Volume 8, No. 3, March – April 2017
ISSN No. 0976-5697
International Journal of Advanced Research in Computer Science
RESEARCH PAPER
Available Online at www.ijarcs.info

FHE Implementation of Data in Cloud Computing

Dr. Mohammad Miyan
Associate Professor, Shia P. G. College, University of Lucknow
Sitapur Road, Lucknow, India

Abstract: The Cloud computing is ever-growing field in the present scenario. With the build up of knowledge and also the advancement of technology, a huge quantity of knowledge is generated every day. Storage, accessibility and security of the data type major issues rise within the field of cloud computing. The present paper focuses on homomorphic encryption that is basically used for security of knowledge within the cloud. Homomorphic encryption is outlined because the technique of encryption within which specific operations may be administered on the encrypted data information. The data is kept on a remote server. The task here is working on the encrypted data information. There are two forms of homomorphic data encryption i.e., fully homomorphic encryption and partially homomorphic encryption. Fully homomorphic encryption permit impulsive computation on the ciphertext in a ring whereas the partly homomorphic encryption is that the one within which addition or multiplication operations may be administered on the conventional ciphertext. The homomorphic encryption plays a significant role in cloud computing because the encrypted information of corporations is keep in a very public cloud, therefore taking advantage of the cloud provider's services. The various algorithms and ways of fully homomorphic encryption that have been projected are mentioned in this paper.

Keywords: Cloud computing, Fully Homomorphic Encryption, Homomorphic encryption, Harmonic decryption, Security.

I. INTRODUCTION

The term encryption refers to converting the original data into human unreadable form (encoding). The conversion of the encoded data into original form is known as decryption. By encrypting the data only the authorized person can decode the original data. Thus data confidentiality is achieved by the encryption. In this paper, we reviewed the algorithms proposed for the homomorphic encryption of data in cloud computing [1].

Cloud computing is a type of Internet-based computing that provides shared computer processing resources and data to computers and other devices on demand. It is a model for enabling ubiquitous, on-demand access to a shared pool of configurable computing resources e.g., servers, computer networks, storage, applications and services, which can be rapidly provisioned and released with minimal management effort. Cloud computing and storage solutions provide users and enterprises with various capabilities to store and process their data in either privately owned, or third-party data centers that may be located far from the user–ranging in distance from across a city to across the world. Cloud computing relies on sharing of resources to achieve coherence and economy of scale, similar to a utility such as the electricity grid, over an electricity network [2], [3]. The cloud computing is shown in the figure 1.

Advocates claim that cloud computing allows companies to avoid up-front infrastructure costs (e.g., purchasing servers). As well, it enables organizations to focus on their core businesses instead of spending time and money on computer infrastructure. Proponents also claim that cloud computing allows enterprises to get their applications up and running faster, with improved manageability and less maintenance, and enables Information technology teams to more rapidly adjust resources to meet fluctuating and unpredictable business demand. Cloud providers typically use a "pay as you go" model. This will lead to unexpectedly high charges if administrators do not adapt to the cloud pricing model.

In 2009, the availability of high-capacity networks, low-cost computers and storage devices as well as the widespread adoption of hardware virtualization, service-oriented architecture, and autonomic and utility computing led to a growth in cloud computing.

Figure 1. Structure of Cloud Computing

Companies can scale up as computing needs increase and then scale down again as demands decrease. In 2013, it was reported that cloud computing had become a highly demanded service or utility due to the advantages of high computing power, cheap cost of services, high performance,

scalability, accessibility as well as availability. Some cloud vendors are experiencing growth rates of 50% per year, but being still in a stage of infancy, it has pitfalls that need to be addressed to make cloud computing services more reliable and user friendly [1], [2], [3].

II. SECURITY OF CLOUD COMPUTING

The use of cloud computing has increased rapidly in many organizations. Concomitantly, the problems of third party data security and securely outsourcing computation become increasingly prominent. There is the risk that personal information sent to a cloud provider is often seen as valuable to individuals with malicious intent and might be kept indefinitely or used for other purposes. Also, such information could also be accessed by government agencies, domestic or foreign and this might affect the privacy of user. There are some security issues in cloud computing such as data security, third-party control, and privacy. If all data stored in cloud were encrypted using traditional cryptosystems, this would effectively solve the three above issues.

To perform a required computation on encrypted data stored in cloud, a user must share the secret key with cloud provider. First, cloud provider decrypts the data to execute necessary operations then sends the result to the user. To solve this issue, it is necessary to use a cryptosystem based on homomorphic encryption to encrypt the data. Since these cryptosystems allow to do computation on encrypted data [4], [5].

III. HARMONIC ENCRYPTION

The concept of homomorphic encryption was suggested in 1978 by Ronald Rivest and Leonard Adleman. But for 30 years the progress is very slow. In 1982, Shafi Goldwasser and Silvio Micali proposed their encryption system that able to encrypt one bit in additive homomorphic encryption. Pascal Paillier 1999 suggested another additive homomorphic encryption [6], [7], [8]. The harmonic encryption and decryption is shown by the figure 2.

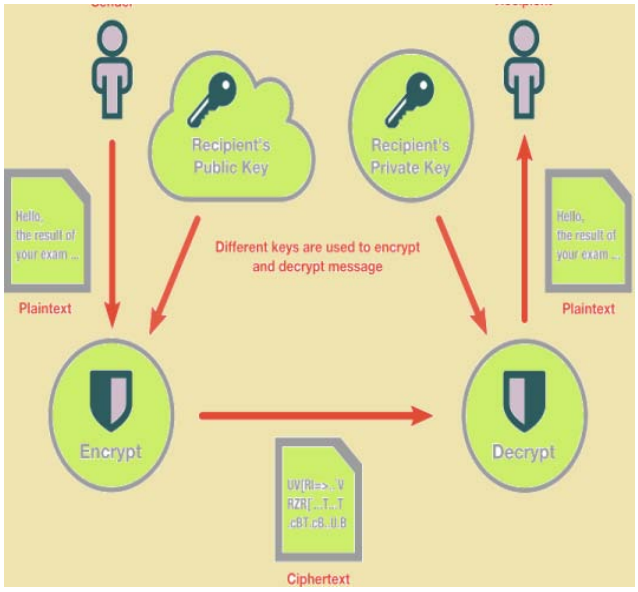

Figure 2. Data encryption and decryption [9] structure

In 2005, Dan Boneh, Eu-Jin Goh and Kobi [10] invented a security system of encryption which conducts only single multiplication but large number of additions. In 2009 [11], Craig Gentry construct a fully homomorphic encryption based system that able to conduct both of addition and multiplication in the same time. Homomorphic encryption is a form of encryption that allows computations to be carried out on ciphertext, thus generating an encrypted result which, when decrypted, matches the result of operations performed on the plaintext. There are two main categories of homomorphic encryption schemes: Partially Homomorphic Encryption (PHE) and Fully Homomorphic Encryption (FHE) schemes. PHE schemes, such as RSA, ElGamal, Paillier [12], etc., allow performing either addition or multiplication on encrypted data. Construction of scheme supporting both operations simultaneously was elusive. Although Boneh et al. [12] came closest, allowing unlimited additions and a single multiplication, It was not until 2009 that the three decade old problem was solved in seminal work by Gentry [11], where he showed that performing both addition and multiplication simultaneously are possible in fully homomorphic encryption.

This is sometimes a desirable feature in modern communication system architectures. Homomorphic encryption would allow the chaining together of different services without exposing the data to each of those services. For example, a chain of different services from different companies could calculate the tax, the currency exchange rate and shipping on a transaction without exposing the unencrypted data to each of those services. Homomorphic encryption schemes are malleable by design. This enables their use in cloud computing environment for ensuring the confidentiality of processed data. In addition the homomorphic property of various cryptosystems can be used to create many other secure systems, for example secure voting systems, collision-resistant hash functions, private information retrieval schemes, and many more.

Let $(\phi; \psi; K; E; D)$ be an encryption scheme, where ϕ, ψ are the plaintext and ciphertext spaces, K is the key space, and $E; D$ are the encryption and decryption algorithms. Assume that the plaintexts forms a group (ϕ, o) and the ciphertexts forms a group $(\psi, *)$, then the encryption algorithm f is a map from the group ϕ to the group ψ, i.e.,

$$f_k: \phi \to \psi;\ \forall\ k \in K \qquad (1)$$
$$\forall\ a, b\ \in \phi;\ k\ \in K,$$

where $k \in K$ is either a secret key or a public key.

The f_k is homomorphic if

$$f_k\ (a\ o\ b) = f_k\ (a) * f_k\ (a) \qquad (2)$$

Let us consider the public key $p_k = (p, q)$ in an unpadded RSA plaintexts form a group (ϕ, x), and the ciphertexts form a group (ψ, x), where x is the modular multiplication. For any two plaintexts $a, b \in \phi$, holds that

- $f\ (a, p_k)$ x $f\ (b, p_k) = a^q$ x $b^q\ (mod\ p)$
- $f\ (a, p_k)$ x $f\ (b, p_k) = (a$ x $b)^q\ (mod p)$
- $f\ (a, p_k)$ x $f\ (b, p_k) = f\ (a$ x $b, p_k\ )$

Hence the unpadded RSA has the homomorphic property. But, the unpadded RSA is insecure [11].

IV. FULLY HARMONIC ENCRYPTION

Principally, FHE permits for capricious computations on encrypted information. Computing on encrypted information implies that if a user includes a operator *f* and need to get *f* $(a_1, a_2,\ldots,a_n)$ for a few inputs $a_1, a_2,\ldots,a_n$, it's doable to instead cypher on encryptions of those inputs, $b_1, b_2, \ldots..,b_n$, getting a result that decrypts to *f* $(a_1, a_2,\ldots,a_n)$. In some cryptosystems the input messages lay at intervals some pure algebraic structure, usually a gaggle or a hoop. In such cases the ciphertexts can usually additionally lie at intervals some associated structure, which may be an equivalent as that of the plaintexts. The operator *f* in older homomorphic coding schemes is often restricted to be an algebraic operation related to the structure of the plaintexts. We will specific the aim of totally homomorphic coding to be to increase the operate *f* to be any operate [12]. The process of FHE is as like in the figure 3.

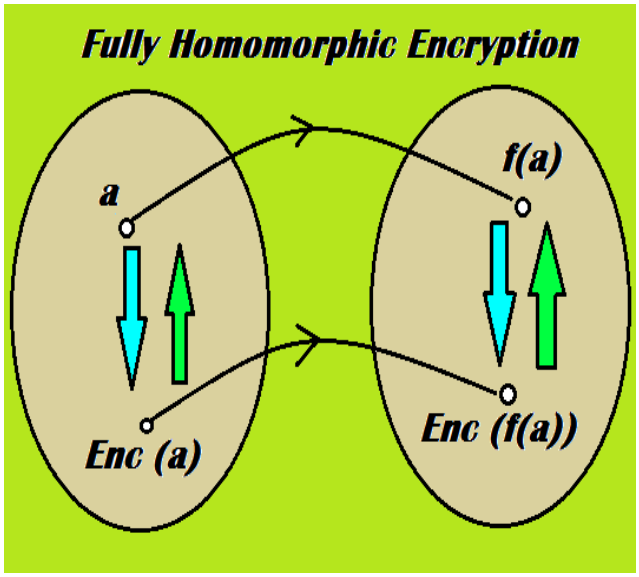

Figure 3. Process of fully harmonic encryption

This aim will be achieved if the theme is homomorphic with relation to a functionally complete set of operations and it's doable to retell operations from that set. Whereas it's invariably a demand that coding schemes are economical in an exceedingly theoretical sense, particularly running in polynomial time within the security parameter, sensible potency wasn't the primary priority in getting the primary FHE schemes. One reason for the shortage of efficiency of those schemes is that they use a plaintext space consisting of one bit and are homomorphic with relation to $+_2$ and X_2 i.e., addition and multiplication modulo 2. While any function of any complexity can be built up from such basic operations, which may require a large number of such operations. In order to move towards better efficiency, some recent variants of FHE schemes restrict the functions *f* in different ways which we will explore later. Although a theoretical view of FHE cares only about maximizing the choices of *f*, a practical view cares also about keeping this choice only as large as needed, and may also prefer a richer structure for the plaintext and ciphertext spaces that just the binary case.

In a thesis submission, Craig Gentry (2009), [11] has proposed the first fully homomorphic encryption scheme to solve the central open problem in cryptography. This allows us to compute the arbitrary functions over the encrypted data without any decryption key i.e., given encryptions $E(m_1),\ldots..,E(m_t)$ of $m_1,\ldots..,m_t$, we can efficiently compute the compact ciphertext that encrypts $f(m_1,\ldots..,m_t)$ for the efficiently computable function *f*.

Maha Tebba et al. (2012), [10] have analyzed that the cloud computing security based on fully homomorphic encryption, is the new concept of security which enables us to provide results of calculations on the encrypted data without knowing the raw data by which the calculations was performed, with respect to the data confidentialy.

Huang Qin-long et al. (2013), [13] have proposed a well-secure and privacy-oriented digital rights management. i.e., known as DRM scheme by using homomorphic encryption in the cloud computing.

B. K. Mohanta et al. (2013), [14] have analyzed the various security issues to present. As the client send the information to the server in the encrypted form to do the computational instructions to the encrypted data server need the personal key from the client.

Shashank Bajpai et al. (2014), [15] have given the model on the cloud computing that accepts encrypted inputs and then perform the blind processing to satisfy the user query without knowing of its content, whereby the recovered encrypted data can only be decrypted by the user who starts the request.

Iram Ahmad et al. (2014), [16] have analyzed that the security of cloud computing based on fully homomorphic encryption is the new concept of the security that is enable to provide the results of calculations on the encrypted data without knowing the raw entries on which the calculations was carried out with respect to the confidentiality of the data.

In a project submitted by Jiarui Huang et al. (2014), [17] developed a model of FHE because of the noise built up from performing the operations. Some have proposed the encryption schemes that have bounded depth but do not apply to much complicated functions. The FE is similar to FHE but gives more access control and can be applied to multiple different cases.

Payal V. Parmar et al. (2014), [18] have presented the basic concept of the homomorphic encryption and the different encryption algorithms as per the characteristics of the homomorphic encryption; Paillier can be suitable for preserving the addition property of homomorphic encryption and ElGamal; RSA can be used for the multiplication property.

X. Yi et al. (2014), [19] said that the homomorphic encryption is the form of encryption that allows some specific types of computations to be carried out on ciphertexts and generate an encrypted result which, on decrypted, matches the result of operations performed on the plaintexts.

Frederik Armknecht et al. (2015), [20] said that fully homomorphic encryption has been dubbed the holy grail of cryptography, an elusive goal which could solve the IT world's problems of security and trust.

Siddhi Khamitkar (2015), [21] has concluded that Fully Homomorphic Encryption has played a big role in the security and the confidentiality of the data and is being applied on the large scale in a lot of applications over the cloud.

Ihsan Jabbar et al. (2016), [12] have concluded that fully homomorphic encryption is the best technique to secure the

client data in the cloud computing because its schemes enable to perform arbitrary computations on encrypted data without decrypting. The Gen10 and DGHV schemes of FHE are insecure when they are used in the cloud computing for securing the data of client. The SDC is the simple and taken as efficient scheme to secure the data in cloud computing.

Kamal Benzekki, et al. (2016), [22] said that the most FHE schemes are based on Gentry's blueprint that consisting of first constructing a SHE and then using Gentry's bootstrapping technique to convert it into again FHE scheme. It turns out that bootstrapping is the major bottleneck and that SHE is actually well-efficient.

Amit Chaturvedi et al. (2017), [23] said that in the cloud computing, the data is place on the third party servers, and the customer is completely unaware about the location of the server. The handling of data, at the server side, is totally in a third party control. Then the cryptography schemes may improve the security level of data. From the customers or organizations view point, the data needs to be encrypted while being stored on a cloud. The cloud servers should be able to compute on the encrypted data and so the queries should be answered on the basis of the encrypted data and hence the responses are also encrypted data.

V. IMPEMENTATION OF FHE

Craig Gentry of IBM in 2009 has planned the primary secret writing system "Fully Homomorphic" that evaluates a discretionary variety of additives and multiplications and therefore calculates any sort of perform on encrypted knowledge. The inner operating of this adds another layer of secret writing each few steps and uses an encrypted key to unlock the inner layer of scrambling. This decipherment "refreshes" the information while not exposing it, permitting an infinite variety of computations on an equivalent. The applying of fully Homomorphic secret writing is a vital stone in Cloud Computing security; additional usually, we have a tendency to may source the calculations on confidential knowledge to the Cloud server, keeping the key that may decrypt the results of calculation. In our implementation, we have a tendency to analyze the performance of the prevailing Homomorphic secret writing cryptosystems, we have a tendency to square measure engaged on a virtual platform with ESX as a Cloud server, a VPN network that links the Cloud to the consumer (enterprise), then later we have a tendency to started by simulating completely different eventualities exploitation the Computer Algebra System Magma tools [7], which focusing on the scale of the general public key and its impact on the scale of the encrypted message, the server delay of the request treatment in step with the scale of the encrypted message and the result decrypting time of the request in step with the cipher text size sent by the server.

VI. FHE ON CLOUD

The application of fully Homomorphic encoding is a very important brick over Cloud Computing Security; additional typically, outsourcing of the calculations on confidential information to the Cloud server is feasible, keeping the key which will rewrite the results of calculation. In our implementation, we tend to analyze the performance of existing homomorphic encoding cryptosystems, and are functioning on a virtual platform as a Cloud server, a VPN network that links the Cloud with the client, so simulating totally different eventualities. For instance a Database-Server act with shopper victimization FHE Cryptosystem is as shown within the figure below.

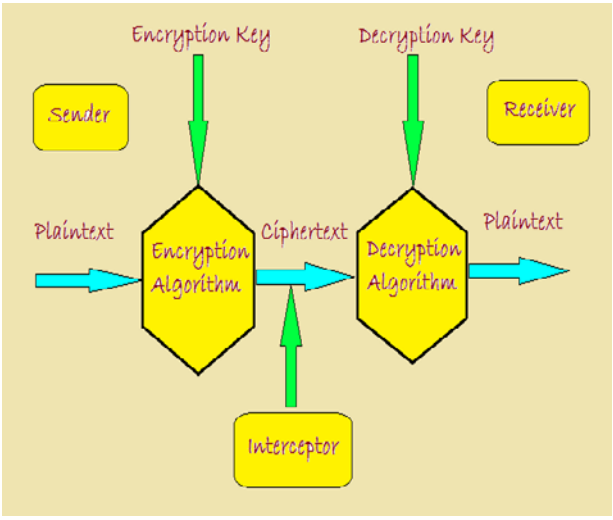

Figure 4. FHE Cryptosystem

The double layer of secret writing causes the system runs too slowly for sensible use. We tend to area unit acting on optimizing a similar for specific applications like looking out databases for records cut back the time quality. Additionally to trust an awfully new secret writing theme for confidentiality isn't possible and it needs sizable (~10 yrs) of usage exposure. A team from MIT's computing and computer science Laboratory, World Health Organization worked in conjunction with the University of Toronto and Microsoft analysis, wanted to mix multiple schemes to resolve these challenges. The system starts with homomorphic secret writing, with a secret writing algorithmic program embedded in a very illogical circuit that is itself protected by attribute-based secret writing this ensures the method stays encrypted.

VII. CONCLUSION

The security of cloud computing supported fully homomorphic encoding could be a new thought of security that is to modify to supply the results of calculations on encrypted knowledge while not knowing the raw entries on that the calculation was applied respecting the confidentiality of data information. Our work relies on the applying of fully Homomorphic encoding to the protection of Cloud computing i.e., to research and improve the present cryptosystem to permit servers to perform varied operations requested by the consumer and to boost the quality of the homomorphic encoding algorithms and study the interval to requests in step with the length of the general public key. During this analysis we've analyzed the various security issue show once consumer send information to the server in encrypted type to perform any process operation to it encrypted knowledge server want the non-public key from the consumer. If consumer provides that non-public key then the privacy isn't ensured, once more a way to make sure that no one can perform any unauthenticated operation with the information. Currently we've given some fully homomorphic encoding theme developed by researchers which permit us to perform computation on encrypted

knowledge while not mistreatment secret key of consumer. It's nothing however a replacement layer applied to the cloud computation.

As cloud computing could be a massive space and use of cloud computing is increasing daily. once more cloud having three service model that area unit code as a service (*saas*), platform as a service (*paas*), and infrastructure as a service (*iaas*) therefore most are trying to maneuver into the cloud because it provide a lot of flexibility and reduced value. Here we've implement absolutely homomorphic encoding theme wherever all kind of operation area unit will be performed while not knowing secret key. The most defect of this theme is that when the dimensions of information cypher becomes terribly massive which is able to cause significant burden for network and storage.

VIII. REFERENCES

[1] M.Teeba, S. E. Hajii, "Secure Cloud Computing through Homomorphic Encryption, International Journal of Advancements in Computing Technology (IJACT)," Volume 5, Number 16, December 2013, pp. 29-38.

[2] K. Benzekki, A. E. Fergougui, and A. E. B. E. Alaoui, "A Secure Cloud Computing Architecture Using Homomorphic Encryption," International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 7, No. 2, 2016, pp. 293-298.

[3] F. Farokhi, I. Shames and N. Batterham, "Secure and Private Cloud-based control using semi-homomorphic encryption", IFAC-Papers OnLine 49-22, 2016, pp. 163–168.

[4] B. K. Mohanta and D. Gountia, "Fully homomorphic encryption equating to cloud security: An approach," IOSR Journal of Computer Engineering (IOSR-JCE), Volume 9, Issue 2 (Jan. - Feb. 2013), pp. 46-50.

[5] H. Qin-long, M. Zhao-feng, Y. Yi-xian, F. Jing-yi, and N. Xin-xin, "Secure and privacy-preserving DRM scheme using homomorphic encryption in cloud computing," The Journal of China Universities of Posts and Telecommunications, December 2013, 20(6): 88–95.

[6] X. Wang, "One-round secure fair meeting location determination based on homomorphic encryption," Information Sciences, 372 (2016), pp. 758–772.

[7] C. Stuntz, "What is Homomorphic Encryption, and Why Should I Care?," March 18, 2010. https://en.wikipedia.org/wiki/Homomorphic_encryption.

[8] R. Rivest, "Lecture Notes 15: Voting, Homomorphic Encryption," October 29, 2002. https://en.wikipedia.org/wiki/Homomorphic_encryption.

[9] Supinfo International University text for Data encryption and decryption structure, 2016. https://www.supinfo.com/articles/single/3654-modern-type-of-cryptography.

[10] M. Tebaa, S. E. Hajji, A. E. Ghazi, "Homomorphic Encryption Applied to the Cloud Computing Security," Proceedings of the World Congress on Engineering 2012, Vol I, WCE 2012, July 4 - 6, 2012, London, U.K.

[11] C. Gentry, "A Fully Harmonic Encryption Scheme", A Dissertation Submitted to the Department of Computer Science and the Committee of Graduate Studies of Stanford University in partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy, September 2009.

[12] I. Jabbar and S. Najim, "Using Fully Homomorphic Encryption to Secure Cloud Computing," Internet of Things and Cloud Computing, Vol. 4, No. 2, 2016, pp. 13-18. doi: 10.11648/j.iotcc.20160402.12

[13] H. Qin-long, "Secure and privacy-preserving DRM scheme using homomorphic encryption in cloud computing," The Journal of China Universities of Posts and Telecommunications, December 2013, 20(6): 88-95. DOI: 10.1016/S1005-8885(13)60113-2

[14] B. K. Mohanta and D. Gountia, "Fully Homomorphic Encryption Equating to Cloud Security: An approach," IOSR Journal of Computer Engineering (IOSR-JCE), Volume 9, Issue 2 (Jan. - Feb. 2013), pp. 46-50.

[15] S. Bajpai and P. Srivastava, "A Fully Homomorphic Encryption Implementation on Cloud Computing," International Journal of Information & Computation Technology, Volume 4, Number 8 (2014), pp. 811-816.

[16] I. Ahmad and A. Khandekar, "Homomorphic Encryption Method Applied to Cloud Computing," International Journal of Information & Computation Technology, Volume 4, Number 15 (2014), pp. 1519-1530.

[17] J. Huang, M. Zhang, W. Chen and Yi-Shiuan Tung, "Computing on Encrypted Data," 6.857 Final Project, May 15, 2014.

[18] P. V. Parmar et al., Survey of Various Homomorphic Encryption algorithms and Schemes, International Journal of Computer Applications, Volume 91 – No.8, April 2014, pp.26-32.

[19] X. Yi, R. Paulet and E. Bertino, "Homomorphic Encryption and Applications", 2014, XII, 126p, ISBN: 978-3-319-12228-1.

[20] F. Armknecht et al., A Guide to Fully Homomorphic Encryption, https://eprint.iacr.org/2015/1192.pdf.

[21] S. Khamitkar, "A survey on Fully Homomorphic Encryption," IOSR Journal of Computer Engineering (IOSR-JCE,) Volume 17, Issue 6, Ver. III (Nov – Dec. 2015), pp. 10-14. DOI: 10.9790/0661-17631014.

[22] K. Benzekki et al., "A Secure Cloud Computing Architecture Using Homomorphic Encryption," International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 7, No. 2, 2016 pp. 293-298.

[23] A. Chaturvedi, A. Kapoor and V. Kumar, "A review of homomorphic encryption of data in cloud computing," International Journal of Computer Trends and Technology (IJCTT), Volume 43, Number 2, January 2017, pp. 75-80.