IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, August 2023

Performance Evaluation of Tensor Based Light Weight Fully Homomorphic Encryption Scheme

Vinay Kumar Devara[1], Dr. Anshul Mishra[2], Dr. D. Ramesh [3]
Research Scholar Department of Computer Science[1]
Research Supervisor Department of Computer Science and Engineering[2]
NIILM University, Kaithal, Haryana, India[1,2]
Research Co-Supervisor Department of Computer Science[3]
Kakatiya University, Warangal-TG, India[3]

Abstract: *FHE is an outstanding research field on cryptography which facilitates to perform computation on encrypted data. In this research, an effective, simple, secure, and lightweight homomorphic encryption structure is introduced for SHE structure which depends on the integers and utilized matrix. Then employ the coppersmith matrix multiplication method to perform matrix multiplication at the encryption and decryption step to make the scheme more secure. Additionally, a new algorithm is introduced to generate a key and refreshed it for every computation with a stipulated time interval. In the future, this proposed system can be used for asymmetric cryptosystem with two keys, one for encryption, and other for decryption which is more scalable and reliable with high-security measures.*

Keywords: Homomorphic Encryption, Keygen (), Enc () And Dec () Schemes

I. INTRODUCTION

Cloud Computing is an evolving technology that has the potential to preserve and access infrastructures and resources through the Internet with minimum cost. Cloud users utilize IT products provided by cloud for the project development in the specific time, described by Liu [1]. Generally, cloud computing is buying resources like network setup, storage space, servers, applications, tools, and software services from online with high efficiency and minimum cost within a limited period. Today, most IT companies have their cloud server for simple and template-dependent accessed products. The biggest challenge with cloud services is to provide high privacy and security for cloud customers for secure transaction of data. Generally, data encryption was a common method utilized for data security. In the data encryption technique, algorithm and key were utilized to encrypt the data. There are two types of encryption techniques such as symmetric-key cryptography utilizes the same key for encryption and decryption process whereas asymmetric uses two different keys, the public key for encryption of plaintext and the private key for decryption. The cloud server stores the user's data in an encrypted format called ciphertext. Operations like XOR, multiplication, addition, sorting, and searching could perform on the ciphertext. The client decrypts the cipher text which is preserved in the cloud server with a key or the client might provide a key to CSP for performing computations on the server. But clients won't believe CSP for the performing computations/updating of sensitive data by the CSP.

Hamlen et al. [2] suggest the confidential purpose, data on the server-side is decrypted by the client itself to acquire the plaintext to modify, and then updated data is encrypted to develop cipher text to save in the cloud server. In this process of encryption and decryption, few security defects were noticed like frequent computation and decryption may create change for the attacker to exploit data authentication and integrity. To mitigate twice encryption as above cited and exploit eves over the network and provide security to perform outsourced computation, uses the new kind of enhanced encryption process called a Homomorphic Encryption.

1.1. HOMOMORPHIC ENCRYPTION

It is a kind of encryption process which do computation on previously encrypted data without any decryption process and matched the result of performed computation on the plaintext. HE is the most popular technique in cloud computing especially for performing client-server communication with numerous navigations. In the cloud, customers

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, August 2023

could preserve their confidential data but for performing computational operations like sorting, searching and updating, customer needs to decrypt the ciphertext and do modification and then forwarded to cloud server in encrypted form.

Rivest et al. [3] at 1978 introduced privacy homomorphism which performs computation on the ciphertext itself without the decryption process. For data updating and decryption, there is a need for customers to depend on CSP described by Nearing et al. [4]. Despite the cloud, users are dependent on CSP for the distribution of the key. HE offers a platform to the client not requested to share keys to CSP for accessing and storing data on the cloud. HE is the process of conversion of data to cipher text which was examined and worked as if remains in original form.

Homomorphic Encryption facilitates clients to do a complex mathematical operation on the encrypted data without decryption. In mathematics, homomorphic means transformation from one data set to another dataset by preserving association among elements in both datasets. The term "Homomorphic" was derived from the "same structure" which is a Greek word. This is because data of homomorphic encryption retains identical mathematical operations and the same structure on encrypted and decrypted data. HE plays a key role in cloud computing by facilitating cloud customers to preserve encrypted data over the public cloud and also utilize the benefits of CSP's analytic services proposed by Bajpai et al. [5].

Homomorphic Encryption is an arbitrary computation done on encrypted data on the server-side without the decryption process. But data decrypted with a private key by the client matches the same work done on the plaintext. The following equation expresses the basic definition of homomorphic encryption. Any kinds of operations such as XOR, multiplication, addition, filtering, sorting and searching can be done on ciphertext on the server-side without the private key. With Homomorphic Encryption operation done on ciphertext will be the same as operation done on the plaintext.

Operation (plaintext) = decrypt (operation) (encrypt) (plaintext) ……..(3.1)

Figure 1 illustrates that there isa need for the client to search for data "ABC" over the cloud server and encrypted data is referred to as a query that is forwarded to the cloud server to search "ABC" string over the cloud server. HE searches strings related to client requests and once there quest is determined then it sends backs to the client in the form of encrypted data.

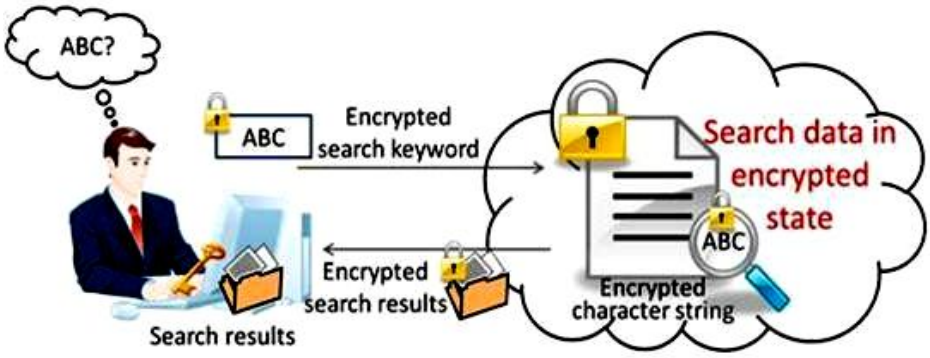

Figure 1: Concept of Homomorphic Encryption

The client uses the private key to decrypt the encrypted data. In cryptography, HE does the same process directly on cipher text without doing the decryption and forwards back to the client in the form of encrypted data. But both processes remain the same. Let ciphertext is denoted as C. When the key satisfies the function F, the encryption scheme had homomorphic as surveyed by Armknecht et al. [6].

$$c_1, c_2 \in C, F(c_1 \circ c_2) = F(c_1)\ ^{o}F(c_2) \text{ ……. (3.2)}$$

Generally, the homomorphic cryptosystem can be used with the additional property which is utilized to calculate the encryption of product or sum for two messages with the public key. Table 1 illustrates several HE operations with applications.

Table 1: Homomorphic Encryption operations with Applications

Crypto System	Additive	Multiplicative	Mixed	XOR	Applications
Paillier	✓	×	×		e-voting system, threshold scheme

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, August 2023

RSA	×	✓	×		To secure the Internet, Banking and credit card transaction
El Gamal	×	✓	×		In Hybrid systems
Gentry	×	×	✓		Multiparty computation
BGV	×	×	✓		Forth security of integer polynomials
Extended Homomorphic Cryptosystem	×	×	✓		Secure transmissional message in MANET
Nissim, Goh, and Bone	✓	✓	✓		Stack Exchange, One multiplication and many additions
Goldwasser-Miceli	×	×	×	✓	Biometric authentication

The benefits and applications of the HE scheme don't require CSP. In the medical field, analyses and treatment of disease are carried out without revealing confidential data and search for DNA markers without disclosing DNA, third parties, and spam filtering- Blacklisting encrypted mail scan PGP traffic and implement electrocoating system easily. The Homomorphic encryption schemes are characterized into three types such as,

Partially Homomorphic Encryption (PHE)

Some what Homomorphic Encryption (SHE)

Fully Homomorphic Encryption (FHE)

II. SYMMETRIC KEY FHE SCHEMES

The proposed method uses a secure new symmetric key based somewhat homomorphic encryption scheme. This scheme used the integers as plaintext that is in the form of matrices and these matrices are lightweight. In this scheme, some computational matrix principle like orthogonality is used to reduce the computational complexity while performing generation of secure secret key at key generation step and adopted Coppersmith et al. [7] matrix multiplication, in this method, reduces the matrix exponent by O ($n^{2.376}$). Using this method, gained better computational time of performing matrix multiplications at encryption and decryption steps and also employed the Chinese Remainder Theorem for finding the best congruent integer element of the matrices. The performance of the proposed scheme proved the homomorphic additive and multiplicative properties. Furthermore, compared the performance results with existing state-of-the-art methods and gained better computational time.

Design of Symmetric Key FHE Schemes

Here, introduced a fully homomorphic encryption system depending on the symmetric key. The proposed structure contains sub-modules such as:

Generation of key

Procedure for encryption

Procedure for decryption

Procedure for akey refresh

Steps and description of FHE structure are explained clearly in the below section:

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, August 2023

Algorithm for Key Generation

Algorithm 1 Key Gen (L,b)

Choose randomly 2b pairs of odd number such as yj and xj where $1 \leq j \leq b$ are chosen numbers. These numbers must be pairwise co- prime with each other, consist of L-bit size.

SelectorthogonalmatrixKwithdimension4inZR.Thebelow procedure is followed to select K.

The randomly elect matrix K in the ZR search space. Verify whether it's able to follow the orthogonal property that is K. KT=KT.K=I or

K−1=KT, then the search is finished, else repeat till to find the or thoronol matrix.

Find transpose of matrix K

KT transpose(K)←mod ZR

Note: - From the property of or thoronol matrices,K−1=KT

K orthogonal matrix would act as a symmetric key in the proposed cryptosystem.

Algorithm for Encryption

Algorithm 2: - Enc_ Procedure (N, mj, R, K, KT)

Take Nas plaintext.

Select random integer S, where S∈Z R and S≠N.

Develop matrix X with dimension (b ×3), where every row contains one occurrence of N and the remaining 2 occurrences of S.

Implement CRT described by Guo et al., [9] to acquire a solution for simultaneous equations

*a*i (1≤i≤3)Xjimodmjwhere1≤i≤b.

Utilize Coppersmith et al., [7] algorithmic arithmetic procedure for computation of matrix multiplication which is as follows.

Receive cipher text C = KT ×d (N, a^1, a^2, a^3) × K where (N, a^1, a^2, a^3) represents a diagonal matrix with a parameter as a diagonal element.

Algorithm for Decryption

Algorithm 3: Dec_ Procedure(C,R,K,KT)

1. Find Plain text, N = K×C×KT

2. $N=\left[K\right]_{n\times n}\times\left[C\right]_{n\times n}\times\left[K^T\right]_{n\times n}$

3. $N=\begin{bmatrix} N_{11} & N_{12} & N_{13} & N_{14} \\ N_{21} & N_{22} & N_{23} & N_{24} \\ & ..... & & \\ N_{n1} & N_{n2} & N_{n3} & N_{n4} \end{bmatrix}_{n\times n}$

4. $N \leftarrow [N_{11}]$

Algorithm for Key Refresh Procedure

Algorithm 4: Refresh Procedure (R)

The symmetric key is refreshed, an or thogonal matrix is defined as k_r and morpho (t)where grandmother (t) is a random function that generates new key K with dimension t randomly in $S_p$ace. The function rancorous (t)misrepresented$_k=(aX_{k-1}+c)\bmod m$

Here X_k is anew key,(k−1)is the previous key,(c,m)are constants, X0 is the primary key that is any random integer between(2,m)and "a" is a random number in <m space.

Constraints:

Both (c,m)must be co-primes an dare in Z_R space.

(a−1) should be divide able by all prime factor so fame, and

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, August 2023

Newly generated key series should be<m within queness.

Properties of Symmetric Key FHE Scheme

This section describes the verification of the correctness condition of the decryption step and satisfies the correctness of both multiplicative and additive homomorphic properties.

Correctness of Decryption Scheme

Analysis and observation of the proposed structure are examined below to proves the correctness of the decryption algorithm, it's noticed that,

Dec_ Procedure(C,R,K,K^T)K×C×K^T $\Longrightarrow$

Since the researcher knows about the or thoronol matrix's property as, K. KT=KT.K= I or $K^{-1}=K^T$

So Dec_ Procedure (C,R,K,K^T) K×K^T×d (N,a_1, a_2 ,a_3)×K×K^T

I×d (N, a_1,a_2,a_3)×I

d(N, a_1 ,a_2, a_3)

[N11] $\Rightarrow$

N $\Rightarrow$

III. PERFORMANCE EVALUATION OF KEY GEN (), ENC () AND DEC () SCHEMES

The proposed efficient symmetric key based homomorphic scheme is a lightweight structure that uses lightweight matrix computational operations when compared with polynomial computations. This proposed FHE structure is parallelizable. In this scheme, encryption and decryption provide benefits of doing outer product matrix-vector multiplication which is much parallelizable.

To improve computational complexity, Coppersmith et al. [7] an algorithm is used in encryption and decryption steps for the computation of matrix multiplication. This technique enhances the computational complexity of matrix multiplication as O ($m^{2.376}$).

Somewhat Homomorphic Encryption

The following subsection represents the performance and comparative analysis of the proposed scheme with state-of-the-art approaches. This section described the experimental results of the proposed scheme by using the following setup. The minimal required cloud server is OpenStack, operating system Ubuntu 16.04 version, RAM size 4GB, the clock speed of CPU is 2.8GHz, core i5 processor, and python as programming language version 3.6.

The performance of the Somewhat Homomorphic Encryption scheme depend son time consider to gene rate a key estimation time for the degree of polynomial, decryption, and encryption-related to plaintext size.

Table 2: Performance Evaluation of Key Gen (), Enc () and Dec ()

Size of R in bits	Time taken for Key Gen (in sec)	Time taken For Enc (in sec)	Time taken for Dec (in sec)
16-bits	0.332	0.173	0.096
32-bits	28.41	55.20	15.0
50-bits	59.85	71.19	22.40
64-bits	118.42	204.91	102.45

Table 2 illustrates the performance of SHE based on the data size, key generation time, encryption time, and decryption time. The data size is taken from 16 to 64 bits. Figure 2 represents the execution time for various data sizes of homomorphic encryption. The x-axis denotes the size of Rin bits of plain text, the execution time for a key generation, encryption and decryption denote data they-axis.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, August 2023

.

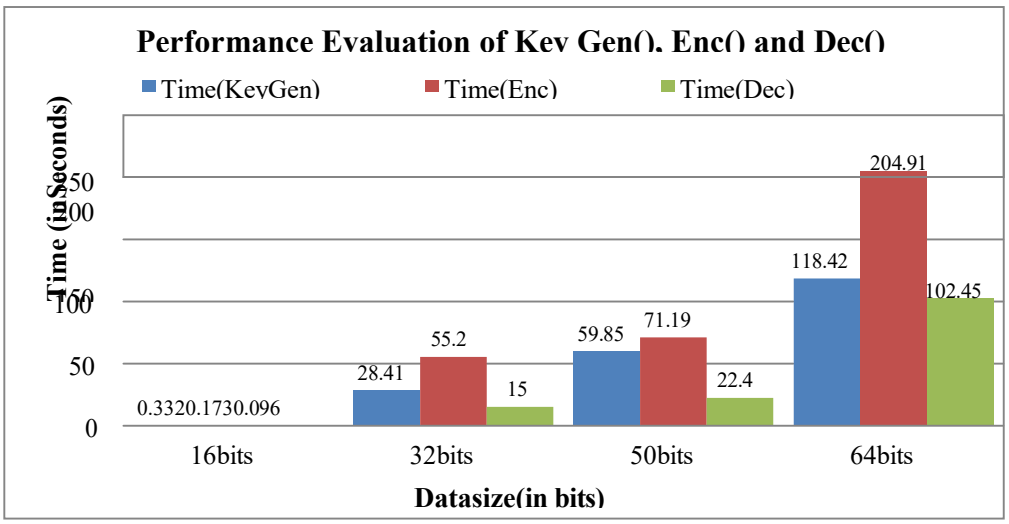

Figure 2: Performance Evaluation of Key Gen (), Enc () and Dec ()

Comparison of Key Gen (), Enc (), Dec () and Re-crypt Schemes

In this section, a comparison of proposed schemes with various fields like the dimension of plaintext size, Time is taken for Key Gen (), various sizes, encryption, decryption, and re-cryption functions with existing schemes.

Table 3: Comparison of Key Gen (), Enc (), Dec () and Re-crypt

Item of Comparison	DHGV SHE	CMNT SHE	Symmetric key-based SHE Scheme
Dimension (Bits)	2048	8192	33024
Keygen (Time)	40sec	8min	7min
Key size (MB)	70	285	375
Enc () (sec)	0.6	0.7	6.5
Dec () (sec)	0.23	0.12	0.8
Recrypt	31sec	3min	20min

The proposed scheme is outperformed in terms of key generation, reducing the key size, less in the time occupied for encryption and decryption operations. In the proposed scheme, adopted the bootstrapping step designed by Craig Gentry to refresh the ciphertext after certain bound computation i.e. when decryption gets wrong computation results. The bootstrapping step can refresh ciphertext by homomorphically decrypting using an encrypted secret key.

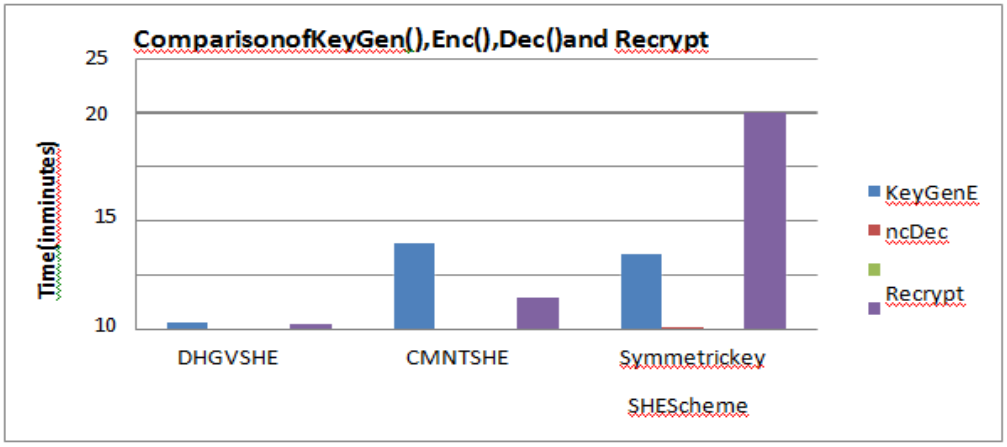

Figure 3: Comparison of Key Gen (), Enc (), Dec () and Re-crypt

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, August 2023

By using Gentry's bootstrapping process, gained the best results compared with existing homomorphic schemes of recrypt (bootstrapping) operation by 20 minutes. It makes the proposed scheme more accurate and noise-free arbitrary computations over cip her text after bound. Table 4 and figure 3 denotes a comparison of current methods with the proposed SHE method based on the security, encryption time, key generation time, decryption time, dimension, and less compactness. This analysis proved that the proposed method had high efficiencies compared to other methods.

Comparison of Fully Homomorphic Properties

The performance of an efficient symmetric key-based scheme's time taken in milliseconds for one homomorphic addition and one homomorphic multiplication operations are 1.9 and 42 milliseconds of maximum data size is 65793 bits are illustrated in table 4. (Time in milliseconds).

Table 4: Comparison of Fully Homomorphic Properties

Item of Comparison	DHGV SHE	CMNT SHE	Symmetric key-based SHE Scheme
Modulus(bits)	257	8209	65793
Homomorphic addition (ms)	0.7	0.7	1.9
Homomorphic multiplication(ms)	39	38	42

The comparison of the fully homomorphic properties of the proposed scheme with existing schemes is shown in figure 4. The proposed method achieves a better time for homomorphic addition and multiplication even though the data modulus size is a maximum of 65793 bits that is 8.01 times higher than the existing CMNT scheme modulus size. Further research directions are to implement the lightweight fully homomorphic encryption scheme by the continuation of this objective and proposes a tensor power-based optimization technique to reduce the computational complexities involved in the FHE scheme.

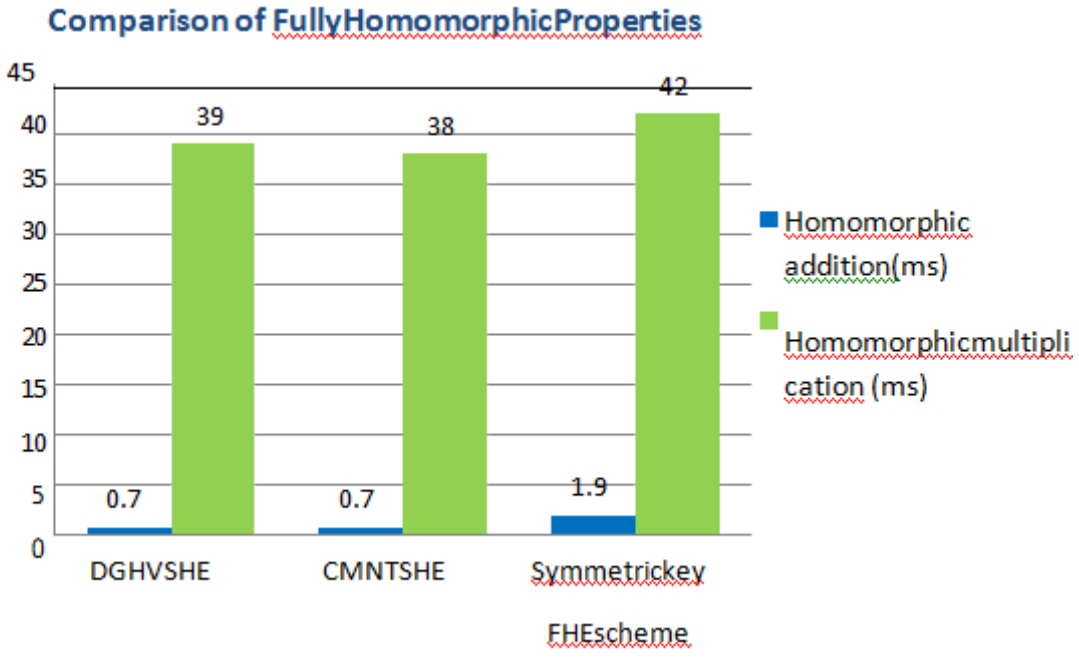

Figure 4: Comparisons of Fully Homomorphic Properties

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, August 2023

IV. CONCLUSION

The symmetric based Fully Homomorphic Encryption (FHE) algorithm with the steps of Key generation, Encryption procedure, Decryption procedure, and key refreshing procedure. Then verify the proposed method's efficiencies by using additive Homomorphic and multiplicative Homomorphic computations. In the end, the experimental result of the proposed method was measured that proved their high performances and compared with existing methods.

REFERENCES

[1]. W. Liu, "Research on cloud computing security problem and strategy "in 2nd IEEE International Conferenceon Consumer Electronics, Communications and Networks (CECNet), pp.1216-1219, 2012.

[2]. K. Hamlen, M. Kantarcioglu, L. Khan, and B. Thiazinam, "Security issues for cloud computing" International Journal of Information Security and Privacy, vol.4, pp. 36-48, 2010.

[3]. R. L. Rivest, L. Adleman, and M. L. Dertouzos, "On data banks and privacy homomorphisms" Foundations of secure computation, vol. 4, no.11, pp. 169-180, 1978.

[4]. M. Naehrig, K. Lauter, and V. Vaikuntana than, "Can homomorphic encryption be practical?" in Proceedings of the3rd ACM workshop on Cloud computing security workshop, pp. 113-124,2011.

[5]. B. L. Chen, W. C. Kuo, and L. C. Wuu, "Robust smart-card- based remote user password authentication scheme" International Journal of Communication Systems, vol. 27, no.2,pp.377-389,2014.

[6]. J. Li and L. Wang, "Noise-free Symmetric Fully Homomorphic Encryption based on noncommutative rings" The International Association for Cryptologic Research, pp. 641-642, 2015.

[7]. Don Coppersmith and Shmuel Winograd "Matrix multiplication via arithmetic progressions" Journal of Symbolic Computation, vol. 9, no.3, pp.251-280, 1990.

[8]. Z. Chen, J. Wang, Z. Zhang, and X. Song, "A fully homomorphic encryption scheme with better key size" China Communications, vol. 11, pp. 82-92, 2014.