ISSN(Online) : 2456-8910
International Journal of Innovative Research in Applied Sciences and Engineering (IJIRASE)
Volume 4, Issue 3, DOI: 10.29027/IJIRASE.v4.i3.2020.638-643, September 2020

Homomorphic Encryption for Solving Security Issues in Cloud Computing

[1]Rachna Jain, [2]Meenu Gupta, [3]Akash Gupta
[1,2,3]Department of Computer Science & Engineering
[1,3]Bharati Vidyapeeth's College of Engineering, Delhi, India, [2]Chandigarh University, Punjab
Email :rachna.jain@bharatividyapeeth.edu, gupta.meenu5@gmail.com,
akashgupta752000@gmal.com

Abstract:

Even with so many cryptographies present, homomorphic encryption has attracted wide attention from various scholars of this field due to its special and optimized performance. This lies down to the fact that the common cryptography methods cannot perform computations on encrypted data, and homomorphic encryption can. Also, homomorphic encryption provides automatically encrypted operation results. Since data breach is the biggest threat in the field of cloud computing, homomorphic encryption has a wide and very useful application in data security in a cloud environment. The main aim of this paper is to give an outline of the current security threats in the field of cloud computing and also throw light on the findings of homomorphic encryption, which include functional computations with operations such as addition, multiplication and then study them for the security of the cloud. The results produced only selected features vital for the prediction of cancer. Also, its performance has been paralleled against other factors such as Accuracy, Precision, Recall and Specificity, and F-measure. The experimental results show that the Decision Tree classifier outperforms all other classifiers with an accuracy of 94.7 % increased to 97% after Cuckoo Optimization.

Keywords: Partial homomorphic encryption, fully homomorphic encryption, Security concern, cryptography, cloud computing.

1. INTRODUCTION

Cloud computing is defined as deploying on a cloud platform that is the Internet. It allows accessing to data over the Internet instead of relying on the computer's physical memory [13]. In today's time, due to huge volumes of data present and needed, it is very difficult to imagine storing all your data on your system's hard drive. Even though storing data on the cloud provides various advantages like flexibility, reduction in cost for the organization, etc., it has one major issue, i.e., data security. Cloud service providers have adopted multiple measures to provide better protection to data of its users, but still, this problem is so vast that we need to focus more on it [7]. It is a system, which could assure data security in the IT world. Gentry gave the scheme of fully homomorphic encryption. He gave the solution by making the use of lattices.

In homomorphic encryption, the user is given a key, which is so cured, and private, and operations like addition, multiplication, etc. are carried out on the encrypted data. This leads to the branch of technology known as homomorphic encryption. When we decrypt the result we obtain, it seems as if we carried out real calculations and computed the results [9]. Many cloud service providers allow users to use their platforms and create their web services. Along with the online platform, these cloud service providers also provide better security and flexibility, which has helped these organizations to store data at a lower cost [12] [14]. There are three types of cloud computing [15] [16]:

- **Public Cloud Computing:** It means using the Internet to provide services. These cloud services provide assistance and platform to the public, and the user himself, or a particular organization, can protect a part of the platform and network.
- **Private Cloud Computing:** In private cloud computing, a single company owns a data platform that often has much more specific security controls than a public cloud does. This can lead to users having more confidence and control.
- **Hybrid Cloud Computing:** This form of computing is a blend of the other two types of cloud computing. Public and Private clouds combine to form this.

There are various security problems that users have to deal with while putting data on the cloud. When a user deploys data in a public cloud, that user loses any control over its confidentiality, integrity, or

ISSN(Online) : 2456-8910
International Journal of Innovative Research in Applied Sciences and Engineering (IJIRASE)
Volume 4, Issue 3, DOI: 10.29027/IJIRASE.v4.i3.2020.638-643, September 2020

even its availability. With an increase in the number of devices, applications, and parties involved, the danger of data compromise grows in the cloud, and that increases the number of points of access [17]. Virtualization and multi-tenancy also reside in the list of major issues for the users of cloud computing. To eliminate the feasibility of data or ongoing transactions from one existing domain into another, users want to make sure that the domains are properly isolated from each other since the cloud is a shared environment. System vulnerabilities are the next major issue. System vulnerabilities are bugs found in programs that attackers use to penetrate a system to steal sensitive data, disrupt service operations, or take control of the whole system. Now, attackers attain the ability to operate your login information to alter and manipulate data through hijacked identities. Systems from various industries are placed near to each other through multi-tenancy in the cloud and permitted to access shared resources and memory, creating a subsequent surface susceptible to attacks. Malware threats are small sections of code or scripts implanted into the services present on the cloud that act as if they are valid and normally run on cloud servers. This says that this code can be put into cloud services and seen as part of the software or service that is running in the cloud servers themselves. Attackers can easily compromise the degree of sensitive information, and also steal data once threats are injected. Due to their nature, APIs can be a big danger to cloud security. They don't allow organizations to modify characteristics of their cloud services to suit the particular business needs, but they also authenticate, and effect encryption [18]. A solution to all these problems like data confidentiality, privacy, and integrity can be found in implementing homomorphic encryption.

2. RELATED WORKS

Cloud security is compromised a lot these days. To fight it, many techniques have been devised, and one of these is homomorphic encryption. Dijk et al. in 2010 [1] talked about the four stages of homomorphic encryption and how encryption, decryption takes an all different turn in this case. They have referred to somewhat homomorphic encryption and used the techniques of modular arithmetic. Gentry's methods have been discussed quite elaborately. It used addition and multiplication as the basic modules. Smart et al. in 2010 [2] described more about Gentry's fully homomorphic model and gave a proposal of its own. They dwelt primarily on integers in their model and used ciphertexts. Jules et al. in 2010 [3] presented an encryption technique that could be delivered to secure the cloud in a better way. They argued that cryptography isn't the only way to secure text or messages. It showcases the drawback of the technique when client sharing occurs. Gennaro et al. in 2010 [4] elaborated further on the concept of encryption techniques and how it can be used in different platforms. The privacy of the client is kept at the pedestal. The user has no clue what happens at the backend. Chun g et al. in 2010 [5] talked about the various computation ways that can make the encryption better. A large pubic key is eliminated in this model. A delegator is appointed, and two stages of computation, like online and offline, are used.

Atayero et al. in 2011 [6] emphasize the mathematical portion of the encryption strategy. IT showed that the RSA scheme was followed by partial and fully homomorphic encryption. It also presented a generalization of Gentry and the program he suggested. Tianfield et al. in 2012 [7] discussed the various security issues prevalent nowadays in the field of cloudlike data confidentiality, integrity, malware attacks. Lopez et al. in 2012 [8] gave the limitations of homomorphic encryption. It deals with multiple parties performing the computation. User interaction is important in the field. Boyd et al. in 2015 [11] told about the idea of functions and applying them to encrypted data. Fully homomorphic encryption discussed in detail. The generation of key, encryption, evaluation, and decryption were the stages of encryption. More people explained security and how efficient techniques like homomorphic encryption can be applied to protect the cloud.

3. PARTIAL HOMOMORPHIC CRYPTOSYSTEMS

A cryptosystem is called partially homomorphic when either multiplicative or additive homomorphism exists. When both present, then it has not happened [20]. Some examples of the above are:

Table 1: Examples of the partial homomorphic cryptosystem

RSA	Shows Multiplicative homomorphism
ElGamal	Shows Multiplicative homomorphism
Paillier	Shows Additive homomorphism

3.1. RSA

In RSA, the key provided to the user is made public to perform encryption, and it is not like the key that needs to be decrypted, which is hidden and kept secret in this cryptosystem.

ISSN(Online) : 2456-8910
International Journal of Innovative Research in Applied Sciences and Engineering (IJIRASE)
Volume 4, Issue 3, DOI: 10.29027/IJIRASE.v4.i3.2020.638-643, September 2020

On multiplying two or more ciphertexts with each other (i.e., multiplicative homomorphism), the decrypted result that is obtained comes out equal to the consequent multiplication of the two or more initial values.

Example: If, E (y): encryption of message y, Modulus m: RSA public key, and e: exponent
Then, Encryption of the message y would be given by E (y) = ye mod m and, the homomorphic property is provided by:

E (y1). E (y2) =y1ey2e mod m = E (x1.x2)

3.2. ElGamal

The ElGamal encryption system is based on the Diffie-Hellman Key Exchange. This algorithm is asymmetric and finds its use in encryption using the concept of the public key. An added form of security is achieved in this system by encrypting keys in an asymmetric fashion that was previously used for symmetric encryption of incoming messages. Upon performing the multiplication of each component of several ciphertexts with their corresponding elements, the result arrived after decryption is identical to the product of text values. (Multiplicative homomorphism)
In the ElGamal system, there is a cyclic group of C (of order l) with the generator n and, Public key is (C, n, l, h) where h=ny, where y is the secret key and, Encryption of the message m would be E(m)=(gs,m.hs) for some randoms[21]

3.3. Paillier

It is an asymmetric type of algorithm that relies on probabilistic assumptions used for encryption on public keys of text values. In this, after computing the product of each component of the various cypher texts available to the respective parts, the result obtained after the decryption process is equal to the addition of original text values. (i.e., Additive homomorphism).

Encryption :

plaintext $m < n$

select a random $r < n$

ciphertext $c = g^m \cdot r^n \bmod n^2$

Decryption :

ciphertext $c < n^2$

plaintext $m = \dfrac{L(c^\lambda \bmod n^2)}{L(g^\lambda \bmod n^2)} \bmod n$

Fig.1. Public-Key Cryptosystem

If,
There are two numbers *x* and *y* (such that x and y are prime) and *n=x.y* and,
Lambda (*n*)=*lcm (x-1, y-1),g* is some integer,
Then, The public key would be *(n, g)* and
The private key would be *(Lambda, u);*
where *u = L(g lambda mod n2)-1 mod n L (u)=u-1/u [22]*

4. FULLY HOMOMORPHIC ENCRYPTION

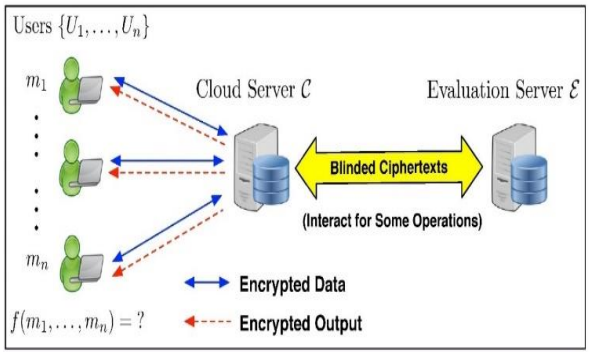

Fig.2. Procedure for Homomorphic encryption

The calculations for any encrypted form of data can be computed using this encryption technique. This way, information remains confidential when it is being processed. This enables tasks to be accompanied by knowledge, even in suspicious environments. In cloud computing, a large amount of data is sent into cloud storages (which are often unencrypted), so a huge amount of trust is required in these cloud storage providers. The Cloud Security Alliance lists data breach as the top threat to cloud security [10]. Data encryption using conventional encryption methods removes the problem of a data breach, but the user would be unable to operate on the encrypted data. The user would have to download the required data to its system and perform computations locally. By using Fully Homomorphic Encryption, computations would be achieved by cloud on the user's behalf, and it will return the encrypted result only. Principally, the fully homomorphic system allows for arbitrary computation on encrypted data [10]. This means that if the user has a function g and wants to get g (a1, a2, ... an) for inputs a1, a2... an), it would be possible to instead compute on the input's encryption (b1, b2, ... bn) and then we will obtain a result which will decrypt to g (a1, a2... an). The basic idea behind Fully Homomorphic Encryption is the ability to apply functions on data that is being encrypted [11].

4.1. Homomorphic Encryption Schemes as Building Blocks of Delegation of Computation

Apart from outsourcing data, outsourcing computations is a vital pillar in the field of cloud computing. A user can wish to delegate the computation of a function g to the server. But, the server may be prone to malfunctions or maybe just

ISSN(Online) : 2456-8910
International Journal of Innovative Research in Applied Sciences and Engineering (IJIRASE)
Volume 4, Issue 3, DOI: 10.29027/IJIRASE.v4.i3.2020.638-643, September 2020

malicious. This wouldn't allow the user to trust the result of the computation. The user could want to have a proof that the computation was performed accurately and certifying this proof should also be more efficient than the user doing the computation. Chung et al.[5] used fully homomorphic encryption to design strategies to assign computation, hence improving the results of Gennaro et al. [4]. At the same time, Van Dijk and Juels et al. [3] examined the infeasibility of FHE solving the various security and privacy issues prevalent in cloud computing alone. For a fully homomorphic system, the system can behave in a variety of ways. A fully homomorphic system will have the encryption like-

$F(a1 \Phi a2) \rightarrow F(a1) \Phi F(a2)$, ……(1)

where a1 and a2 are in T.

T is the set of texts. Φ is some random function. → denotes no decryption on texts that have been performed during computation. [6] In this case, the random function can be +, x, or another required feature.

Additive:

$F(a1+a2) \rightarrow F(a1) + F(a2)$ ……(2)

where a1 and a2 are in T.

Multiplicative:

$F(a1 \times a2) \rightarrow F(a1) \times F(a2)$ ……(3)

where a1 and a2 are in T

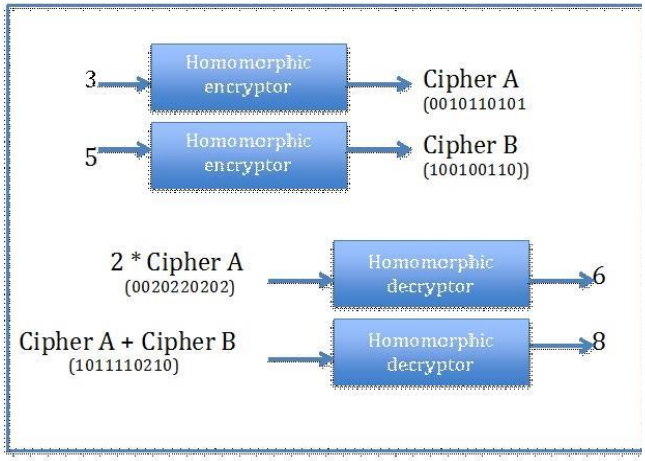

Fig.3. Procedure for Homomorphic encryption

The above diagram represents the complete procedure of homomorphic encryption [19].

5. TIME ANALYSIS OF ENCRYPTION SCHEMES W.R.T. DIFFERENT SYSTEMS

For tables, results are based on a database hosted on a remote machine. Graphs show the comparison of overhead with encrypted data over plain data for addition and multiplication operations for four algorithms.

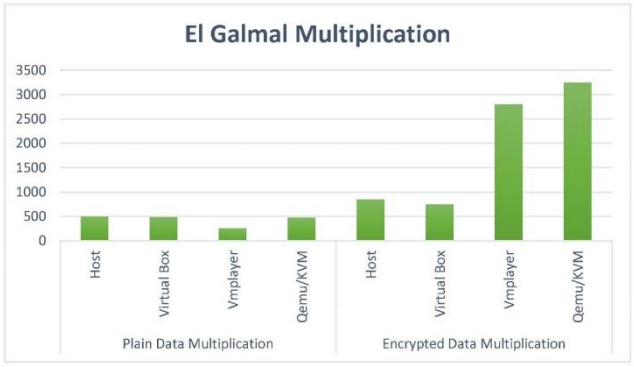

Fig.4. Comparison of overhead with encrypted data over plain data for multiplication operation for Elgamal Algorithm (y-axis in nanoseconds)

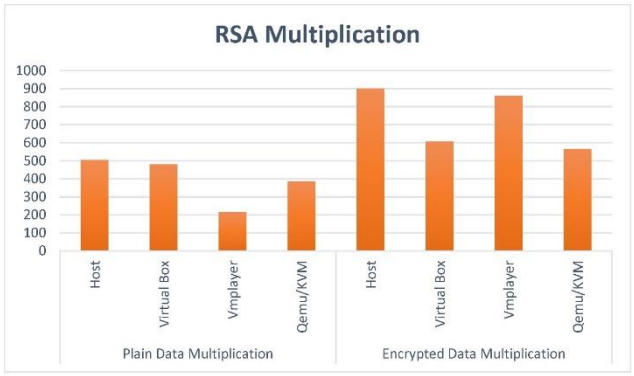

Fig.5. Comparison of overhead with encrypted data over plain data for multiplication operation for RSA

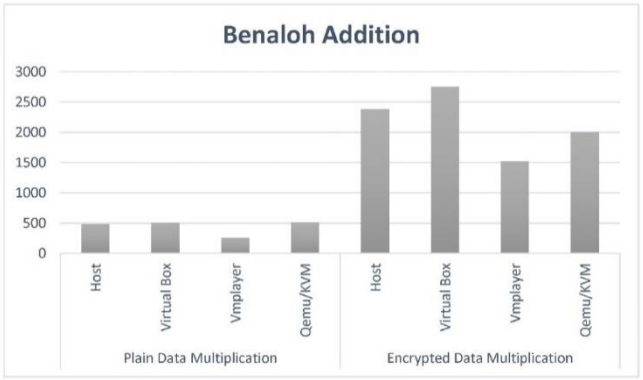

Fig.6. Comparison of overhead with encrypted data over plain data for addition operation for Benaloh Algorithm (y-axis in nanoseconds)

ISSN(Online) : 2456-8910
International Journal of Innovative Research in Applied Sciences and Engineering (IJIRASE)
Volume 4, Issue 3, DOI: 10.29027/IJIRASE.v4.i3.2020.638-643, September 2020

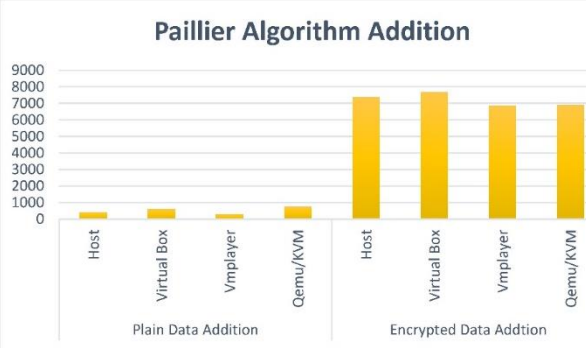

Fig.7. Comparison of overhead with encrypted data over plain data for addition operation for Paillier Algorithm (y-axis in nanoseconds)

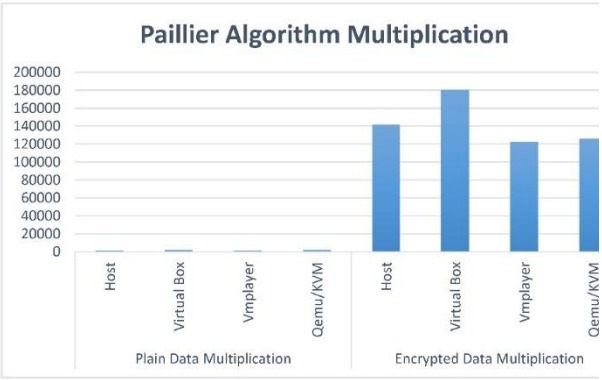

Fig.8. Comparison of overhead with encrypted data over plain data for multiplication operation for Paillier Algorithm (y-axis in nanoseconds)

6. Conclusion

In this paper, we have tried to understand various security issues and risks that are involved in inputting data on the cloud. To provide a more secure and better computation of resources, we have used the basis of homomorphic encryption. In Homomorphic encryption, the data is encrypted and mathematical operations are performed on encrypted data, produce a result, and the results are sent to the data's owner for decryption. In this way, the information is kept confidential and protected from the untrusted environment. But along with various advantages, it has some limitations too. Firstly, the support of multiple users is a problem in fully homomorphic encryption. Different users using the same system (which relies on an internal database used in computations) would wish to protect their data from the service provider. One solution could be that the provider has a separate database for every user, which would be encrypted under that user's public key. But, for large databases and the number of users, this can be rather infeasible. L´opez-Alt et al. [8] have shown promising directions to address this problem by defining and constructing multi-key FHE.

References

[1]. Van Dijk, Marten, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. "Fully homomorphic encryption over the integers." In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 24-43. Springer, Berlin, Heidelberg, 2010.

[2]. Smart, Nigel P., and Frederik Vercauteren. "Fully homomorphic encryption with relatively small key and ciphertext sizes." In *International Workshop on Public Key Cryptography*, pp. 420-443. Springer, Berlin, Heidelberg, 2010.

[3]. Van Dijk, Marten, and Ari Juels. "On the impossibility of cryptography alone for privacy-preserving cloud computing." *HotSec* 10 (2010): 1-8.

[4]. Gennaro, Rosario, Craig Gentry, and Bryan Parno. "Non-interactive verifiable computing: Outsourcing computation to untrusted workers." In *Annual Cryptology Conference*, pp. 465-482. Springer, Berlin, Heidelberg, 2010.

[5]. Chung, Kai-Min, Yael Kalai, and Salil Vadhan. "Improved delegation of computation using fully homomorphic encryption." In *Annual Cryptology Conference*, pp. 483-501. Springer, Berlin, Heidelberg, 2010.

[6]. Atayero, Aderemi A., and Oluwaseyi Feyisetan. "Security issues in cloud computing: The potentials of homomorphic encryption." *Journal of Emerging Trends in Computing and Information Sciences* 2, no. 10 (2011): 546-552.

[7]. Tianfield, Huaglory. "Security issues in cloud computing." In *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1082-1089. IEEE, 2012.

[8]. López-Alt, Adriana, Eran Tromer, and Vinod Vaikuntanathan. "On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption." In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pp. 1219-1234. 2012.

[9]. Tebaa, Maha, Said El Hajji, and Abdellatif El Ghazi. "Homomorphic encryption method applied to Cloud Computing." In *2012 National Days of Network Security and Systems*, pp. 86-89. IEEE, 2012.

[10]. Top Threats Working Group. "The notorious nine: cloud computing top threats in 2013." *Cloud Security Alliance* (2013): 1-10.

[11]. Armknecht, Frederik, Colin Boyd, Christopher Carr, Kristian Gjøsteen, Angela Jäschke, Christian A. Reuter, and Martin Strand. "A Guide to Fully Homomorphic

Encryption." *IACR Cryptol. ePrint Arch.* 2015 (2015): 1192.

[12]. Tyagi, Aayushi, Aprajita Srivastava, and Rachna Jain. "Security Issues in Cloud Computing."

[13]. Hayes, Brian. "Cloud computing." (2008): 9-11.

[14]. Gupta, Meenu, and Neha Singla. "Evolution of cloud in big data with hadoop on docker platform." In *Web Services: Concepts, Methodologies, Tools, and Applications*, pp. 1601-1622. IGI Global, 2019.

[15]. Gupta, Meenu, Rajeev Yadav, and Gundeep Tanwar. "Insider and flooding attack in cloud: A discussion." In *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 530-535. IEEE, 2016.

[16]. Furht, Borko. "Cloud computing fundamentals." In *Handbook of cloud computing*, pp. 3-19. Springer, Boston, MA, 2010.

[17]. Shaikh, Farhan Bashir, and Sajjad Haider. "Security threats in cloud computing." In *2011 International conference for Internet technology and secured transactions*, pp. 214-219. IEEE, 2011.

[18]. Vines, Ronald L. Krutz Russell Dean, and R. L. Krutz. *Cloud security: A comprehensive guide to secure cloud computing*. Wiley Publishing, Inc, 2010.

[19]. Rezaei, Shahbaz, and Xin Liu. "Deep learning for encrypted traffic classification: An overview." *IEEE communications magazine* 57, no. 5 (2019): 76-81.

[20]. Gentry, Craig. "Fully homomorphic encryption using ideal lattices." In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pp. 169-178. 2009.

[21]. Tsiounis, Yiannis, and Moti Yung. "On the security of ElGamal based encryption." In *International Workshop on Public Key Cryptography*, pp. 117-134. Springer, Berlin, Heidelberg, 1998.

[22]. Paillier, Pascal. "Public-key cryptosystems based on composite degree residuosity classes." In *International conference on the theory and applications of cryptographic techniques*, pp. 223-238. Springer, Berlin, Heidelberg, 1999.