bits the computation time for proposed scheme is 532 ms which is less in comparison with RSA and Paillier schemes. When the key size is 2048 bits the proposed scheme requires 1027 ms which is very less in comparison with RSA and Paillier.

Table1 Computation time of various schemes

Key Size(bits	RSA	Paillier	FHE(Proposed)
64	3.9876	9.8765	2.2341
128	8.3456	20.9801	4.8732
256	25.8764	48.4521	14.8721
512	89.4563	160.0987	76.4352
1024	557.3456	915.2341	532.4532
2048	2003.8764	4067.0234	1027.8239

Figure 3 describe the computation time of proposed and other related schemes. From figure 3 it is clear that our proposed scheme is efficient in term of computation time

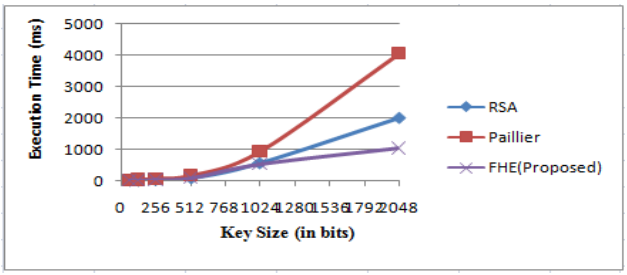

Figure 3. Computation time of various schemes

VII. CONCLUSION

The proposed solution provides the probabilistic full homomorphic encryption capability for integers and rational numbers. The hardness of proposed scheme is based on the factoring problem. By comparing it with existing approaches, we have arrived at the conclusion that our scheme is indeed efficient in terms of time and computation cost and can be implemented in real time.

References

[1]. R. Rivest , A. Shamir and L. Adleman , "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems" , Communications of the ACM , Vol . 21 , No. 2, February 1978.

[2]. S. Goldwasser and S. Micali, Probabilistic Encryption, JCSS Vol. 28 No 2, pp. 270–299, 1984.

[3]. P. Paillier, "Impossibility proofs for RSA signatures in the standard model," in Proceedings of the RSA Conference 2007, Cryptographers' (Track), vol. 4377 of Lecture Notes in Computer Science, pp. 31–48, San Fancisco, Calif, USA, 2007.

[4]. C. Gentry, Fully homomorphic encryption using ideal lattices, Symposium on the Theory of Computing (STOC), 2009, pp. 169-178.

[5]. Gentry, C.: A fully homomorphic encryption scheme. PhD thesis, Stanford University (2009), http://crypto.stanford.edu/craig

[6]. Xiang Guangli, Cui Zhuxiao, "The Algebra Homomorphic Encryption Scheme Based on Fermat's Little Theorem", Communication Systems and Network Technologies (CSNT) 2012 International Conference on, pp. 978-981, 11-13 May 2012.

[7]. Chen, Liang, Zhang Tong, Wen Liu, and Chengmin Gao. "Non-interactive Exponential Homomorphic Encryption Algorithm." In Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2012 International Conference on, pp. 224-227. IEEE, 2012.

[8]. M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, *Fully homomorphic encryption over the integers*, in EUROCRYPT, Springer, Berlin, 2010, pp. 24–43;

[9]. Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. FOCS, 2011.

[10]. T. ElGamal, A Public-Key Cryptosystem a Signature Scheme Based on Discrete Logarithms, IEEE Trans. on Information Theory, IT-31, pp. 469–472, 1985.

[11]. Hrestak, Darko and Stjepan Picek. "Homomorphic encryption in the cloud." *2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (2014): 1400-1404.

[12]. Alattas, Reem. "Cloud Computing Algebraic Homomorphic Encryption Scheme," In International Journal of Innovation and Scientific Research, ISSN 2351-8014 Vol. 8 No. 2 Sep. 2014, pp. 191-195.

[13]. Kangavalli, R. and S. Vagdevi. "A mixed homomorphic encryption scheme for secure data storage in cloud." *2015 IEEE International Advance Computing Conference (IACC)*(2015): 1062-1066.

[14]. Silverberg, A. (2013), 'Fully homomorphic encryption for mathematicians', Women in Numbers 2: Research Directions in Number Theory 606, 111.

[15]. V. Vaikuntanathan. Computing blindfolded: New developments in fully homomorphic encryption. In FOCS, pages 5–16, 2011.

[16]. D. Stehle and R. Steinfeld, Faster fully homomorphic encryption, in ASIACRYPT, Lecture Notes in Comput. Sci. 6477, M. Abe, ed., Springer, Berlin, 2010, pp. 377–394.