ADVANCED TECHNIQUE OF IMPROVING HOMOMORPHIC ENCRYPTION SCHEME IMPLEMENTATION INTO CLOUD COMPUTING

[1] A.Jeeva Rathinam, [2] Dr.E.Ramaraj
[1] Mphil Research Scholar, [2] Professor and Head
[1] Department of Computer Aplications, [2] Department of Computer Science,
Alagappa University, Karaikudi, Tamilnadu, India

Abstract : Cloud computing is a developing technology that is yet unclear to many security issues. Data in the hopeless clouds can be encrypted using all encryption algorithms. Simultaneously this data provides more advanced security which can be achieved by stuffing technique in the cloud. Homomorphism computing has been suggested as a method to secure processing in insecure servers. One of the drawbacks of homomorphic processing is the enormous execution time taken to process even the simplest of operations. Some of the disadvantages is that they are directly related to data redundancy, Privacy and more Noisy Level, and hence, decreasing the security level. The Improving Homomorphic Encryption (HE) is achieved on the encrypted data without decrypting it in computationally powerful clouds and the Secure Multi-Level Computation (SMLC) can be used in the cloud to ensure security and privacy of the users. Results are provided in terms of computational complexity and privacy. This allows evaluating a scheme for numerous sets of input parameters. In this paper, we have proposed a scheme that integrates the multi-level computation with homomorphic encryption to allow calculations of encrypted data without decryption. The Advanced cryptography techniques used in research cloud model are illustrated and the upwards are compared with Homomorphic Encryption and Multi-Level Computation.

IndexTerms **- Homomorphic Encryption; Cloud Computing, Multiparty Computation**

I. INTRODUCTION

There is a need for an appropriate or more suitable big data infrastructure that supports the storage and processing on a high scale. Now a days the world is data centric, hence the big data processing and analysis have become the most important chore for any large establishment. Fully homomorphic encryption (FHE) allows computation to It has thus attracted attention for cloud computing applications. One of the main challenges is to relax the requirement of extremely high computational resources. For example, as shown in Gentry's work with various parameters of interest, the bit-length of a cipher text encrypted from a one-bit message is more than one hundred thousand bits. It also takes a large amount of time to process a homomorphic evaluation, which consists of homomorphic multiplication and homomorphic addition.

In the field of computation on encrypted data ideally one would like to perform both, additively and multiplicatively arithmetic operations on encrypted data in its most flexible way. This is what FHE achieves. However, for most application fields FHE comes with non-acceptable performance degradation, both with respect to runtime and with respect to cipher text versus clear text data size.

One may argue that the most recently proposed Leveled Fully Homomorphic Encryption (LFHE) scheme which is based on the building blocks of the SHE scheme may be a more advanced choice. However, we argue that for performance reasons this is surely not the case. Additional costs for the LFHE extension of the SHE mainly arise due to a freshness function which is applied subsequently to each addition operation and multiplication operation on cipher text polynomials with the two objectives to weave in i) a fresh key as well as ii) a new modulus. Whereas to our understanding the first aims at what we coined "perfect forward computation" for computation on encrypted data (e.g. in similarity to "perfect forward secrecy" for communication protocols), a new modulus per computation level aims to reach "fully" homomorphic in particular for the limited operation in SHE, namely the multiplication.

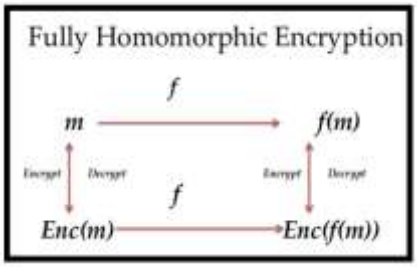

Fig.1 Fully Homomorphic Encryption diagram

To solve the problem of security, modern algorithms for symmetric and asymmetric encryption are not suitable, because they do not allow the security of information processed in the clouds. Homomorphic encryption allows confidentiality of the information and results of the computations. In the implementation of computational algorithms using homomorphic ciphers, there is a problem associated with the conversion of the algorithm into the set of operations supported by a homomorphic cipher. Efficient use and implementation of homomorphic ciphers become important.

Homomorphic encryption has the following applications:

- Cloud computing
- Electronic voting
- Secure information search
- Secure wireless decentralized communication networks
- Outsourcing services for smart cards
- Feedback systems
- Obfuscation for software security

A cryptosystem that supports both addition and multiplication is called fully homomorphic. Fully homomorphic cryptographic systems allow processing of data in encrypted form. The homomorphic property of various cryptographic systems can be used to create secure voting systems, hash functions that are resistant to collisions, private information of search engines and makes possible widespread use of public cloud computing, ensuring the confidentiality of processed data. When using homomorphic encryption and Multi-Level Computations to ensure the security of information in the cloud computing, the probability of distortion of one or more chunks of the result is high.

II. RELATED WORK AND LITERATURE SURVEY

C.Gentry [6] computed arbitrary functions of encrypted data which describes a fully homomorphic encryption technique that keeps information private, but that leaves a worker that does not possess the private decryption key to compute any result of the data, even when the purpose of the data is really complex.

M. Tebba et al. [2], projected a technique to implement operations on encrypted data in the cloud which will supply us with the comparable results following calculations as if we have processed straight on the raw data.

C. Rong et al. [5] handled a audit on different security objection in Cloud Computing.

Yu et al. checked the active attacker attack in three examining techniques for distributed data in the cloud and also expected a solution to antidote the deficiency without sacrificing any enticing features of these techniques.

L. Wei et al. [9] expected a privacy cheating hopelessness and secure calculating examining design, or SecCloud, which is a first design traversing protected storage and secure computation examining in cloud.

A. Partially Homomorphic Encryption

In PHE only some operations can be performed on the encrypted data. For example addition and multiplication are the two given operations, and then any one of them can be performed on the encrypted data but not both. PHE schemes are useful for certain application which requires specific computations. Helios Voting scheme [8] make use of PHE. By this approach the encrypted voted ballots can be publicly stored in the cloud and the public can count the votes as well as verify their votes for each candidate without producing any proofs to the third party.

B.Somewhat Homomorphic Encryption

In SHE scheme more than one operation can be performed on the encrypted data. But the limitation of SHE is that no all operations apply to all types of data. Even though it can support multiple operations, but only in limited numbers. SHE is suitable for a variety of real time applications such as financial, medical and recommender systems. Since SHE supports a limited number of operations it will be much faster than fully homomorphic schemes.

C. Fully Homomorphic Encryption

FHE scheme supports any number of operations on any encrypted data. The circuit designed for FHE is homomorphically evaluated. This will be suitable for any sort of application working with encrypted data. Compared to PHE and SHE, FHE is little bit less efficient due to the computational overhead. As of today, for a particular application PHE and SHE is showing better efficiency compared to FHE scheme.

D.Additive Homomorphic Encryption

Additive Homomorphism deals with addition of encrypted data. Let us consider two data a and b¬ and is encrypted to Enc (a) and Enc(b).On applying additive homomorphic encryption, we get as given in (1),

$$Enc\ (a) + Enc\ (b) = Enc\ (a + b) \qquad -(1)$$

According to additive homomorphism, the sum of the encrypted messages is equivalent to the encrypted sum of those messages.

E.Multiplicative Homomorphic Encryption

Multiplicative homomorphism talks about multiplication of the encrypted data. Let us assume two data a and b and is encrypted to Enc(a)and Enc(b).While applying multiplicative homomorphic encryption leads to (2)

$$Enc\ (a) \times Enc\ (b) = Enc\ (a \times b) \qquad -(2)$$

F.Algebraic Homomorphic Encryption

If f is a function from A->B and k E Kf and a, b E Af then the equations (3), (4), (5) form Algebraic Homomorphism.

$f(k \times a) = k \times f(a)$ - (3)

$f(a + b) = f(a) + f(b)$ - (4)

$f(a \times b) = f(a) \times f(b)$ - (5)

According to algebraic homomorphy developing algebraic homomorphic encryption scheme is really a challenging one. Lots of proposals are evolved, but none of them given a satisfactory solution.

III. PROBLEM FORMULATION FOR SECURE CLOUD COMPUTING

A. Availability

In this scheme cloud service providers have multiple servers. When one server fails, there is no security issue as another server is ready to provide services.

B. Integrity

The data integrity means the correctness and trustworthiness of the data. It ensures that the computation on sensitive data is correct. The data should not be modified by the illegal user.

C. Confidentiality

Confidentiality is to avert receptive information from the reach of the assailant, while creating sure that the approved users have access to it. Services require user to trust the cloud with their data. But in the untrusted cloud Data owners do not trust the cloud. Thus user side protection is necessary. Users encrypt their data before storing into the cloud with the help of a public key.

D. Cycle attack

In this attack, the cipher text is encrypted repeatedly and the no of iterations are counted until the original text appears. It can decrypt any cipher text.

E. Cipher text attack

In this attack, both the plaintext and the cipher text is known to the attacker and he can use this to discover private exponent and once it discovered it is easy to find then. Multiple parties wish to perform operations on their inputs. This requires decryption of their data. This poses security problems in the case of untrusted Clouds.

1. OBJECTIVES

The Cloud environment requires protection and confidentiality of user data while leveraging computation ability of entities in the cloud network directly on encrypted data. This paper focuses on an issue that is attractive to many types of research, which is a data encryption for cloud computing. Cloud environment requires security and confidentiality of user data while leveraging the computational ability of entities in the cloud network directly on encrypted data. In this paper, we have proposed a scheme that integrates the multiparty computation with homomorphic encryption to allow calculations of encrypted data without decryption.

IV. OUR CONTRIBUTION

In our paper, we have expected an efficient cryptographic technique by padding the multiple party data before encrypting it. The user's data is encrypted using padding scheme Optimal Asymmetric Encryption Padding (OAEP) together with Hybrid Encryption algorithm that is based on RSA Small and Efficient RSA (HE-RSA). In order to allow multiple parties to compute a function on their inputs while preserving Integrity and Confidentiality. The proposed scheme integrates the multi-Level computation with homomorphic encryption to allow calculations of encrypted data without decryption. The output of this nature allows maintaining confidentiality and integrity in the cloud environment.

Key Generation:

1: Select two large prime numbers p and q such that n = p × q

2. Calculate $\emptyset(n) = (p-1)(q-1)$

3. Select a random key e, where $1 < e < \emptyset(n)$ and gcd $(e; \emptyset(n)) = 1$

4. Calculate the private key d such that $d = e^{-1} mod\ \emptyset(n)$

The public key=[$e; n$]

The private key=[$d; n$]

Encryption:

Encryption of message m using public key produces the cipher text c

$c = m^e \ mod\ n$

Decryption:

Decrypting the cipher text c using the private key gives the plaintext m

$m = c^d \ mod\ n$

Homomorphic Multiplication:

$c_1 = m1^e \ mod\ n$

$c_2 = m2^e \bmod n$

$c_1 \times c_2 = m1^e \bmod n \times m2^e \bmod n == (m1 \times m2)^e \bmod n$

I.e. $Enc(m_1) \times Enc(m_2) = Enc(m_1 \times m_2)$

V.PROPOSED ALGORITHM FOR SECURE CLOUD COMPUTING

The proposed secure cloud computing algorithm is ensuring the security and privacy of individual data in the cloud along with the enhancement of the security mechanism like Homomorphic Encryption and Multi Level Computation (MLC). The Proposed Algorithm is based along the four phases: Key Generation, Encryption, Homomorphic Encryption (HE) and Multi Level Computation (MLC), and Deception. The main goal is to minimize the running time, cost, and the overhead during these four phases. In the proposed Algorithm, the number of exponents during key generation (in the step 1) has been enlarged in comparison to the some existing Algorithms (i.e., RSA). In addition to that, a dual encryption process (in the step 2) has been implemented in this Algorithm to prevent the general attacks against some existing techniques. In step 3, we have integrated the fully homomorphic encryption and multi-level computation to allow the calculations of encrypted data without decryption in the cloud.

Algorithm 1 Algorithm for Secure Cloud Computing

Step 1: Key generation algorithm: keygen(p,q) Randomly choose two large primes p ,q and compute n=p.q

$\emptyset(n) = (p - 1)(q - 1)$

$Y(n, h) = (p_h - p_0)(p_h - p_1) \ldots\ldots\ldots\ldots\ldots (p_h - p_{h-1})(q_h - q_0)(q_h - q_1) \ldots\ldots\ldots (q_h - q_{h-1})$

Select random integer r such that $1 < r < n$ and

$gcd(R, \Phi) = 1$ and $gcd(R, Y) = 1$

Compute e such that $r.e = 1 \bmod \Phi$, $1 < e < \Phi(n)$

Compute d such that $d.e = 1 \bmod Y$, $1 < d < Y(n)$

Public key(pk): (e,n)

Secret key(sk): (r,d,n)

Step 2: : Encryption: Enc(M,pk)

Suppose Sender and Receiver send data to M1 and M2

respectively to the cloud

$G : \{0, 1\}_{K0} \dashrightarrow \{0, 1\}_{K0}$

$H : \{0, 1\}_{K-K0} \dashrightarrow \{0, 1\}_{K-K0}$

$r \longleftarrow \{0, 1\}_{K0}$

$S = (M \ 0_{K1}) \oplus G(r)$

$t = r \oplus H(s)$

$C \longleftarrow S_e \bmod n_e \bmod n$

Return C

Step 3: Homomorphism and Multi-level computation Homomorphic computations are performed on Sender and

Receiver encrypted data C1 and C2 respectively.

$C1 = ((M1)_e \bmod n)_e \bmod n$

$C2 = ((M2)_e \bmod n)_e \bmod n$

$C1.C2 = [((M1)_e \bmod n)_e \bmod n][((M2)_e mod \ n)_e \bmod n]$

$= ((M1)_e \bmod n)_e((M2)_e \bmod n)_e \bmod n$

$= ((M1M2)_e \bmod n)_e \bmod n$

$Let\ C = C1.C2,\ M = M1M2$

$C = (M_e \bmod n)_e \bmod n$

Step 4: Decryption: Dec(C,sk)

Sender and Receiver decrypt the computed data C using their respective private keys

$W \longleftarrow (C_r \bmod n)_d \bmod n$

Parse W as *st*

$r \leftarrow H(s) \oplus t$
$M_1 \leftarrow s \oplus G(r)$, *parse* M_1 *as* MZ

VI.RESULTS AND DISCUSSION

After combining Fully Homomorphic encryption and Multi Level Computation (FHE + MLC), the confidentiality and integrity of the data is maintained and the overhead is less than Fully Homomorphic Encryption but more than Multi Party Computation. So, we have received moderate overhead based on the Fully Homomorphic Encryption and Multi Party Computation (Shown in Table I). Fig. 2 summarizes the approximate efficiency, and cost of each of the techniques across a wide range of computations, depicting the multiplicative performance overhead incurred over unsecured computation. In addition to its inefficiency, homomorphic encryption has other limitations. Table II and Fig. 3 were comparison of existing and proposed approach.

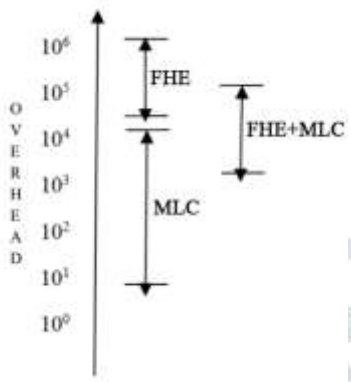

Fig. 2. A graphical depiction of the multiplicative performance overheads over unsecured computation incurred by Fully Homomorphic Encryption (FHE), Multi Level Computation (MLC) and Fully Homomorphic Encryption + Multi Level Computation (FHE + MLC)

Table 1 Comparison of Cryptographic Approach

Cryptographic Technique	Confidentiality	Integrity	Interaction	Overheads
FHE	YES	NO	NO	More Overheads
MLC	YES	YES	YES	Less Overheads
FHE+MLC	YES	YES	YES	Moderate Overheads

Table 2 Comparison of Existing and Proposed System Approach

	Existing System (FHE)	Proposed System (FHE+MLC)
Confidentiality	85%	88%
Integrity	91.7%	95.3%

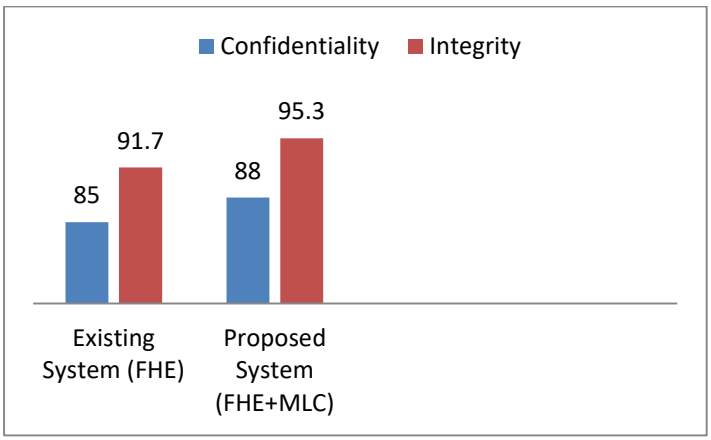

Fig. 3 Comparison of Existing and Proposed System

VII.CONCLUSION

In this paper, proposed a secure cloud computing model in which efficient cryptographic technique Based on Homomorphic Encryption (HE) And Multi-Level Computation(MLC) was used to encrypt user's data followed by operations on their data while maintaining integrity and confidentiality. The output is same as if the operations have been carried on raw data. A level is able to jointly perform computations without revealing their data to the other party. Here, designed and developed secure homomorphic encryption and multi-party computation techniques tailored specifically for a private semi-trusted cloud setting. This setting allows developers to design the private cloud together with the cryptographic techniques (i.e., HE+MLC) necessary to protect it.

FUTURE WORK

Future work on compromised of increasing the efficiency of our proposed scheme so that the computational time will be reduced.

REFERENCES

[1] Mell P, Grace T. The NIST definition of cloud computing, NIST Special Publication, 2009, pp. 800–145.

[2] Tebaa M, Hajji S.E, Ghazi A.E. Homomorphic Encryption Applied to the Cloud Computing Security, Proceedings of the World Congress on Engineering, London, U.K., Vol.1, No.1, 2014, pp. 4-6.

[3] Wang Z, Sun G, Chen D. A new definition of homomorphic signature for identity management in mobile cloud computing, Journal of Computer and System Sciences, Vol. 80, N0. 3, 2014, pp. 546-553.

[4] Yakoubov S, Gadepally V, Schear N, Shen E, Yerukhimovich A. A Survey of Cryptographic Approaches to Securing Big-Data Analytics in the Cloud, IEEE High Performance Extreme Computing Conference (HPEC), 2014, pp. 1–6.

[5] Rong C , Nguyen ST, Jaatun MG. Beyond lightning: A survey on security challenges in cloud computing, Computers and Electrical Engineering, Vol. 39, No. 1, 2013, pp. 47-54.

[6] Gentry C.Computing Arbitrary Functions of Encrypted Data, Communications of the ACM, Vol. 53, No. 3, 2010, pp. 97-105.

[7] Wang C, Wang Q, Ren K, Lou W. Ensuring Data Storage Security in Cloud Computing, Quality of Service, 2009, pp. 1–9.

[8] Yu Y, Niua L, Yang, G, Mu Y, Susilo W. On the security of auditing mechanisms for secure cloud storage, Future Generation Computer Systems, Vol. 30, 2014 pp. 127-132.

[9] Wei L, Zhu H, Cao Z, Dong X, Jia W, Chen Y, Vasilakos AV. Security and privacy for storage and computation in cloud computing, Information Sciences, Vol. 258, 2014, pp. 371-386.

[10] Lopez-Alt A, Tromer V, Vaikuntanathan E. On-the-Fly Multiparty Computation on the Cloud via Multikey Fully Homomorphic Encryption, Proceedings of the forty-fourth annual ACM symposium on Theory of computing, 2012, pp. **[11]** Brakerski Z,and Vaikuntanathan E. Efficient fully homomorphic encryption from (standard) LWE, SIAM Journal on Computing, Vol.43, No.2, 2011, pp. 831–871.

[12] Bellare M, and Rogawayy P. Optimal Asymmetric Encryption How to Encrypt with RSA, Advances in Cryptology Eurocrypt 94 Proceedings, Vol. 950, 1995, pp. 1–19.

[13] Shen E, Varia M, Cunningham RK, Vesey WK. Cryptographically Secure Computation, IEEE Computer Society, Vol. 48.

[14] Zissis D, Lekkas D. Addressing cloud computing security issues, Future Generation Computer Systems, Vol. 28, No. 3, 2012

[15] Hashem IAT, Yaqoob I, Anuar NB, Mokhtar N, Gani A, Khan S.U. The rise of 'big data' on cloud computing: Review and open research issues, Information Systems, 2015, Vol. 47, pp. 98-115.

[16] Zhou J, Dong X, Cao Z, Vasilakos AV. Secure and Privacy Preserving Protocol for Cloud-based Vehicular DTNs, IEEE Transactions on Information Forensics and Security,Information Systems,Vo. 10, No. 6, 2015, pp. 1299 - 1314.

[17] Zhao J, Wang L, Tao J, Chen J, Sun W, Ranjan R, Kolodziej J, Streit A, Georgakopoulos D. A security framework in G-Hadoop for big data computing across distributed Cloud data centres, Journal of Computer and System Sciences, Vol. 80, No. 5,

[18] Zuech R, Khoshgoftaar TM, Wald R. Intrusion detection and Big Heterogeneous Data: a Survey, Journal of Big Data, Springer, Vol. 2, No. 3, 2015, pp. 1-40.

[19] Hongbing C, Chunming R, Kai H, Weihong W, Yanyan L. Secure Big Data Storage and Sharing Scheme for Cloud Tenants, China Communications, 2015, pp. 106–115.