International Journal of Electrical and Computer Engineering (IJECE)
Vol. 11, No. 2, April 2021, pp. 1771~1779
ISSN: 2088-8708, DOI: 10.11591/ijece.v11i2.pp1771-1779

Enhancing cloud computing security by paillier homomorphic encryption

Muna Mohammed Saeed Altaee, Mafaz Alanezi
Department of Computer Science, College of Computer Science and Mathematics, University of Mosul, Iraq

Article Info

Article history:

Received Feb 8, 2020
Revised Jul 26, 2020
Accepted Nov 5, 2020

Keywords:

Banking data
Cloud computing
Homomorphic encryption
Paillier algorithm
RSA

ABSTRACT

In recent years, the trend has increased for the use of cloud computing, which provides broad capabilities with the sharing of resources, and thus it is possible to store and process data in the cloud remotely, but this (cloud) is untrusted because some parties can connect to the network such as the internet and read or change data because it is not protected, therefore, protecting data security and privacy is one of the challenges that must be addressed when using cloud computing. Encryption is interested in the field of security, confidentiality and integrity of information that sent by a secure connection between individuals or institutions regardless of the method used to prepare this connection. But using the traditional encryption methods to encrypt the data before sending it will force the data provider to send his private key to the server to decrypt the data to perform computations on it. In this paper we present a proposal to secure banking data transmission through the cloud by using partially homomorphic encryption algorithms such as (paillier, RSA algorithm) that allow performing mathematical operations on encrypted data without needing to decryption. A proxy server will also use for performing re-encryption process to enhance security.

Corresponding Author:

Mafaz Mohsin Khalil Alanezi
Department of Computer Science
College of Computer Science and Mathematics
University of Mosul, Main Almajmuea Althaqafia Street, Mosul, 41002-Iraq.
Email: mafazmhalanezi@uomosul.edu.iq

1. INTRODUCTION

Cloud computing (CC) definition that provided by National Institute of Standards and Technology (NIST) of U.S. [1]: "Cloud computing is a model for enabling convenient, on demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

This cloud model promotes availability and is composed of five essential characteristics, three service models, and four deployment models". The use of CC has increased rapidly in many organizations and institutions in addition to increasing issues related to CC environment at the same time, one of these issues are security challenges, which include maintaining the security of information and the securely outsourcing computation that performed by untrusted third-party (cloud). Because there is a risk that personal information as well as sensitive information may exposed by some individuals who may use them for certain purposes may be malicious [2, 3]. Utilizing CC for the banking system may help save time and costs, but also concerns about security challenges may expose customer data to disclosure. It has become possible to communicate and share data between the bank and the customer remotely without the need for high costs, but CC remains untrusted because some parties can connect to the network and expose the data so we face

security and privacy challenges to maintain the confidentiality of banking data that may be critical and sensitive [4, 5].

Homomorphic encryption can be considered the ideal solution for securing and processing financial data in the cloud environment because it has features that enable us to perform mathematical operations on encrypted data inside the cloud without the need to decrypt it, consequently, the output from these operations is also encrypted, and only client who owns the private key can decrypts and obtain the result, which is the same if we perform the same operations on the raw data [6]. In section 2 we will talk about challenges facing banking data in the cloud. In section 3 we will list some of the proposals that used related solutions to our proposal. In section 4 we will define homomorphic encryption with mentioning its most important features and its basic categories. In section 5 we will explain the scheme of RSA algorithm. While in section 6 we will explain the scheme of paillier algorithm. In section 7 we will focus on our proposed model as a solution with the phases in which the banking data go through in the cloud, according to our scheme with the results in section 8.

2. RELATED WORKS

Due to the fact that CC is broad so storing data may subject to disclosure or change by some parties for curious or malicious purposes. Using the cloud leads to a loss of control over the banking data (account numbers, total deposits, loans ...) [4]. Therefore, different attempts and methods appeared. In 2006 B. Schoenmakers and P. Tuyls proposed a secure computation to preserve privacy using paillier scheme [7]. In 2009 C. Gentry proposed fully homomorphic encryption [8]. Huang *et al.* proposed a scheme for secure and preserving (Digital Rights Management) DRM by using homomorphic encryption in 2013 [9]. Zhang *et al.* proposed a secure method for image retrieval in the cloud computing using paillier scheme in 2014 [10]. In 2015 Tebaa *et al.* proposed a system to preserve the confidentially and privacy of banking database stored in cloud [4]. Raisaro *et al.* proposed data model can be shared in cloud environment in with privacy-preserving way using El Gamal scheme in 2018 [11]. Attempts are still ongoing by researchers to find the best ways to secure the cloud in various fields.

3. BANKING DATA IN THE CLOUD

Despite the great progress in the field of cloud computing there are several projects proposed in it, many of these projects have many problems especially when it comes to the banking area [12]. There are still a lot of concerns with the use of the cloud computing environment by many companies and individuals not only banks but in the area of banking data there are special challenges these challenges are [13, 14]:

a. Security: confidentiality of personal and financial data, such as account number, account balance and applications that perform critical tasks, is crucial. As banks cannot bear the consequences of a security breach.
b. Regulatory compliance: Financial institutions must choose appropriate service and operating models to control security concerns and compliance matters. There are some emerging cloud security services to address the risks of privacy, data security and compliance, as well as prevent attacks aimed at data theft, and to detect any breach of compliance with a robust server use for secure of virtual data centres [13].

4. HOMOMORPHIC CRYPTOSYSTEM

Many users, companies and organizations use the cloud servers to store big data and process it remotely without the use guaranteed methods to solve privacy and security problems [6]. Conventional encryption methods are one of these methods, but when data is encrypted with the public key, the client must send his private key to the cloud to decrypt the data to perform operations on it inside the cloud, which makes the data exposed. Some of these data are sensitive, such as medical or banking data, so in this paper, we present an ideal solution that guarantees the security and privacy of data in the cloud and at the same time allows mathematical operations such as addition and subtraction to be performed without the need to decode it [1, 15].

4.1. Definition of homomorphic cryptosystem

An encryption called holomorphic if: x, y is represent any integer value and from E(x) and E (y) we can compute E (*f*(x, y)), in which *f* be (+, ×, XOR) without knowing or using private key. After decoding the result, we find it is in the same case if the mathematical operation made on the raw data [16].

4.2. History of homomorphic cryptosystem

Homomorphic cryptosystem or homomorphic encryption (HE) introduced by authors (Rivest, Adleman and Dertouzos) in 1978 [17]. It considered an optimal solution because it allows the data owner to encrypt the data before sending it to the cloud and storing it on the storage server and thus all mathematical operations will be performed on encrypted data and the final results sent to the beneficiary party are encrypted and only the owner of the private key can decrypt and get the results [1, 6].

Homomorphic cryptosystem or homomrphic encryption (HE) can be classified into two mainly categories namely partially homomorphic cryptosystems or partially homomrphic encryption (PHE) and Fully Homomorphic cryptosystem or fully homomorphic encryption (FHE) [18]. Figure 1 shows the most famous encryption schemes in each category [18]. In PHE we can perform one mathematical operation on the encrypted data such as addition or multiplication but not both. While In 2009, Craig Gentry presented the first FHE scheme in his PhD dissertation [8]. This technique allowed performing arbitrary functions over encrypted data without the need to the decryption [8, 18].

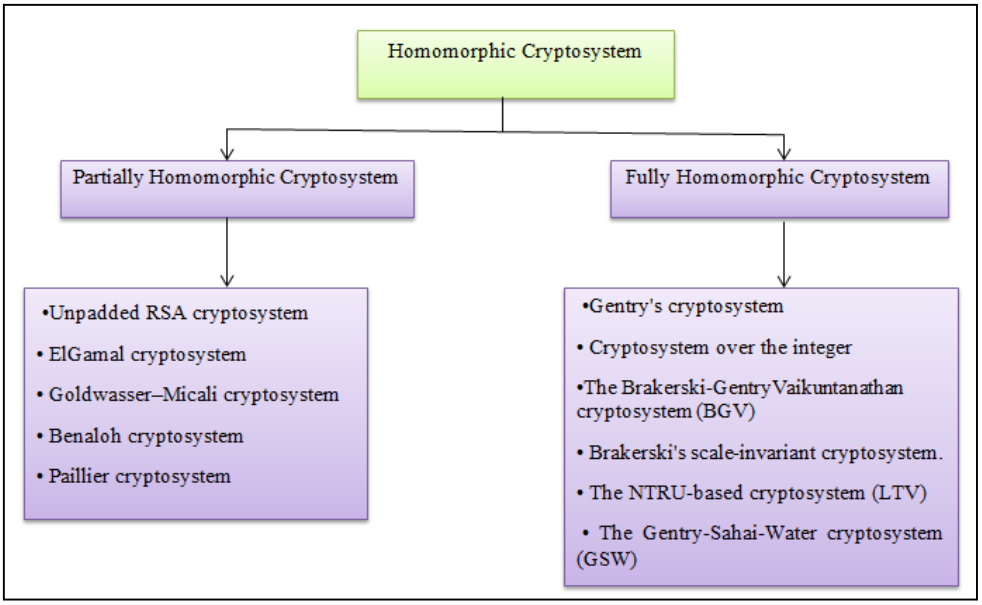

Figure 1. The main categories of homomorphic cryptosystem

4.3. Features of homomorphic encryption

In this paper we focused on PHE and Paillier algorithm will be used to encrypt banking data that belongs to its customers by using public key, before sending it to the cloud to be stored in a storage server and therefore any process will take place on the encrypted data and the result that the customer or investigator wants to get it will encrypted by its public key and decrypted by the customer's or investigator's private key who is the only owner of the key. So we will first mention to the features of HE which made it the ideal solution to the problems of security and privacy in the cloud [16]:

a. A homomorphic encryption is additive if:

$$E\left(\sum_{i=1}^{I} mi\right) = \prod_{i=1}^{I} E(mi) \quad (1)$$

where (m_i) is the raw data and its value is unknown, such as Paillier and Goldwasser-Micali algorithms.

b. A homomorphic encryption is multiplicative if:

$$E\prod_{i=1}^{I}(mi) = \prod_{i=1}^{I} E(mi) \quad (2)$$

where (m_i) is the raw data and its value are unknown, such as Paillier and Goldwasser-Micali algorithms.

4.4. Functions of homomorphic encryption

HE consists of key Functions that collectively constitute the process of securing data before storing it in the cloud in addition to retrieving the results in a secure manner these functions are [19]:

a. Key generation: for each client a pair of keys will be generated these keys are usually called public key and a secret or private key (pk, sk).
b. Encryption (E): Using public key pk client encrypts the plain text m and generate the cipher text (c), this (c) will send to the server.

$$E_{pk}(m) = c \quad (3)$$

c. Evaluation (EV): Server has a function f for doing evaluation of c EV (f(c)), and performed this as per the required function using client's public key Cpk.
d. Decryption (D): Generated EV (f(c)) will be decrypted by client using its sk and gets the original result.

$$Dsk\ (c) = m \quad (4)$$

4.5. RSA algorithm

RSA algorithm proposed by R. Rivest *et al.* in 1978 it's a public key algorithm. And it has multiplicative homomorphic propriety [16]. Algorithm 1 describe RSA algorithm [19].

Algorithm 1: RSA algorithm

a) Key generation: G*(p,q): pk,sk*
1. Chose two large prime numbers p and q randomly and independently of each other
2. Compute RSA modulus $n=p.q$
3. Compute $\varphi(n)=(p-1).(q-1)$ in which gcd $(\varphi(n),n)=1$
4. Choose e $\in\{2,\ldots, \varphi(n)-1\}$ where e is a random integer, such that gcd (e, $\varphi(n)$)=1
5. Compure $d=e^{-1}$ (mod$\varphi(n)$) (means $e.d=1$ mod $\varphi(n)$)

– the public key $(pk)=(n,e)$
– the private key $(sk)=(d)$

b) Encryption E(m,pk)
1. plain text $m\in\mathbb{Z}_{n^2}$, $pk=(n,e)$, $Z_n=\{0,1,\ldots\}$
2. Compute $c=m^e \bmod n$ $c\in Zn$

c) Decryption D(c, sk)
1. Cipher text $c \in\mathbb{Z}^*_{n^2}$, $sk=d$
2. Compute message m

$m=c^d \bmod n$ $m\in Zn$

4.5.1. Proof of multiplicative homomorphic properties for RSA algorithm

Suppose that we have c_1 and c_2 two ciphers where:

$C_1=m_1^e$
$C_2=m_2^e$

then

$C_1. C_2= (m_1. m_2)^e \bmod n$.

So, RSA has a multiplicative homomorphic properties [20].

4.6. Paillier algorithm

The Paillier algorithm cryptosystem proposed by French researcher Paillier in 1999 is public key cryptography algorithm [21]. And it is one of the supporting methods for PHE, which supports only one operation on cipher text [22]. In this method for each user will generate pair of keys public key (*pk)* that can be published and distributed to the rest of the parties and the private key (*sk*) that remains secret and cannot published. When a message is sent from the party Bob to the party Alice, the message will be encrypted with Alice's public key and then sent as an encrypted message and when it arrives to Alice, he uses his private key that corresponding to the public key to decrypt it [19, 23]. Algorithm 2 describe Paillier algorithm [23, 24].

Algorithm 2: Paillier algorithm

a) Key generation: G*(p,q): pk,sk*
1. Chose two large prime numbers p and q randomly and independently of each other
2. Compute RSA modulus $n=p.q.$
3. Compute $\varphi(n)=(p-1).(q-1)$, in which gcd $(\varphi(n),n)=1$
4. Compute Carmichael's function $\lambda=1cm(p-1,q-1)$
5. Select generator g ehere $g\in\mathbb{Z}^*_{n^2}$

g can be chosen in two ways:
a) Randomly from $\mathbb{Z}^*_{n^2}$ where gcd($g^2 \bmod n^2$.1/n.n)=1
b) Select $\propto$ and β randomly from $\mathbb{Z}^*_n$ g=($\propto$n+1) $\beta^n \bmod n^2$

6. Calculate the following modular multiplicative inverse:
   $\mu=(L(g^{\lambda} \bmod n^2)^{-1} \bmod n$
   This is means that gcd $(L(g^{\lambda} \bmod n^2), n=1$, where the function L is defined as: $L(u)=u-1/n$.
- the public key $(pk)=(n,e)$
- the private key $(sk)=(d)$

As we show that there is a mathematical relationship between the two keys. In which the private key corresponds to public key but can not be feasibly (devired from it).

a) Encryption $E(m,pk)$
   1. plain text m $\in \mathbb{Z}_{n^2}$, $pk=(n,g)$, $Z_n=\{0,1,\ldots, n-1\}$ mcc
   2. Select random r where $r \in \mathbb{Z}^*_n$
   3. Compute Chiper text according to the equation (3)

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

b) Decryption $D(c, sk)$
   1. Cipher text $c \in \mathbb{Z}^*_{n^2}$, $sk=(\lambda, \mu)$.
   2. Compute message m according to the equation (4)

   $m=L(c^{\lambda} \bmod n^2). \mu \bmod n$

4.6.1. Proof of additive homomorphic properties for Paillier algorithm

Paillier algorithm support an Additive HE properties that allows performing one operation on the encrypted data [25]. Suppose that we have c_1 and c_2 two ciphers where:

$c_1= g^{m1}.r_1 \bmod n^2$.
$c_2= g^{m2}.r_2 \bmod n^2$.

This means that $E(m_1, r_1). E(m_2, r_2) = g^{(m1+m2)}. (r_1 .r_2)^n \bmod n^2$
$= E(m_1 + m_2 \pmod n)$

Also, $D(E(m_1, r_1) . E(m_2, r_2) \bmod n^2) = m_1 + m_2 \bmod n$
and this will lead to the following identities [12]:

$\forall m_1, m_2 \in Z_n, k \in N$:

- $E(m)^k \bmod n^2) = k\, m \bmod n$.
- $D(E(m_1) g^{m2} \bmod n^2 = m_1 + m_2 \bmod n$.
- $D(E(m_1)^{m2} \bmod n^2$, $D(E(m_2)^{m1} \bmod n^2 = m_1.m_2 \bmod n$.

This means that Paillier algorithm support an Additive HE property [23, 26].

5. PROPOSED MODEL

In this paper we will present a solution for secure banking data storage and processing it in the cloud servers remotely by using PHE and re-encryption proxy server to enhance the security. The bank can exposed the resources of cloud computing to store its banking data that belongs to its customers and perform mathematical operations on it remotely on the cloud in a secure manner. We assume that we have a cloud storage server SS, and we have bank's application which represent the data provider which will use paillier algorithm to encrypt its data before storing it on the cloud storage server to be shared or sent to the customer. And we assume that we have a proxy server PS use to perform re-encryption operation to enhance the security, also we have the customer's application that represents the investigator that might be a customer or perhaps one of the bank branches sites. Figure 2 shows the structure of the proposed model for secure banking data. The proposed model will be working as the following:

a. By using paillier algorithm for both of SS and PS the two keys (*pk, sk*) will be generated and their (*pk*) keys will be combine to generate shared public key (SH_{pk}) which will be published and send to bank's application and customer's application.
b. The bank's application splits the data and uses the SH_{pk} to encrypt the data using paillier cryptosystem method and sends it to be storing on the cloud SS as encrypted data.
c. When the customer's application sends a request which is either an inquiry about the value of the balance or a request to withdraw an amount from the existing balance it must send its (C_{pk}) along with the request, and then it must receive an authentication.
d. The desired results are obtained which are either a the customer's balance status or the amount remaining in the credit card balance after the withdrawal process, the SS will perform the mathematical operations to obtain the amount remaining in the credit card balance, after that a re- encryption process will performing.

e. In re-encryption process each of SS and PS will sequentially decrypts the data (status, result) using its private key and re-encrypts it using (C_{pk}) and will send it to the customer. The statues
f. The customer's application will decrypt the status or result using its private key (C_{pk}) and thus he will obtain the required result.

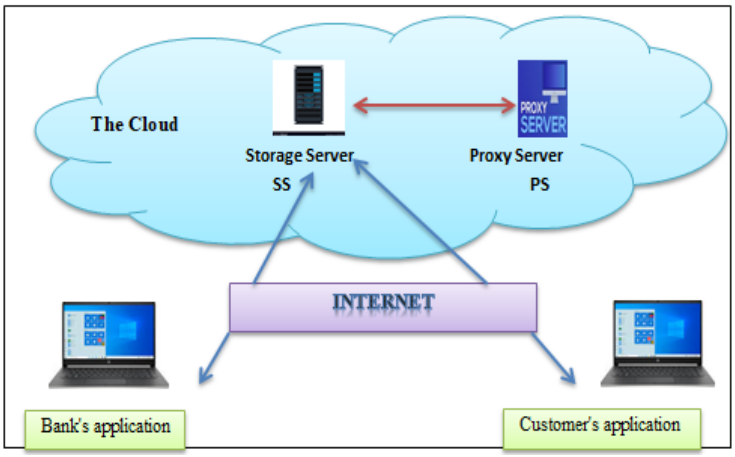

Figure 2. The structure of the proposed model for secure banking data

6. RESULTS AND DISCUSSIONS

When measuring efficiency while simulating our proposed method, we set up the proxy server and storage server on separated cloud to prevent the collision. By Intel E5-2680 V3 processor, frequency 2.50 GHz. And 1 GB RAM. And at the customer part we used laptop device with Intel Core i5 processor M-4300, frequency 2.6 GHz, RAM 8 GB. Table 1 show the database stored in bank's application. While Figure 3 shows the encrypted database on the cloud storage server.

Table 1. The database stored in the bank's application

Customer-id	Customer-first name	Address
C257846945	Muna	Mosul
C344795612	Mohammed	Baghdad
C559667102	Hammed	Mosul
C224789555	Ali	Basra
C7478945872	Khalid	Babul

2	1414795841673453	0067857878412112:	15951989389425504'	338537556142908105888889489148
3	1414795841673453	0067857878412112:	15951989389425504'	167724147567873038728945228836
4	1341507679240820	0763165154812574!	15951989389425504'	366000508608163319867483949900
5	1341507679240820	0763165154812574!	15951989389425504'	200375752527645824480601627553
6	0630263810432476	1807227423402023-	15951989389425504'	818644300135799317042869064750
7	0630263810432476	1807227423402023-	15951989389425504'	130659140102188133189693555150

Figure 3. Shows the encrypted and stored database inside the cloud

We measured the average time to encrypt the 1 KB database in the bank's application and the average time for re-encryption in the cloud. The preliminary results obtained are shown in Table 3 with minimum key size (128 bit) and maximum key size (2048 bit). The difference between the time taken to encrypt 1 KB of data and the time taken to encrypt 10 KB using paillier algorithm by different sized keys is shown in Figure 4. While Figure 5 will show that the key size is an important factor that affects time spent re-encryption and decryption using Paillier algorithm.

Table 3. The average time spent for encryption and re-encryption process

Size of key	Average time for encryption	Average time for re-encryption
128 bit	0.00397 second	0.002351 second
2048 bit	0.24147 second	0.005636 second

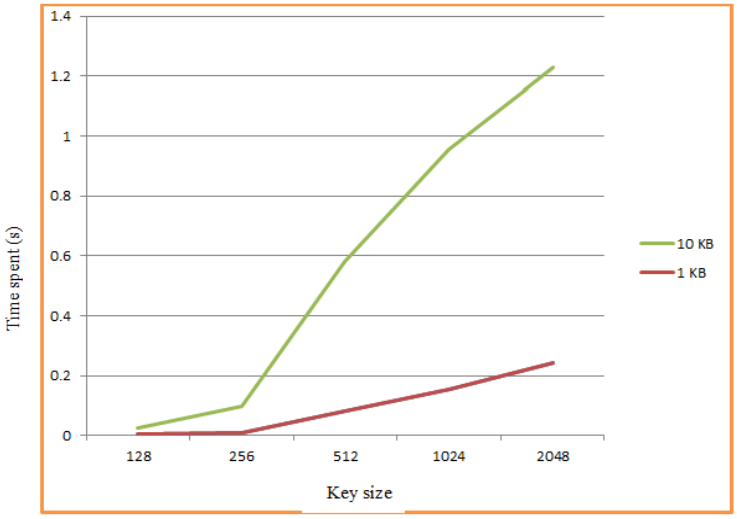

Figure 4. The average time spent for encryption using Paillier algorithm

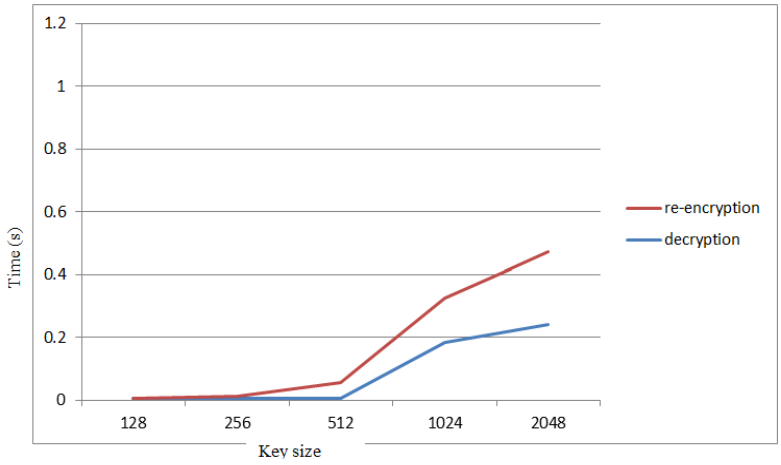

Figure 5. The average time spent for re-encryption and decryption using Paillier algorithm time

7. CONCLUSION

In spite of the increasing trend by companies, medical institutions and banks to use cloud computing technologies now fears are still present about the possibility of data being exposed to the risk of disclosure while storing them or performing mathematical operation on them, especially sensitive data such as account numbers or personal information that pertains to the customer such as address, phone numbers and others. Conventional encryption is not feasible in cloud computing. Because the bank will have to send its private key to the cloud to decrypt the data and perform addition or subtraction operations, so the data in the cloud is exposed to disclosure. In this paper we proposed a good way to encode bank data before sending it to the cloud using homomorphic encryption algorithm which is Paillier algorithm, data will be stored as encrypted data and any process that will take place inside the cloud will perform on encrypted data and the result that we get will also be encrypted and the customer or the party who owns the private key is alone who can decode and get the result. These operations will decrease the time and cost that required in the banking systems.

ACKNOWLEDGEMENTS

The researchers thank the Department of Computer Science, College of Computer Science and Mathematics, University of Mosul.

REFERENCES

[1] I. Jabbar and S. Najim, "Using fully homomorphic encryption to secure cloud computing," *Internet of Things and Cloud Computing*, vol. 4, no. 2, pp. 13-18, 2016.
[2] H. Takabi, James BD Joshi, and Gail-Joon Ahn., "Security and privacy challenges in cloud computing environments," *IEEE Security & Privacy*, vol. 8, no. 6, pp. 24-31, 2010.
[3] J. Sen, "Security and privacy issues in cloud computing," *Cloud Technology: Concepts, Methodologies, Tools, and Applications. IGI Global*, pp. 1585-1630, 2015.
[4] M. Tebaa, et al., "Hybrid homomorphic encryption method for protecting the privacy of banking data in the cloud," *International Journal of Security and Its Applications*, vol. 9, no. 6, pp. 61-70, 2015, doi: http://dx.doi.org/10.14257/ijsia.2015.9.6.07.
[5] S. N. Kumar and A. Vajpayee, "A survey on secure cloud: security and privacy in cloud computing," *American Journal of Systems and Software*, vol. 4, no. 1, pp. 14-26, 2016, doi: 10.12691/ajss-4-1-2.
[6] R. A. Hallman, et al., "Homomorphic Encryption for Secure Computation on Big Data," *Proceedings of the 3rd International Conference on Internet of Things, Big Data and Security (IoTBDS 2018)*, 2018, pp. 340-347.
[7] B. Schoenmakers and P. Tuyls, "Efficient binary conversion for Paillier encrypted values," *Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Berlin, Heidelberg*, 2006, pp. 522-537.
[8] C. Gentry, "Fully homomorphic encryption using ideal lattices," *Proceedings of the forty-first annual ACM symposium on Theory of computing*, 2009, pp. 169-178.
[9] Q. L. Huang, et al., "Secure and privacy-preserving DRM scheme using homomorphic encryption in cloud computing," *The Journal of China Universities of Posts and Telecommunications*, vol. 20, no. 6, pp. 88-95, 2013.
[10] Y. Zhang, et al., "A secure image retrieval method based on homomorphic encryption for cloud computing," *19th International Conference on Digital Signal Processing*, 2014, pp. 269-274.
[11] J. L. Raisaro, et al., "Feasibility of Homomorphic Encryption for Sharing I2B2 Aggregate-Level Data in the Cloud," *AMIA Summits on Translational Science Proceedings*, vol. 176, 2018.
[12] A. Elzamly, et al., "A New Conceptual Framework Modelling for Cloud Computing Risk Management in Banking Organizations," *International Journal of Grid and Distributed Computing*, vol. 9, no. 9, pp. 137-154, 2016.
[13] S. Rani and A. Gangal, "Security issues of banking adopting the application of cloud computing," *International Journal of Information Technology*, vol. 5, no. 2, pp. 243-246, 2012.
[14] M. D. Ryan, "Cloud computing security: The scientific challenge, and a survey of solutions," *Journal of Systems and Software*, vol. 86, no. 9, pp. 2263-2268, 2013.
[15] E. D. G. M. Someswar, "Security Techniques for Protecting Data in Cloud Computing," *Global Research Academy, Hyderabad, India*, 2014.
[16] M. Tebaa, Saïd El Hajji, and Abdellatif El Ghazi., "Homomorphic encryption applied to the cloud computing security," *Proceedings of the World Congress on Engineering*, vol. 1, no. 2012, pp. 4-6, 2012.
[17] R. L. Rivest, Len Adleman, and Michael L. Dertouzos, "On data banks and privacy homomorphisms," *Foundations of secure computation*, vol. 4, no. 11, pp. 169-1804, 1978.
[18] M. Ogburn, et al., "Homomorphic encryption," *Procedia Computer Science*, vol. 20, no. 1, pp. 502-509, 2013.
[19] M. Alkharji, Hang Liu, and Washington CUA, "Homomorphic encryption algorithms and schemes for secure computations in the cloud," *Proceedings of International Conference on Secure Computing and Technology*, 2016.
[20] I. Ahmad and A. Khandekar, "Homomorphic encryption method applied to cloud computing," *International Journal of Information and Computation Technology*, vol. 4, no. 15, pp. 1519-1530, 2014.
[21] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," *International conference on the theory and applications of cryptographic techniques. Springer, Berlin, Heidelberg*, 1999, pp. 1-15.
[22] P.V. Parmar, et al., "Survey of various homomorphic encryption algorithms and schemes," *International Journal of Computer Applications*, vol. 91, no. 8, pp. 26-32, 2014.
[23] K. K. Chauhan, et al., "Homomorphic encryption for data security in cloud computing," *International conference on information technology (ICIT)*, 2015, pp. 206-209.
[24] N. Saputro and K. Akkaya, "Performance evaluation of smart grid data aggregation via homomorphic encryption," *IEEE Wireless Communications and Networking Conference (WCNC)*, 2012, pp. 2945-2950.
[25] P. Y Ryan, "Prêt à Voter with Paillier encryption," *Mathematical and Computer Modelling*, vol. 48, no. 9-10, pp. 1646-1662, 2008.
[26] H. Y. Lin and W. G. Tzeng, "An efficient solution to the millionaires' problem based on homomorphic encryption," *International Conference on Applied Cryptography and Network Security. Springer, Berlin, Heidelberg*, 2005, pp. 456-466.

BIOGRAPHIES OF AUTHORS

Muna Mohammed Saeed Altaee She is a faculty member at the Institute of Fine Arts, Mosul, affiliated to the Iraqi Ministry of Education, holds a Bachelor's degree in Computer Science from the Department of Computer Science, University of Mosul, Iraq in 2003. Student in the Department of Computer Science, University of Mosul Iraq to obtain a master's degree in 2020 Its domain will address current computer and cloud computing research.

Mafaz Alanezi She is a faculty member at the Department of Computer Science, University of Mosul, IRAQ. She obtained her Ph.D. degree in Computer Science in the field of Computer and Network Security from University of Mosul, Iraq in 2012. Her M.Sc. degree was also in Computer Science in the field of Image Processing from the University of Mosul, Iraq in 2003. Her current area of research deals with Computer and Network Security, Artificial Intelligence, and cloud computing.