Data Security Using Homomorphic Encryption for Cloud Based Medical Record Management System

Mustafa Abdalrassual Jassim

College of Engineering, AL Muthanna University, Iraq

*Corresponding author: Mustafa944@yahoo.com

Received:20/11/2017, Accepted:3/4/2018, published: 29/04 /2018

DOI: 10.52113/2/05.01.2018/44-51

Abstract: Homomorphic encryption technique is latest development in the field of data encryption. The replacement algorithm is taking part in this technique. In this paper, this technique is used for security of patient's medical record on cloud. Proposed technique besides data security, stores and retrieves cloud data is very fast. Paper proposes a model and encryption techniques with analysis before conclusion.

Keyword: Homomorphic, Encryption, Decryption, Medical, Data Centres, Cloud

1. Introduction

The world is facing problems, such as the growing chronic diseases, and the increasing medical expenses. Blending the latest information technology into the healthcare system will greatly mitigate the problems [1]. Equitable access to health services is one of the health justice criteria. E-health can sometimes be helpful in this regard. This study is aimed to find the use of cloud computing services across health system.[2] Cloud Computing is defined as a technology which uses the internet and central servers, Cloud services provide the necessary infrastructure at lower cost and better quality. Cloud computing when used in Healthcare sector reduces the cost of storing, processing and updating with improved efficiency and quality [3] facilitated access to the E-health services and big data in health systems are the main features of exploiting cloud computing services in health systems [2]. Cloud computing is a wide and diverse phenomenon. Allows users to store and use a large amount of data in the future. Data security, privacy, confidentiality and authentication must be maintained using Homomorphic Encryption, which provides confidentiality of data through encryption and decryption.[4,5]Data security is one of the most important things in cloud computing to avoid attacks on health care data. Therefore, encryption must be high quality and provide high levels of protection without compromising the performance of the network. It is a way to provide security to protect customers and to encrypt data in the health care cloud.[4,6] Without loss of generality, it focuses particularly on patient data security for used applications and resulting improved diagnostic capabilities. The system can be applied to any sensor that has similar capabilities with a background application that has similar properties. In this system, leaving only the responsibility of the graphical user interface of the mobile terminal device while accessing the monitoring results. The aim of this system is to push the entire workload into the cloud, allowing thin devices to be used as a GUI acquisition sensor. This will enable real cloud computing by turning the end nodes of this system into simple devices, while leaving the system core in the cloud. Moreover, since almost any information is stored on end devices, data privacy concerns" [7] Because of the loss of patient data or mobile device by the patient or the doctors are minimized, if not eliminated. [8]For the best knowledge, this is the first type of technology that focuses on the application of Homomorphic Encryption to fully monitor the long term patient. Our contributions to this project are as follows. It is proposed to be a cloud-based medical application where the contract end is simple and disposable. The core of the application resides accurately in the cloud, allowing flexibility for the hospital in its data center strategies. The following is the formulation of the complete Homomorphic Encryption as the core of this idea. The challenges are known to make this possible for a particular application based on the mobile device. The report is structured as follows. In the next section, fully Homomorphic Encryption makes basic information about the patient safe. In the next section, the proposed system presents in detail and identifies the challenges facing different parts of it, followed by a conclusion and our proposals for future action.

2. Proposed Model

The proposed model is shown in figure 1, 2, there are three processes that take place in whole working of the model. This goes as follows.

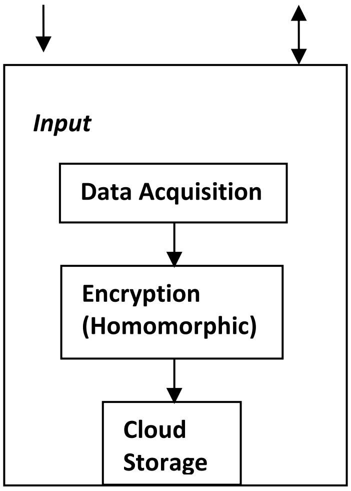

Fig. (1): Input module.

1. The data acquisition proves is the first module of the project. This is the data entry module is the patient's data record. The patient is givens a username & password, which is required for the patient to login to the cloud server. That represents the topmost level to the data security. The data entry to the cloud server can be made by the doctor. This process will make the doctor able to download the previous

history of the patient which was uploaded by all other doctors who checked the patient and prescribed the medicines, if will also have all the reports uploaded time to time by the patient to get exact history of the patient.

2. The data acquisition has some backend processes associated with the project where the username & password is first checked for authenticity checking and then the related data is fetched. This data is in encrypted format, this data is then decrypted. The data encryption and decryption is using Homomorphic technique.

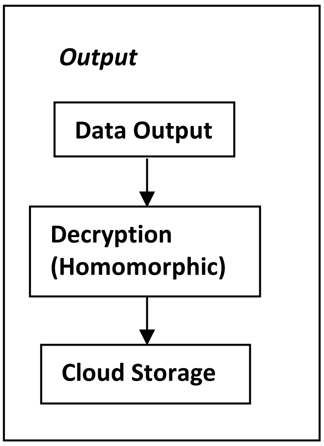

Fig. (2): Output module.

3. The data is then updated by the doctor or medical consultant and saved to the cloud server figure (3). Again while saving to the cloud server, the data is first Homomorphic encrypted and then it is stored or saved. This process prevents the data and make it secure.

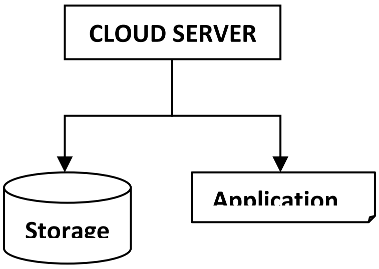

Fig. (3): Cloud Server.

4. To get this process in working, we are needed to have four modules. These modules together will make this system workable. This project will only be working if it has all five modules. For demonstration, we can take standalone system server instead of cloud server.

5. First among the modules those come in to picture is administration module. This module gives all the rights to the administrator of the system. Here administrator can assign username and password of the patient to the users that are patients. In case of loss of username or password this module helps to recover. This module also looks into the data security and working of encryption and decryption modules. It also looks in the services provided by the cloud server

6. The next module is the patient module which behaves like a user interface for data entry and retrieval process. This consist of a database which will store data like username, password, patients details, prescriptions by different doctors, some other reports and relevant data related to the patient.

7. The next module is of data encryption which encrypts the data before storing to the

cloud server. This module can be explained in detail with the help of detail flowchart.

8. The next module is of data decryption which decrypts the data before being serving to the patient. This module can be explained in detail with the help of detail flowchart.

3.Homomorphic Encryption and Decryption

This is the data security method where the data is encrypted so that even if someone gets the data it will be in encrypted form so that it cannot be read or not in readable form. [9] This method of encryption that is used in the project is Homomorphic encryption. [10] Here the word Homomorphic means adding some data by replacing some. We are using the spaces as special characters those will be replaced with some special characters from a predefined database, were used encryption and decryption as described below Figure (4, 5):

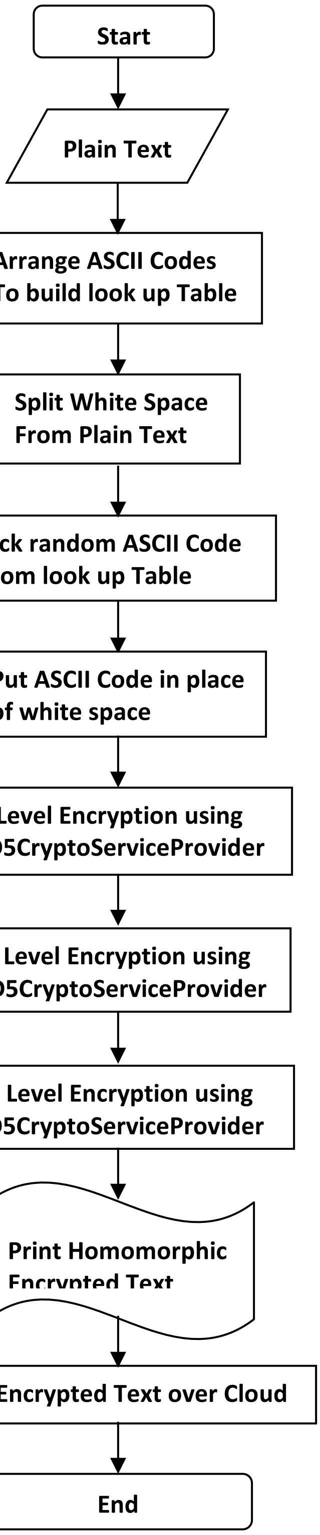
Start
Plain Text
Arrange ASCII Codes To build look up Table
Split White Space From Plain Text
Pick random ASCII Code From look up Table
Put ASCII Code in place of white space
1st Level Encryption using MD5CryptoServiceProvider
2nd Level Encryption using MD5CryptoServiceProvider
3rd Level Encryption using MD5CryptoServiceProvider
Print Homomorphic Encrypted Text
Save Encrypted Text over Cloud
End

Fig. (4): Flow Chart to do Homomorphic Encryption.

This will convert the whole text in to one sentence. This will also be needed to encrypt using standard defined techniques of encryption. [11] We are using advance encryption standard

(AES) technique to encrypt the cipher text we received by morphing using Homomorphic method. [12] This is mostly used in practice and has a 128 bit key.

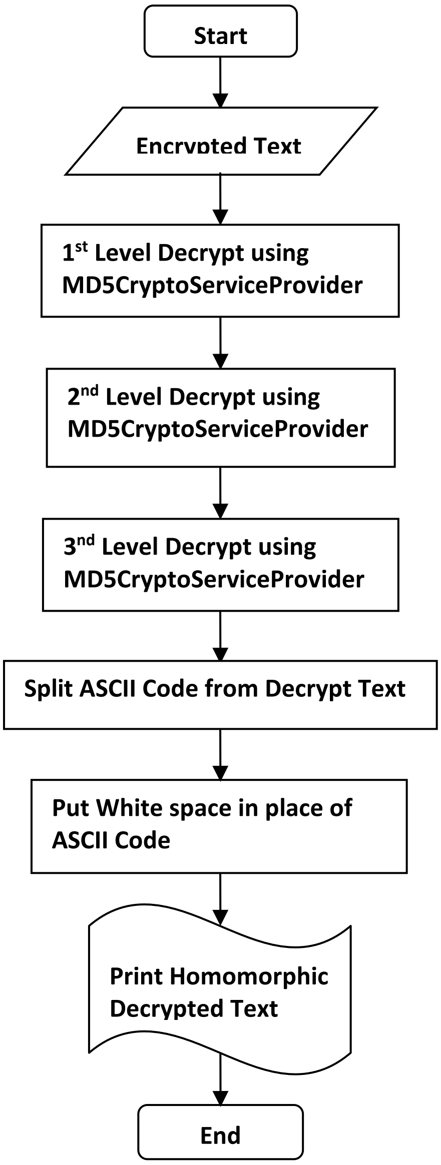
Start
Encrypted Text
1st Level Decrypt using MD5CryptoServiceProvider
2nd Level Decrypt using MD5CryptoServiceProvider
3nd Level Decrypt using MD5CryptoServiceProvider
Split ASCII Code from Decrypt Text
Put White space in place of ASCII Code
Print Homomorphic Decrypted Text
End

Fig. (5): Flow Chart to do Homomorphic Decryption.

Even we are not only using this AES for one level, but we are implementing the AES for six levels. This makes it almost impossible for any hacker or intruder to decrypt the data stored in such a format. [13]

We are using fifteen different symbols to represent spaces. This will confuse the pattern recognition software to know and understand pattern of data being used after some interval. This makes our encryption most secure [14] [15] and can be deployed on cloud server. We are using exactly opposite method to decrypt the data. We will perform six level of AES decryption [16] and then removal of special characters introduced in the text. This removal of special characters will be done as per the data base entries. In this way we will be performing the data encryption and decryption using advance encryption standard encryption & decryption techniques. [17] Before that we will morph the text with Homomorphic techniques. We will be replacing the spaces with special characters and then perform six level of encryption and decryption as needed. [18]

4. Performance Analysis

The performance of the project can be analysed according to the reading tables obtained from the project performance. We analysed the performance of the project based on the complexity of time and place. We performed tests on parameters such as the time it takes to encrypt the file, the time it takes to decrypt the file, change the file size after encryption, and the time needed to upload and upload the encrypted file. We got results showing little time to encrypt and decrypt the file. This made our project less complex in terms of time. The difference in file size was within large limits making the project less complex. According to Table 1, we can analyze that the project we have developed involves less complexity in space and time. This also shows the time needed to download and download a file on the cloud server.

Table 1: Performance Analysis.

File No	Time Taken For Encryption (milli.)	Time Taken For Decryption (milli.)	Size Before Encryption (bytes)	Size After Encryption (bytes)	Time For Uploading Encrypted File (Ticks)	Time For Downloading Encrypted File (TIcks)
1	236	918	30833	87170	5478	82092
2	201	720	28652	81562	136866	78450
3	2658	9490	78548	223298	12316	77039
4	144	341	18183	52142	4557	78505
5	20	37	5362	14778	4431	77353
6	61	147	12176	34458	4699	83374
7	94	347	18647	50754	51080	78211
8	51	152	12519	33722	4665	77922
9	20	52	5362	14778	3922	79664
10	14	29	4543	12314	3526	79529

Figure (6) shows that the size has not considerably increased after encryption solving space complexity of the algorithm in use.

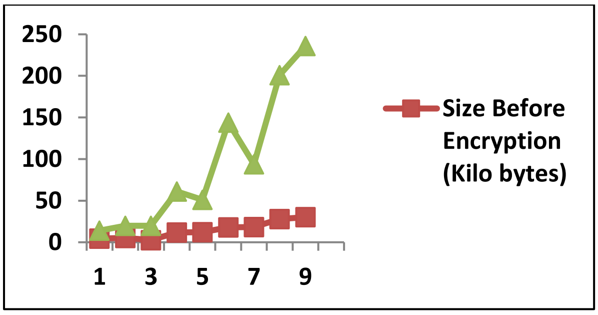

Fig. (6): After Encryption Result Chart.

This not only depends on file size but also on network speed and other network-related parameters. Fig (7) shows the file size before and after encryption

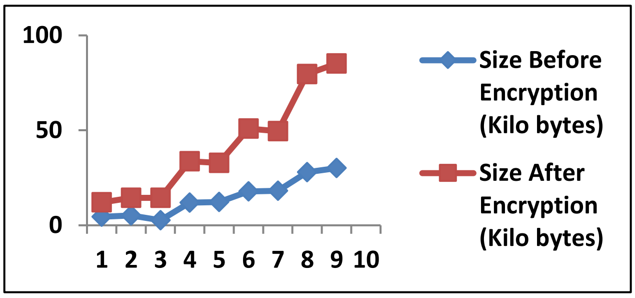

Fig. (7): File Size Comparison before and After Encryption.

But this analysis gives us an approximate idea of the loading time up and down on the cloud. This graph shows that the volume did not increase significantly after the encryption solution to the complexity of the space algorithm used. On the control of different schemas and tables, we can say that the efficiency of our project algorithm is high enough, it uses data conversion, download and download with the complexity of space and time less.

Table 2: Performance Analysis Report for Plain Text.

File No	Time For Uploading Plain File (Ticks)	Time For Downloading Plain File (TIcks)
1	2168	87569
2	2222	80820
3	2099	78392
4	2086	79126
5	2374	78305
6	2147	77587
7	2249	80569
8	2253	79146
9	2221	84191
10	2269	77125

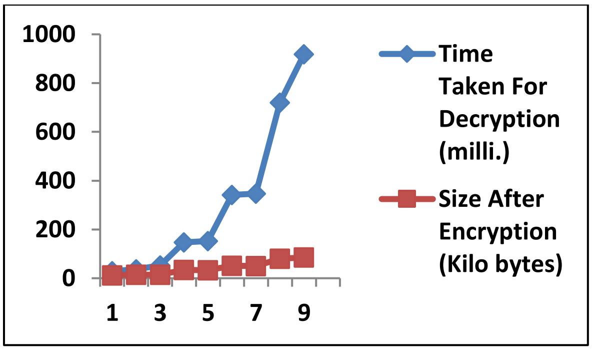

Fig. (8): shows the time it takes to decrypt encrypted files.

Also, it makes our project more secure and reliable to be implemented. Here we have not considered other report sizes such as ECG, MRI, etc. which contains graphic images. The results of this project are based on text data samples.

5. Conclusion

In today's modern world, where everything is getting computerized and high-tech with net-based services sector, the health system could also be using cloud computing based medical applications those are an active area. This will reduce the operational costs of the health care organization by eliminating data centers run by health centers just as important as improving health care by providing doctors with long-term patient data as a helping tool. We have shown on the basis of data analysis that our project to store and dispense data quickly to the user for Long-term digital measurements as a vital patient over the course of his / her treatment that is longer than what can be obtained within the hospitals and even outside. In this system, the possibility of a system to provide long-term health monitoring accurately through the use of cloud computing resources, accessed by thin devices such as tablets or mobile phones, while ensuring the privacy of the patient's complete data.The area which was not touched in this project was the mobile application, through which the data can be uploaded or downloaded using mobile instruments like Tabs or cell phones. This can be kept for the advanced researchers. We have also not covered the graphic images like ECG, MRI, EEG, etc. achieving computation in the cloud requires operating on encrypted data.

References:

[1] Ma, Y., Wang, Y., Yang, J., Miao, Y. and Li, W., 2017, Big Health Application System based on Health Internet of Things and Big Data, IEEE Access, 5, pp. 7885-7897.

[2] Moghaddasi, H., and Tabrizi, A. T., 2017, Applications of Cloud Computing in Health Systems, Global Journal of Health Science; Vol. 9, No. 6, p. 33.

[3] Rathi, G., Abinaya. M, Deepika. M¸ Kavyasri. T, 2015,Healthcare Data Security in Cloud Computing, Vol. 3, Issue 3.

[4] Manish M Potey, C A Dhote, Deepak H Sharma, 2016, Homomorphic Encryption for Security of Cloud Data, Procedia Computer Science 79, pp 175 – 181.

[5] El Bouchti, A., Bahsani, S. and Nahhal, T., 2016, Encryption as a service for data healthcare cloud security, IEEE Xplore, pp 48-54.

[6] Soubhagya, B., Venifa Mini, G.,Jeya A. and Celin J., 2013, A Homomorphic Encryption Technique for Scalable and Secure Sharing of Personal Health Record in Cloud Computing,Vol. 67, No.11.

[7] Ramaiah, Y. G. and Kumari, G. V., 2012, Efficient Public key Homomorphic Encryption Over Integer Plaintexts, IEEE, pp. 123-128

[8] Ramaiah, Y. G. and Kumari, G. V., 2013, Complete Privacy Preserving Auditing for Data Integrity in Cloud Computing ,12th IEEE International Conference on Trust, Security and

Privacy in Computing and Communications IEEE, pp. 1559-1566

[9] Yukun, N., Xiaobin, T., Shi, C., Haifeng, W., Kai, Y. and Zhiyong, B., 2013, A Security Privacy Protection Scheme for Data Collection of Smart Meters Based on Homomorphic Encryption, IEEE Euro Con, 1-4 Zagreb, Croatia, pp. 1401-1405

[10] Xiong, A. P., Gan, Q. X., He, X. X., & Zhao, Q., 2013, A Searchable Encryption Of Cp-Abe Scheme In Cloud Storage, IEEE, pp. 345-349

[11] Abozaid, G., El-Mahdy, A. and Wada Y., 2013, A Scalable Multiplier for Arbitrary Large Numbers Supporting Homomorphic Encryption, 16th Euro micro Conference on Digital System Design 2013 IEEE, pp. 969-975.

[12] Alejandro, L. and Ra´ul, E., 2013, A Cryptographic Scheme for Secure Cloud Computing, 10th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), Mexico City, Mexico. September 30-October 4, IEEE, pp. 221-226.

[13] Chuan Yao, L. X. , 2013, A Secure Cloud Storage System from Threshold Encryption, 5th International Conference on Intelligent Networking and Collaborative Systems, IEEE, pp. 541-545.

[14] Hasan, P. K. , 2013, Hybrid Attribute- and Re-Encryption-Based Key Management for Secure and Scalable Mobile Applications in Clouds, IEEE , pp. 01-14.

[15] Huang, W. W., 2013, FPGA Implementation of a Large-Number Multiplier for Fully Homomorphic Encryption. IEEE, pp. 2589-2592.

[16] Hur, J., 2013, Improving Security and Efficiency in Attribute-Based Datin Attribute-Based Data Sharinga Sharing. IEEE explore, pp. 2271-2282.

[17] Balamurugan J. D., 2013, Low Power and High Speed AES Using Mix Column Transformation. International Conference on Current Trends in Engineering and Technology, ICCTET’13 IEEE, Coimbatore, India, pp. 216-219.

[18] Lai, J.L., Liao, K.H. Lai, Y. T. and Chen, R. J., 2013, Design CAROM Module Used in AES Structure for Sub-Byte and Inv-Sub-Byte Transformation. IEEE CSI China: IEEE, CSI, pp. 198 - 201.