https://doi.org/10.33261/jaaru.2019.26.4.008 *Association of Arab Universities Journal of Engineering Sciences (2019) 26(4): 65–72*

Association of Arab Universities
Journal of Engineering Sciences
مجلة اتحاد الجامعات العربية للدراسات والبحوث الهندسية

كلية الهندسة - جامعة بغداد

جمعية كليات الهندسة
اعضاء اتحاد الجامعات العربية

Towards Query Processing Over Homomorphic Encrypted Data in Cloud

Lina Samir Malouf [1]

[1] *AlBaath University, Homs, Syria, leena_m84@hotmail.com*

**Corresponding author: Lina Samir Malouf, email: leena_m84@hotmail.com*

Published online: 31 December 2019

Abstract— With data growth very fast, the need for data storage and management in the cloud in a secure way is rapidly increasing, leading developers to find secure data management solutions through new technologies. One of the most advanced technologies at present is cloud computing technology that functions as an online service. Cloud computing technology relies on an external provider to provide online demand services. On the other hand, this technology is pay-for-use technology which means that the user must pay for each service provided by the provider. When we have a look back at the literature, we can find that regular database management systems with query processing specifications do not meet the requirements in cloud computing. This paper focuses on homogeneous coding, which is used primarily for knowledge security within the cloud. Homomorphic encryption has been clarified because of encryption technology in which specific operations can be managed on encrypted data information.

Keywords— Cloud database, Query processing, Encryption, Homomorphic encryption.

1. Introduction

As NIST(National Institute of Standards and Technology) official definition, it can be defined cloud computing as a model for enabling ondemand access to a shared pool of computing resources can be managed with minimal management efforts within a third party service provider [5]. Many service providers offer their infrastructures in the cloud as services but the main concern for many institutions or ordinary users is the security of their data which can be stored in databases in the cloud. One of the main guarantee to keep cloud data safe is to work with encryption methods over the cloud data [4]. We will focus in this study on the homomorphic encryption, which is divided into two types of encryption, partially homomorphic encryption within which addition or multiplication operations may be administered on the conventional cipher text; and fully homomorphic encryption, which permits impulsive computation on the ciphertext in a ring. Not with standing, the privacy and data security is a major concern that hinders the cloud database services acceptance according to their business requirements. When you publish a database in a public database service, the service provider has full physical control of the database. Data in the database may be accessed incorrectly by the service provider in an accidental or intentional manner, or by attackers who threaten database servers. Here, If the issue of cloud privacy and security is not satisfactorily addressed, the cloud services may not be fully exploited [3]. Cloud database management systems nowadays deal with huge amounts of data. The service provider of the cloud database deals with many barriers with its job of providing many services and we can perform queries with encrypted data. Coordinated encryption plays an important role in cloud computing because corporate encrypted information remains in a public cloud, thus benefiting from cloud provider services [6]. A number of theoretical and practical solutions have been proposed to address requests under different scenarios. Query processing of relational data has been widely studied over the last decade. The most important challenge in cloud computing nowadays is that Current cloud offerings are essentially public (rather than private) networks, exposing the system to more attacks. There are also requirements for audit ability. The solution is to make cloud computing environment secure by using encrypted storage, Virtual Local Area Networks, and network middle boxes. The most important feature of cloud database management system is data encryption, which is used to prevent unauthorized access to the sensitive data, and the cloud application must not be able to decrypt data directly before it is accessed[8].

Scalability can be added as an extra layer outside the virtual guest system, providing more secure facilities than those in the applications themselves, centralizing program responsibilities for confidentiality, and reviewing capability in the single logic layer [4].

2. Data Encryption:

Users and companies are concerned about their data when data are stored and managed in a Cloud or outsourcing servers. The main solution is to encrypt data before they send data to a Cloud or outsourcing servers to prevent the server or provider from accessing the data that are stored on its own machines [2][7]. We can perform data encryption by using symmetric or asymmetric encryption systems, while many suggestions rely on symmetric encryption. When a client requests any information from an encrypted table, the client must request the whole table because the quadrature cannot be filtered in the encoded table representation. This will require an excessive workload and require encryption and decryption of the data and to minimize this workload, we must supply the original database table with additional indexing information, and indexes are usually stored with encrypted tables on the cloud [3]. Encryption can be defined as a flag that is used to secure sensitive data. The encryption system consists of the following sections:

1-Plaintext: which refers to the original form of data, data to be protected during transport and storage. Encrypted text: is an illegible form of plain text after encryption.

2-Encryption algorithm: used to convert plain text to encrypted text, with a mathematical process.

3-Decoding algorithm: Performs an inverse operation of the encryption algorithm, and converts encrypted text to plain text [3].

4-Encryption key: A value used by the sender with an algorithm to convert plain text to encrypted text.

5-Key decryption: A value used by the receiver with the algorithm to convert encrypted text to plain text.

Among all services, DaaS (database as a service) is the most sought-after service for cloud users, exchanging, storing and modifying their information online. As this service is processed over the Internet, there is an opportunity to snoop on data and penetrate it. Very confidential data likeonline banking, credit card numbers and passwords. are easily available to the intruder in the absence of security over the network [10]. Some security models are designed to overcome this type of attack. Many security algorithms rely on encryption standards, which means that the original text is encrypted to a text form that is encrypted by the sender and transmitted over any network when receiving encrypted text from the sender by the recipient, and decrypts and reads the body text as shown in Fig1.

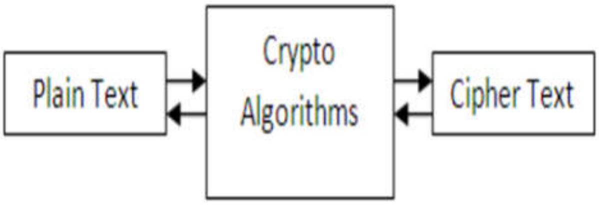


Figure 1: cryptography algorithm

Encoding nowadays is a combination of three types of algorithms.

They are

(1) asymmetric key.

(2) symmetric key.

(3) hash.

An encryption key which is shared between sender and receiver that can encrypt and decrypt the data known as the shared / symmetric key [9]. Both symmetric keys and asymmetric key algorithms can be used to encrypt data in cloud storage. The cloud volume contains a large set of databases, and for a large, asymmetric database performance, the performance of the key algorithm is slower when compared to symmetric key algorithms [11].

3. Processing Query on Encrypted Database Table:

In the meantime, data is protected on the server [8]. This method also avoids decrypting unwanted data by decrypting data requested by the user only. This technology can help us process a query on a cloud or server with no decryption on the data, and the data is decrypted only at the client's side to protect data from leakage. When data is indexed and broken by any technique, the same method must be used to index and divide data in a query before sending the query to the cloud or server [1]. Typically, when a user submits a query, it assigns the query to a server query that runs on the encrypted table in the server location. Once the server executes the query, the server returns a set of encrypted records to the client that decrypts it and selects only the correct records, and gives them to the user [13].

3.1 *Review of the literature*

In 1998, Blaze, Bleumer Strauss (BBS) [16] proposed the idea of "atomic proxy encryption", which used proxy to convert texts into encrypted texts without seeing plain text. Then Dodis and Ivan [17] performed the encryption with the proxy by separating the user's secret key into two components, they proposed a one-way encryption factor for ElGamal, RSA IBE system, but the disadvantage is that the delegate needs to store additional secrets to the service provider decryption, it is difficult for the service provider to manage keys. This is followed by some pairing based on proxy re-encoding schemes, such as [18,19], have been proposed and used. In 1978 Ronald Reist, Leonard Adelman and Michael Dertouzos first proposed the concept harmonized encryption. Since then, progress has been minimal for 30 years. Cryptographic Encryption System Goldwasser and Silvio Micali suggested in 1982 he an encryption system which has been reached a remarkable level of safety, it was homomorphic added encryption, but can only encrypt one bit. at the same concept in 1999 as suggested by Pascal Paillier improved the security encryption system and was also an additive harmonized encryption. The Paillier encryption system is an asymmetric algorithm for public key encryption. It is believed that the problem of calculating the residues of rank n is arithmetically difficult. The assumption of the basic composite residue is the hypothesis of ascites on which this system is based. Several years later, in 2005, Dan Boneh, Eu-Jin Goh and Kobi Nissim invented the system encryptable security encryption, which we can implement unlimited number of plug-ins but only one batting. Craig Gentry of IBM proposed the first encryption system named "completely homomorphic" that arbitrarily reside the number of plug-ins and multipliers thus calculated that can calculate any type of functionality on the encrypted data. The problem of building such a scheme was first proposed within a year as RSA development. RSA is an algorithm used by modern computers to encrypt and decrypt messages. It is an asymmetric cryptographic algorithm, Asymmetric means that there are two different keys. This is also called public key encryption, because one of the keys can be given to anyone. You must keep your other key. The solution proved more elusive. For over 30 years, It was not clear whether the homomorphic encryption is so complete or not. During this period, the best result was delivered by Boneh- Goh-Nissim cryptosystem that supports the evaluation of an unlimited number of add-ons but at most multiplication. The first homomorphic encryption system was just as advertised by IBM on June 25, 2009. Its scheme supports evaluation of arbitrary depth circuits. Construction begins from somewhat homomorphic coding scheme using perfect networks that are limited to low-grade polynomials on the encrypted data. Limited because each text is encrypted in some sense noisy, this noise grows as one adds up multiply encrypted texts, so eventually it makes noise, which makes the encrypted text incomprehensible.) Then shows how to modify this system to make it unobtrusive in particular, shows that by modifying its form to some extent schema bit, it can actually evaluate your own decryption circle self-reference property. Through repeated self-inclusion. In special case of the ideal stratification based on the nobility is based on a fairly homogeneous scheme, This procedure effectively bootstrapping the Encryption Update the cipher text by reducing associated noise so that it can be used then in more plug-ins and complications without resulting in an indecipherable cipher text. Gentry based the security of his scheme on the assumed hardness of two problems: certain worst-case problems over ideal lattices and the sparse (or low-weight) subset sum problem.

Encryption is defined as a form that refers to the ability of some messages to be encoded, to create a cryptography for a value associated with the original messages. In other words, this means that for encoding k messages (mm1, ..., mmk) with effective computable functions. In fact, some of the proposed cryptographic systems are homogenous by some of the algebraic processes such as multiplication or addition. An encryption process named Rothbum defined it as: it is a method where we can get the encryption key clearly defined and known, on the other hand the decryption key connected to this encryption key is kept secret [12]. There was little progress in a completely homomorphic concept Homomorphic additive encryption has a noticeable impact on security, but can only encrypt bit by bit. The use of homomorphic encryption makes us enable to compute encrypted data. So, it is very important to use homomorphic encryption to secure communication protocols. The current schema of homomorphic encryption with public key have some properties like: Encrypt (PKey, x1) and Encrypt (PKey, x2), anyone can calculate either (PKey, x1 + x2), or Encrypt (PKey,x1*x2) product, but not both.

3.2 *The encryption with private key:*

The idea of encryption was used in the ancient world widely specially to deliver messages during wars. It was used in a simple way, one of these algorithms used the shifting methodology for example we can shift each character for a pre-defined number of shifts for example the specific number is 3, then it means that A will be replaced by C, B will be replaced by E and so on.

ABCDEFGHIJKLMNOPQRSTUVWXYZ

CDEFGHIJKLMNOPQRSTUVWXYZABC

Many other similar character methods were used in small communications [14]. The encryption is very old, it was used in very ancient decades, so we can divide it into two categories.

1- Classical age.

2- Computer age.

In the classical age there were no electronic devices or computers to solve the encryption problem so people used the written methods to discover the meanings of letters. After the development of machines, the encoder made life very complicated and it was very easy to break all traditional encryption operations used before when it is often called key-based technologies. Many practices were done in the encryption field to use a key word, ROT13 starts to use the fixed key 13 to encrypt characters which considered as the most useful technique and it was very popular algorithm at the early computer decades and anyone who wanted to use the ROT13 algorithm schema, have to use the same key at the sender and receiver sides to encrypt and decrypt the code. This key is called the secret key.

As an example in ROT13 encryption, the characters are shifted by 13 places, for example "a" will replaced by "n" and "m" becomes "z" and continues to be addressed if necessary. Play only English alphabet. This work may be the same as the person who teaches the grammar or the grammar of the advanced nation. In addition, this feature is called "account expansion" and the code for giving and skipping encoding [4]. This work works on the following

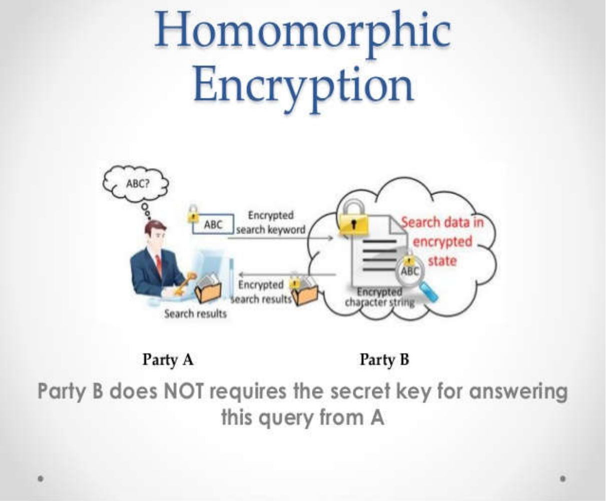
ABCDEFGHIJKLM ↔ abcdefghijklm

NOPQRSTUVWXYZ ↔ nopqrstuvwxyz

The problem we are met again if someone stolen your data, in this case, it is very easy to decrypt it where it is not a reasonable encryption method although it is defined as a secret key encryption scheme [15]. If we observe closely ROT13 is partially similar in particular it can be considered a partially homomorphic method.

4. Homomorphic Encryption:

The basic concept of cryptographic homomorphic encryption can be defined as the information protection from unauthorized users by supporting the process of computing on the data encrypted. The main goal of this encryption is to raise the user information trust while storing and transporting the information, it shores multiple computations and search techniques on cipher data without decrypting the information and use it at the time of processing. We can find many advantages of the homomorphic way for example we don't have to decrypt data in the time of processing, which means a lower processing time for huge amount of data, and on the other hand we don't have to share and exchange the keys as it appears in **Fig.2**.



Homomorphic Encryption

Party B does NOT requires the secret key for answering this query from A

Figure 2: Homomorphic Encryption[22]

We can illustrate homomorphic encryption in more detailed example as follows: Let's assume that we have a text T to be encrypted, then the cipher text will be C=Enc(T) with the encryption key kk; Now we can get the original T from C with the encryption key kk, we assume that we have two texts : C1=Enc(T1) and C2=Enc(T2), We can do homomorphic computations on the cipher texts for example : additive homomorphic encryption C1+C2 or multiple homomorphic encryption C1*C2, then decryption results are M1+M2 and M1*M2 respectively.

4.1 *Fully homomorphic encryption:*

Principally, fully homomorphic encryption permits for capricious computations on encrypted information. Computing on encrypted information implies that if a user includes an operator f and need to get f (a1, a2,…,an) for a few inputs a1, a2,…,an, it's doable to instead cypher on encryptions of those inputs, b1, b2, …..,bn, getting a result that decrypts to f (a1, a2,…,an). Process of fully harmonic encryption This 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[14]. 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 Fully homomorphic encryption 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 +). 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.

4.2 *Fully Homomorphic Encoding On Cloud:*

It is very useful to apply the homomorphic encryption in the field of cloud computing; additional typically, outsourcing computations to confidential information to a cloud server is possible, while maintaining the key that will rewrite the computation results. We can look at cloud computing as a model that enables access to the network everywhere and a common set of computing resources like servers, storage, networks, applications and services that can be quickly delivered and deployed with minimal administrative service provider interaction and efforts. The data storage requirements have been changed with the dramatic growth in digital data. On the other hand, that improvement cloud computing leads the developers to use databases in the cloud which raise the need to find a useful way to manage data in the cloud. There are many examples of cloud databases like dropbox and icloud. Where the data can be stored in an encrypted format and computing can be done over this encrypted data. One of the common cloud characteristic is the multitenant built in and data centers are shared by multiple users. Cloud service providers help the end users to store data in the cloud database, but the most important challenge for the providers is the data security.

Indexing plays an important role in the processing of queries. Here are the most important indexing techniques in the literature: Map reduces value pair - Batch index (inverse index), Atom Latin Pig, Tuple, Local Map, Mapreduce, Batch, Interactive, Inverted Index. Tuple Bit indexing map, HadoopDB Collection is multidimensional and Bricks SQLMR in a simple style [21].

5. Proposed Method:

There is a real need to use an efficient encryption algorithm that computes encrypted data without decryption in order to enforce data security. Homomorphic encryption is an encryption system that can solve this problem because it is able to perform calculations on encrypted data without decrypting. Encrypted results can only be decrypted by the client requesting and decrypting the same results, such as performing the same calculations on the original data. Fully homomorphic encryption is a very important concept in the field of cloud computing security that allows companies and organizations to store their cloud data and take advantage of the cloud provider's analytics without providing key encryption for cloud service providers. The cloud database in the proposed method consists of horizontal rows where data is stored at these rows are distributed among cloud nodes and every node have a specific capacity. Let’s assume that the node capacity is n rows, moreover the data will be partitioned and stored as encrypted data. An encryption key is used to encrypt the data and it may use to encrypt other types of data like user signature to guarantee the data integrity. In the paper (a framework for searching encrypted databases by Pedro Geraldo MR Elvis, Diego Aranha), the purpose of its proposed framework is to develop a database model capable of storing encrypted records and applying relational relational types without knowledge. Any encryption keys or need to be decrypted. We recommend that performance and security be a trade-off, although it ignores encryption at any time for security reasons. This application overrides the SQL window databases. CryptDB is a software layer that provides the ability to store data in a remote database. Query more without revealing sensitive database management system information (database management systems). Provides a proxy layer responsible for encrypting and modifying queries for the database and decrypt the results [Popa et al. 2011]. The context in which CryptDB stands is a modular architecture for database-supported applications, consisting of a database management systems server and a separate application server. To query a database, The evaluation function is created by the application and processed by the agent before sending it to database management systems server. The user interacts exclusively with the application server and administrator To keep your password secret. This password is provided when you log on to the proxy (via the application) which derives all the keys required to interact with the database. When the user logs on abroad, the proxy is expected to delete its keys. While in the proposed method, a communication protocol to control communication between the application or the end-user and the encrypted database. A proxy query is used for this connection. When a request is sent from the user, the request will be converted by the proxy to an encrypted query form. The main contributions of the working paper can be summarized as follows: We design a cloud computing system to maintain complete privacy with flexible access control. For data security, not all initial data and processed data are detected, which is only known to the specific person. The encryption will be done using symmetric homomorphic algorithm in multiple steps:

5.1 *Key Generation:*

Define cryptographic parameters:

rr is a random number of R-bit

i is a large integer Q-bit constant.

KeyGen (q): The q key is a random integer for P-bit.

q is the secret key.

q is the secret key assigned by the server once the client log in, the server will specify a key generation seed for each user and the secret key is generated at the end user by this seed which means the server have no idea about the secret key. The secret key can be taken by the end user any time and this procedure is repeated.

5.2 *Encryption:*

For plain text n:Encryption (q, n): To encrypt a unit, m bit between {0, 1}

d=q+ 2rr + n where d is the encoded text

5.3 *Decryption:*

n = (d mod q) mod 2 where q is greater than 2rr + n so (d mod q) = 2rr + n

Decryption (q,d): Output (d mod q) mod 2 = (2rr + n) mod 2 = n

Symmetric: For two encrypted text

d1 = i1q + 2rr1 + n1

d2 = i2q + 2rr2 + n2

Statistics - Count: d1 + d 2 = (i1 + i2) q + 2 (rr1 + rr2) + n1 + n2

So if 2 (rr1 + rr2) + n1 + n2 is less than q

Then (d1 + d2) mod q = 2 (rr1 + rr2) + n1 + n2

And d1 * d2 = [i1 * i2q + (2rr1 + n1) + (2rr2 + n2)] q + 2 (2rr1 rr2 + rr1n1 + rr2 n1) + n1 n2

Even if 2 (2rr1 rr2 + rr1n1 + rr2 n1) + n1 n2 <<q

Then (d1 * d2) mod q = 2 (2rr1 rr2 + rr1n1 + rr2 n1) + n1 n2

The main processes that can be done by the end user on cloud database are querying and updating, while the data stored at the cloud database is encrypted. We need to find a communication protocol to control the communication between the application or end user and the encrypted database. A query proxy is used for this communication. When a request sent from the end user, the request will be translated by the proxy into an encrypted query form. This encrypted query has the ability to be executed on encrypted data in the cloud database. After the query is processed, the result has to be returned to the end user. Another step will be taken by the query proxy; it will decrypt the query result then it will forward the result to the end user. To translate the query by the query proxy, some aspects are taken into account like the schema of the database and the keys used in the encryption process. In the cloud database with encrypted schema, each field has a value in encryption form and an index for this value. The proxy will receive the query and get the index of the encrypted value stored in the cloud database and then return the result according to the index according to Fig.3.

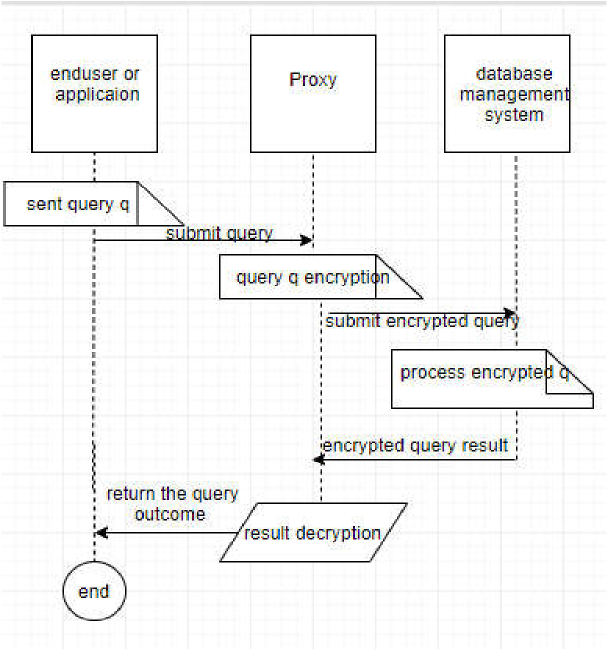


Figure 3: Query processing in encrypted case

So, when a request sent to the cloud server by the end user, the server will deal and operate with the request without even knowing its content, With this structure the data will be encrypted which mean a cloud service with more security options.

5.4 *Implementation:*

The proposed method was implemented using Python programming language. For the tests, a laptop with a Core i7 3.5GHz CPU and 8GB of Ram memory was used. The obtained times are the arithmetic average of 10 executions and the best results obtained demonstrated in **Teble.1**. we use four levels of security inspired by levels of : "very small", "small", "medium" and "large" " , the interview 42, 52, 62 and 72 bits of security, respectively.

Table.1

dataset	Key generation	Encryptio n	Decryption
Very small	0.55 s	0.0088 s	0.0002 s
small	11 s	0.0245 s	0.00045 s
medium	1 min 5 s	0.456 s	0.0014 s
large	10 min 0 s	2.440 s	0.0378 s

6. Using Homomorphic Encryption in multi tenant architecture:

Cloud computing benefits by providing service to multiple tenants on the same physical device through virtualization at the same time as security for tenant data provides a challenge. Other security risks in cloud computing are multi-tenants sharing involving resources, storage, memory, services and applications with other tenants. This means that customer data can be stored in the same physical device that can be exploited by opponents to launch various attacks such as data breaches / account, flood attack, and so on. The proposed method may be used as data storage method for SaaS or MTA-based platforms that allow for the expansion and modeling of high efficiency data management structures such as databases. In the MTA, each tenant on the cloud may assign a separate database for security reasons, but it is not cost-effective, so the individual database is shared. Sharing the database will result in unauthorized access to tenant data by another tenant, which is a serious security breach. In order to address this problem, a data encryption technology that allows each record to be encoded twice before being stored in the tenant-specific section by the public will be proposed. Known to the client and the cloud provider using the built-in encryption technology. In this technique, tenant storage data is encrypted in the database section twice first by the same tenant who will encrypt the registry and the second time by the cloud service provider The decryption method will be implemented by special keys for the tenant. The record to be stored on the Cloud Service Provider is also not known ER is sent to it. Key information is stored in the metadata table in each section of each tenant, which must be secured by the CSP, and therefore Tenancy data is secured in the cloud environment. Each department belongs to the tenant by different pairs of keys. In the plural model, many user data and resources are in the same computing cloud, controlled and marked by the use of tags to uniquely identify the resources owned by an individual user. In a typical multilingual mode, users are tenants and are provided with a level of control to customize and customize software and hardware to suit their specific needs. However, multiple pluralism offers a unique set of security risks, which have not been fully recognized as a serious problem by policy makers and cloud service providers. The audio security structure must ensure that the tenant does not have access to other leased resources, such as the virtual machine (VM), network bandwidth, and storage. Each tenant must be securely separated by techniques such as access control, VLAN fragmentation, and virtual storage controllers.

7. Conclusion and future work:

Cloud computing provides low maintenance costs, and multiple tenant features. While delivering data to the cloud protection problem Lifts. Here we introduced a homogeneous encryption technology to provide better security as compared to Traditional encryption system. It enables the cloud to perform our calculations on encrypted data stored on the cloud It provides a result in an encrypted form that when decrypted will be the same after performing operations on plain text. In the future we will try to improve the encryption of homogeneous performance in terms of increasing data size And maintenance of keys. Work in the implementation of homomorphic encryption is underway. Data confidentiality is a fundamental challenge to shared computing infrastructure. Cloud computing based on fully homogeneous encryption is a new concept of security that enables the provision of account results to encrypted data without knowing the initial inputs to which computations have been made that respect thedata security. The proposed paradigm applied Fully Homomorphic Encryption to cloud data and the calculations and operations was made on encrypted data to enhance the quality of similar cryptographic algorithms.

References:

[1] 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.

[2] C. Fontaine , F. Galand, A survey of homomorphic encryption for nonspecialists, EURASIP Journal on Information Security, 2007,p.1-15, January 2007

[3] Jerry Archer Alan Boehme Dave Cullinane Paul Kurtz Nils Puhlmann Jim Reavis. Post-Quantum Cryptography, chapter Lattice-based Cryptography. Springer, 2008Cloud Security Alliance, Top Threats To Cloud Computing V1.0, http://www.cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf.Cloud Security Alliance, Security Guidance for Critical Areas of Focus in Cloud Computing V2.1, http://www.cloudsecurityalliance.org/csaguide.pdf.

[4] D. Liu and S. Wang. Programmable order preserving secureindex for encrypted database query. In *Proceedings of the 5thIEEE International Conference on Cloud Computing*, pages502–509, 2012.

[5] D. Liu. Homomorphic encryption for database querying.Australian Provisional Patent 2012902653 (filed by CSIRO), 2012.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in Advances in Cryptology (EUROCRYPT"99), vol. 1592 of Lecture Notes in Computer Science, pp. 223–238, Springer, New York, NY, USA, 1999.

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

[12] T. Ge and S. Zdonik. Answering aggregation queries in asecure system model. In *the 33rd international conference onVerylarge data bases*, pages 519–530, 2007.

[13] X.Wang,"One-round secure fair meeting location determination based on homomorphic encryption," Information Sciences, 372(2016), pp. 758–772.Stuntz, "What is Homomorphic Encryption, and Why Should I Care?,"March 18, 2010.

[14] Hossein Shafagh, Anwar Hithnawi, Andreas Dröscher, Simon Duquennoy, and Wen Hu. 2015. Talos: Encrypted Query Processing for the Internet of Things. In ACM Conference on Embedded Networked Sensor Systems (SenSys).

[15] Z. Brakerski and V.Vaikuntanathan.

Fully homomorphic encryption from ringlwe and security for key dependent messages. In *Proceedings of the 31st annual conference onAdvances in cryptology* Berlin springer, CRYPTO’11,pages505-524,2011.

[16] Blaze, M., Bleumer, G., & Strauss, M. (1998). Divertible protocols and atomic proxy cryptography. Lecture Notes in Computer Science, 1403, 127-144.

[17] Ivan, A. A., & Dodis, Y. (2003). Proxy Cryptography Revisited. Network and Distributed System Security Symposium, NDSS 2003, San Diego, California, Usa. DBLP.

[18] Ateniese, G., Fu, K., Green, M., & Hohenberger, S. (2006). Improved proxy re-encryption schemes with applications to secure distributed storage. Acm Transactions on Information & System Security, 9(1), 1-30.

[19] Ateniese, G., Benson, K., & Hohenberger, S. (2009). Key-Private Proxy Re-encryption. Topics in Cryptology - CT-RSA 2009, The Cryptographers' Track at the RSA Conference 2009, San Francisco, CA, USA, April 20-24, 2009. Proceedings (Vol.5473, pp.279-294). DBLP.

[20]A framework for searching encrypted databases Pedro G. M. R. Alves, Diego F. Aranha 2012.

[21]Inverted Indexing In Big Data Using Hadoop Multiple Node Cluster, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 4, No. 11, 2013

[22]https://theintelligenceofinformation.wordpress.com/2017/03/01/homomorphic-encryption-statistical-machine-learning-and-r-software-package.

معالجة الاستعلامات على البيانات المشفرة باستخدام التشفير المتماثل في السحابة

لينة سمير معلوف [1]

[1] *جامعة البعث، حمص، سورية، leena_m84@hotmail.com*

* *الباحث الممثل لينة سمير معلوف، البريد الالكتروني: leena_m84@hotmail.com*

نشر في: 31 كانون الأول 2019

الخلاصة – الحوسبة السحابية هي تقنية تعنى بعمليات تقنية المعلومات التي تقوم بها أي جهة في "السحابة"، ولها مميزاتها الكثيرة مثل تكلفتها المادية، وكفاءتها، ومرونتها، وسعتها غير المحدودة، وتوفرها بشكل كبير ومستمر بحسب الطلب. هذه المميزات وغيرها تدفع الجهات المختلفة للاستفادة من تقنية الحوسبة السحابية. تعد الحوسبة السحابية هي التكنولوجيا الأسرع نمواً، حيث تقدم خدمات متنوعة عبر الإنترنت. يمكن أن تخدم العديد من المرافق لرجال الأعمال مثل الموارد والبنية التحتية، من خلال دفع المبلغ على أساس الطلب عبر الشبكة مع إمكانية تغيير المتطلبات حسب الحاجة. يمكن لهذه التقنية تلبية جميع المتطلبات في مجال تكنولوجيا المعلومات في أي وقت، كما يمكنها أن تخدم معظم الأجهزة والبرامج اللازمة للشركات من أجل تخزين، وإنشاء، وإدارة، وتشغيل تطبيقات المستهلك على السحابة لأنها توفر الموارد كخدمة للمستهلكين المتعددين عن طريق المحاكاة الافتراضية. من أهم المخاطر التي تواجه مستخدمي السحابة هي أمن البيانات ولذا فإن حماية البيانات الفعالة والتشفير القوي في السحابة أمر ممكن ومتاح من خلال عدد من الحلول السحابية. تركز هذه الورقة على معالجة الاستعلامات عبر البيانات المشفرة باستخدام التشفير المتماثل الذي يستخدم أساسًا لأمن المعرفة داخل السحابة.

الكلمات الرئيسية – قواعد البيانات، الحوسبة السحابية، معالجة الاستعلامات، التشفير، التشفير المتماثل.