I. J. Computer Network and Information Security, 2019, 6, 10-21
Published Online June 2019 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2019.06.02

Cloud-based Framework for Efficient Storage of Unstructured Patient Health Records

Hanya M. Abdallah
Computer Science Department, Faculty of Computers & Informatics, Benha University
E-mail: hanya.abdelhak@fci.bu.edu.eg

Ahmed Taha
Computer Science Department, Faculty of Computers & Informatics, Benha University
E-mail: ahmed.taha@fci.bu.edu.eg

Mazen M. Selim
Computer Science Department, Faculty of Computers & Informatics, Benha University
E-mail: selimm@bu.edu.eg

Received: 16 March 2019; Accepted: 24 April 2019; Published: 08 June 2019

***Abstract*—**Recently, in healthcare sector, the data is steadily growing and becomes more vital. Most of this data is embedded in the medical record of the patient. In fact, Patient Health Records (PHRs) refer to those records that the patient can maintain, access and share among different specialists. Storing these PHRs to the cloud allow the patient to maintain and share them with different practitioners anywhere and anytime. However, he still suffers from some security and privacy issues. Hence, it is necessary to guarantee the security and privacy of this immense volume of patient's confidential data on the cloud. Anonymization and encryption are the two methods that can be adopted to ensure the security and privacy of PHRs on cloud. In this paper, a cloud-based framework for securing the storage and the retrieval of unstructured PHRs is proposed. This framework combines different encryption techniques to encrypt the different contents of the PHR, to compress medical images and to control the access to these records. In addition, the encrypted files are partitioned into a random number of files before being sent to the cloud storage server. These files are of variable number and variable size. When a user requests to access a PHR from the cloud, the proposed framework first controls access of this user before merging the partitioned files. The decryption of these files is performed on the client side not on the cloud using the secret key, which is owned by authorized user only. Finally, extensive analytical and experimental results are presented. It shows the security, scalability, and efficiency of the proposed framework.

***Index Terms*—**Cryptography, Anonymization, Unstructured Health Record, Cloud Computing Security, Chaotic Maps.

I. Introduction

Each patient may consult a number of different physicians during his lifetime. Actually, physicians need to know the complete history of the patient before they could make a proper diagnosis. This medical history can be found in the Patient Health Record (PHR) which is a set of critical information that patients keep about their health. PHR can be either structured or unstructured. In the structured one, data is stored usually in a relational database which has a definite set of attributes. On the other hand, the unstructured PHR has an internal structure although it is not structured via a predefined scheme. it combines both text data such as patient's name, birth date, address, mobile number, patient's medical history, and current medications in addition to image data such as scanned lab reports, X-rays…etc. [1, 2]. However, this information is difficult to be maintained.

The traditional way, in which each physician or medical facility has its own records and the records are not shared with others, is declared to be wasteful. One of the challenges that many healthcare organizations face is storing and sharing health information among professionals, medical facilities and insurance companies for the benefit of the patient without violating his right to confidentiality and privacy.

Lately, cloud computing becomes an excellent solution to store PHRs. It becomes possible for the patient health records to use the cloud computing technology for efficient storage and retrieval systems. Hence, it reduces time consumption and other costly operations. The user can access data from anywhere at any time through the Internet. However, storing the sensitive health

information of the patient in the cloud still suffers from many security and privacy issues. Many technologies such as anonymization, encryption, data sanitization, and randomization are exist to ensure the patients' data security and privacy [3, 4]. However, cryptography and anonymization are the most widely used especially for ensuring security and privacy of cloud-based data. A very less previous work uses anonymization for ensuring privacy and security of unstructured health data [2]. So, different types of encryption techniques are required to ensure security and privacy of unstructured PHRs and thus it prevents any unauthorized people to modify them [5].

The cryptographic techniques are applied to PHRs before placing them in the cloud environment. Since PHR files can include both text and image data, different cryptographic techniques have to be adopted to secure different formats of data in the PHR [6]. In fact, image encryption differs from text encryption due to some intrinsic features of images such as bulk data capacities, high redundancy, strong pixels correlations, etc… In this paper, PHR images are encrypted with chaotic maps due to its security, simple computation, high speed and performance in healthcare field and PHR while text files are encrypted with two standard encryption mechanisms namely AES (Advanced Encryption Standard) for its rapid performance and RSA mechanism (Rivest, Shamir, and Adelman) for its efficient security [7]. Although cryptography is an effective tool to preserve data security, we still need more security that can be achieved through access control. The access eligibility of data users should be guaranteed before accessing the cloud storage. Whereas only authorized persons have access to these files and the data owner is the only person who gives them this access right.

Before the transmission of high-resolution medical images, they need to be compressed without degrading their quality. it is known that image compression techniques can be either lossy or lossless. If the reconstructed image from the compressed image is identical to the original image then it is a lossless compression otherwise it is a lossy compression. Lossless compression is usually preferred but to achieve secrecy, some image quality degradation is accepted. Medical images compression is necessary to be as less resource consuming as possible in order to keep the cloud economically in terms of bandwidth and storage cost. Furthermore, reducing image size minimizes the time it takes for images to be uploaded to or downloaded from the cloud.

Many researchers have been developing different schemes for storing and accessing PHRs in the cloud [5, 6, 8-11]. One scheme is based on one level of security [5, 6, 8]. That is the use of only the encryption and decryption of PHR. Another scheme depends on two levels of security [9-11]. It uses encryption and decryption along with controlling the user's access to the encrypted files, but more security is still needed. Also, researchers develop compression and encryption schemes for securing the medical images. These schemes can be carried out in three ways as follows:

- Compression followed by Encryption (CE).
- Encryption followed by Compression (EC).
- Joint Compression and Encryption (JCE).

Hence, this paper proposes a framework with four levels of security. Firstly, different cryptographic techniques are employed for encryption and decryption of PHR text files and images. Secondly, the encrypted files are partitioned and merged randomly in both the number and the size of files. Thirdly, the patient has full control over the access of users to the cloud thus only authorized users can retrieve the partitioned encrypted files. Finally, the partitioned encrypted files are decrypted on client side with authorized user's private key after the merging process.

The remaining of this paper is organized as follows. Section 2 reviews some related works. The proposed framework is presented in section 3. Section 4 shows experimental results. Finally, the paper concludes in Section5.

II. Related Work

Many techniques for securing and preserving data privacy exist. These techniques can be either cryptographic (such as public-key encryption, symmetric-key encryption, and Hashing) or non-cryptographic (such as anonymization, data sanitization, and randomization) [12]. This section reviews some related work in data privacy and security using anonymization and encryption for both structured and unstructured data. The first subsection reviews anonymization-based methods. The second subsection reviews cryptographic based methods.

A. Anonymization Based Methods

Anonymization is the operation of changing data in a way that conceals the identity information about the patient. It is applied on Quasi-identifiers (QI) (i.e. when combined, it provides information about person's identity). Many anonymization techniques exist such as k-anonymity, l-diversity, t-closeness...etc. [13]. These techniques use operations of generalization, suppression, bucketization...etc. [3, 4]. The above-mentioned techniques are suitable for structured data [14, 15]. However, they appear to be unfeasible for unstructured data because they assume that one or few sensitive attributes exist in the dataset. In addition, the unstructured medical text documents have a large number of sensitive attributes such as symptoms, diagnosis, conditions, treatments, and lab test results.

a. Structured Data Anonymization

In [14] Wang *et al.* introduce a personalized privacy preservation framework over healthcare data on hybrid cloud. This framework firstly splits the dataset into multiple partitions according to a predefined criterion. Then, the non-sensitive attributes in a partition are generalized to the same generalization range. If a

partition still reveals the privacy, some sensitive values are suppressed from the partition.

Another privacy preserving framework for sharing medical records for cloud is introduced by Yang *et al.* in [15]. It is based on the classification of the medical record attributes. The framework vertically splits the medical record table into three tables which are plaintext table (Tp), anonymized table (Ta), and encrypted table (Te). Tp stores only the sensitive attributes of the medical record. Ta stores the quasi-identifiers after being anonymized using top-down approximate technique TD_Approx [16]. Te stores the unique-identifiers and the quasi-identifiers after being encrypted using combination of both symmetric and asymmetric encryption techniques. Through the vertical splitting and after merging the tables in different ways, the medical record can be accessed while meeting the privacy requirements.

b. Unstructured Data Anonymization

Most of the data released today is unstructured. However, there exist a very less previous work that uses anonymization [17, 18, 19] and appears to be unfeasible.

Gardner *et al.* introduce a method for de-identifying unstructured health data [19]. First, the method employs a simple Bayesian classifier and a conditional random field-based classifier for extraction and identification of the unstructured data attributes. Then, the link between the quasi-identifiers is removed and a k-anonymization based technique is deployed to de-identify these attributes. The usage of probability-based classifiers is unfeasible for large amount of unstructured health data.

In [18], Thavavel *et al.* introduce a method for preserving the privacy of unstructured documents. This method is firstly based on converting the unstructured documents to semi-structured one (XML). Then, the XML file is mapped to a node representation. Finally, it converts the node representation into a relational form. With the large amount of unstructured data, this solution appears to be unfeasible.

In [17] Li *et al.* introduce an approach for preserving the privacy of unstructured medical text documents called DAST (De-identification and Anonymization for Sharing medical Texts). This approach involves three modules as follows:

1. Information extraction module: it consists of three components; feature extractor, base classifiers and result aggregator. In this module, the attributes are extracted and classified using a set of independent term classifiers (e.g., rule-based classifier, SVM-based classifier, CRF-based classifier...etc.). Then, the sets resulted from each classifier are integrated.

2. Document clustering module: it clusters the medical documents based on the sensitive attributes (SA) obtained from Module 1. It employs recursive Non-negative Matrix Factorization (NMF) document clustering technique.

3. De-Identification and Anonymization module: it removes unique-identifiers and anonymize quasi-identifiers using value-enumeration and drill-down methods.

B. Cryptographic Based Methods

Cryptography is the science of converting plaintext data into non-readable form called cipher text data. There exist many text encryption techniques to ensure the privacy and security of data on the cloud. Cryptographic techniques have two categories [20]. They are either traditional such as AES, RSA, DES, and BLOWFISH, etc. [21] or advanced such as Homomorphic Encryption, Identity Based Encryption, and Attribute-based Encryption (ABE) with its variations [20, 22].

a. Structured Data Encryption

Many researchers use Homomorphic encryption [23, 24, 25] and Attribute Based Encryption (ABE) techniques [5, 8- 10, 26-28] such as key-policy ABE, cipher-text policy ABE and multi-authority ABE for secure sharing of the structured PHRs. When using ABE, they encrypt the PHRs under a specific set of attributes called an access control policy. No one can access or decrypt them unless he/she has the same set of attributes.

In [11], Alias and Roy present a framework for secure sharing of PHRs in the cloud. It depends on what is called the patient-centric idea. It means that only the patient has full control over his health record. It ensures privacy and confidentiality of patient health records by determining who can access the record and who cannot through applying a combination of encryption techniques (KP-ABE, Multi-Authority ABE and traditional cryptography). By splitting the users into two different domains (personal domain and public domain), the framework is able to overcome the key management problem.

According to [9], Shrestha *et al.* consider a framework for secure access control to Personal Health Information (PHI) based on Multi Authority Attribute Based Encryption (MA-ABE) method. Firstly, the users log on using a username, a password and a unique biometric information. Secondly, Data Access Requester (DAR) requests the database and the cloud for a service of access to store PHI then the user is checked whether he is granted to access the service or not by Single point of contact (SPOC). If the user is authenticated, then the PHI can be accessed using MA-ABE technique for managing system scalability and avoid outsiders attack like eavesdropping, Man-in-middle attack, and denial of service (DOS) attack. The PHI is encrypted with AES before being moved to the cloud.

In [29], a general framework for secure accessing and sharing of PHRs is presented. The main idea of this framework is to secure access of a special set of users to some data stored remotely in the cloud servers. This is achieved by ABE. Moreover, the physician can share patients' health record securely to other specialists for the aim of examination without revealing the privacy of the patient. This process can be performed using a proxy re-encryption mechanism where the user sends the proxy re-encryption key to the cloud server. Then, the user asks the server to re-encrypt the ciphertext to some new attributes. The framework supports break-glass access in

emergency states through including emergency department attribute (ER Dept) using OR logic operator "OR ER Dept" in the access policy. Hence, the emergency department can decrypt any health record file. Also, the framework takes into account user revocation (user cannot access a certain PHR file in case of his attributes become invalid) and cross domain PHR sharing which means that a patient may desire to move his/her PHR to a domain in another country where he/she will settle in.

Mudanna *et al.* [30] introduce an algorithm to provide data confidentiality called CEASE (Cryptographically Enforced Access control for Securing Electronic medical records in the cloud). It essentially depends on two ideas. Firstly, according to a set of attributes offered by user during registration phase, the proxy server applies an access control policy on PHR data and classifies the users as a patient, a physician, or a researcher. Secondly, the patient encrypts the PHR attributed by AES before moving to the cloud and encrypts them using specific set of attributes so no one can decrypt them except they have the same set of attributes. CEASE allows encrypted queries on encrypted data and then decrypt them under a set of attributes so they can read data before delivering it to an end user. Furthermore, it performs partial shuffling among a restricted data block in order to prevent the malicious users to understand the hot health records, which are accessed continuously.

Chandrasekhar *et al.* introduce an authorization protocol for health information exchange (HIE) on cloud. Most of existing ABE schemes combines encryption with authorization, but [25] uses a combination of authorization and authentication. It develops a trapdoor hashing scheme in a specific manner to construct a proxy signature-based protocol. Therefore, it allows patient's health information to be accessed and exchanged between health providers and patients in a selective manner under certain agreed policies. During storage the patient data is encrypted using homomorphic scheme, also transport layer security protocol is used to secure all communications between the patient and healthcare organizations (HCOs). Both patient and HCOs generate their public and private keys, register with the HIE cloud server, and obtaining valid certificates. HCO gives the patient his credentials when creating his health record for the first time. Patient uses these credentials to authorize other HCOs and control the type of health information that can be accessed by these HCOs.

b. Unstructured Data Encryption

A system essentially based on the idea that PHR has two types of data (text data, scanned images) is presented in [6]. In this system, AES is used to encrypt the PHR text files. Scanned images are encrypted using Paillier Cryptosystem after being minimized into pixels. Due to the existence of many decryption key hacking techniques, anyone can easily retrieve the data from the cloud and decrypt it once he/she knows the decryption key.

Arunkumar and Anbuselvi present a framework in [31] that uses only a cryptographic technique for securing PHRs on the cloud. It encrypts health records before they are stored on cloud. The framework uses two layers of protection to secure PHRs. In the first layer, the encryption of PHR images and text files is done using AES with a key size of 128 bit. In the second layer, a number of n files are generated by splitting the encrypted files with a sequence key entered by the patient. These n files are then stored in the cloud. Authorized physicians/users can decrypt the PHR files only if these files are merged using a private key. For merging the partitioned encrypted files, a sequence key is entered by the physician or other authorized persons which matches the sequence key entered by patient in the splitting process.

For the medical image data, traditional encryption techniques are poorly suitable for them because medical images have their special intrinsic features such as strong pixel correlation and high redundancy. In addition, they contain more content, and have bulky data, which makes the encryption process complex and requires more time. Therefore, researchers recently turn to use chaotic systems (maps) for medical image encryption and decryption to resist all the mentioned problems of traditional techniques and overcome the security attacks [32-34].

Chaotic is extended from chaos word, which does not have a defined meaning and does not have a deterministic behavior i.e. it behaves randomly. Chaotic systems are very sensitive to initial conditions/ system parameters so a small change in initial conditions leads to very large change in the cipher text. It achieves better security, simple computation, high speed and performance in healthcare field.

According to [32], Kumar and Fathima present a technique for securing medical x-ray images using chaotic maps. They use Henon and Chebyshev maps as the chaotic systems adopted in this process. Performance characteristics are evaluated with histogram, entropy and correlation. For evaluation, encryption is performed with three approaches, only using Henon map, only using Chebyshev map and a combination of both maps.

Another chaotic medical image encryption algorithm based on bit- plane decomposition is introduced in [33] by Yin *et al.* This scheme uses bit-plane decomposition and three different models of chaotic maps (Arnold Cat Map, Henon Map, and Logistic Map) for permutation and diffusion. Firstly, the medical image is decomposed into 8 bit-planes. Then, the high four-bit-planes are taken to perform scrambling operation using Arnold Cat Map. They contain a large number of pixel information of the original image. Finally, in the diffusion phase, the logistic map is used to set the parameters of the Henon map then XORed with the permuted image to generate the ciphered image.

Although the framework that is presented in [31] provides two layers of protection but it suffers from some drawbacks. It uses one encryption technique (AES) for both PHR text data and images. However, image encryption is different from text encryption due to some intrinsic features of images. Moreover, the framework

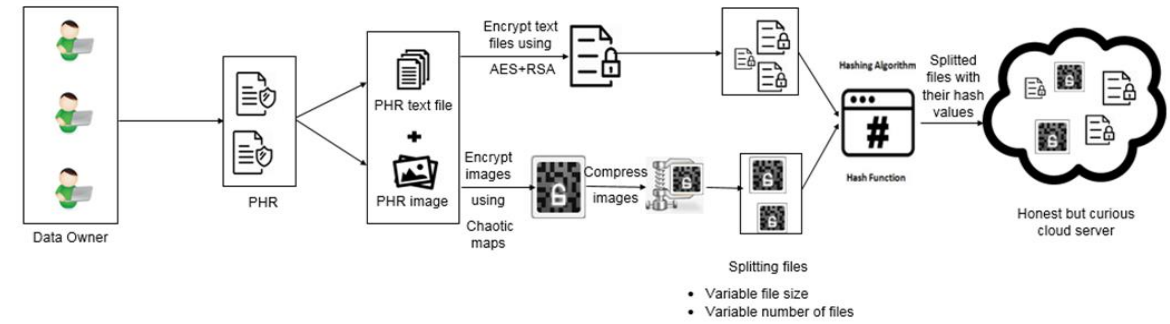
requires some kind of user interaction to enter a sequence number used for splitting /merging the encrypted files. Based on the number of characters in this sequence, the encrypted PHR file is partitioned into definite number of files of the same size (i.e. if the patient enters "bfdk5467" as a sequence number, so the encrypted file will be partitioned into eight files of same size). One more drawback of this framework is that an intruder can access the cloud and view the encrypted files with no access control from the patient. Therefore, the intruder can decrypt the PHR files easily. Once the sequence key is known, there are several techniques such as Key Search technique, Brute Force attack available to hack the keys.

III. Proposed Framework

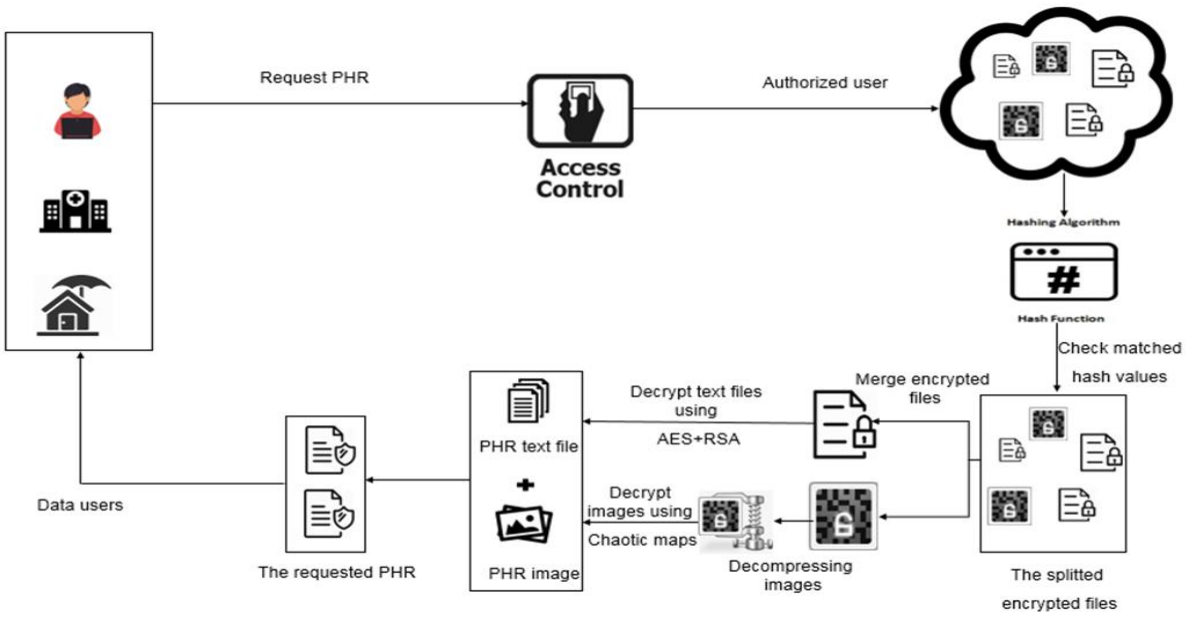
The proposed framework mainly focuses on secure uploading and downloading of PHRs while ensuring their integrity. It consists of three main stages: user registration and login, PHR upload to the cloud storage, and PHR retrieval from the cloud storage. The three stages together provide four levels of protection. In the first level, PHR is encrypted with different cryptographic techniques according to the type of the data in the record. That is, images are encrypted using chaotic maps while text files are encrypted using a combination of symmetric and asymmetric techniques (AES+RSA). In the second level, each encrypted file is further partitioned into a set of files that are variable in both the number and the size. In the third level, the patient has a full access control to determine the users of his record in the cloud. Hence, only authorized users can retrieve the partitioned encrypted files. In the fourth level, the partitioned files are decrypted on client side with authorized user's private key after the merging process is finished. The framework stages are explained in detail as follows:

Stage 0: User Registration and Login

This stage is responsible for the user registration process as well as the processes of logging in and logging out of the system. Users can create an account by providing username, email and password, login and logout. In our framework, we have two types of users: "patient" and "physician or hospital or insurance company". The login module must differentiate between the two roles. When a user goes to log in, the system forwards him to stage one (see Fig. 1.a) if his role is "patient" or to stage two (see Fig. 1.b) if his role is "Physician or hospital or insurance company".



(a) PHR upload to the cloud storage (stage 1).



(b) PHRs retrieval from the cloud storage (stage 2).

Fig.1. The proposed framework for storing and retrieving PHRs in the cloud.

Stage 1: PHR Upload to Cloud Storage

As shown in Fig. 1.a, this stage contains three entities and four processes. The three entities are data owner, patient health records and cloud storage while the four processes include encrypting the PHR Files, compressing the medical images, splitting the output files, and applying a hash function on each file. A brief description of each one of them is given below.

1) Data Owner

A data owner is the patient who possesses PHRs. The patient should be capable of creating, controlling, and sharing his PHRs with a large number of data users. Once the patient logs in the system, he is able to view all his health record files with its description and uploading date. Moreover, he can add or delete files from his health record, and control access to his files through either authenticate user or revoke user.

2) Patient Health Records

Patient health records are a collection of files that the patient needs to store on cloud. These files may include identification sheet, patient's medical history, laboratory reports, scan reports, X-rays, operative reports, pathology reports, their current medications, progress notes, etc. As can be noticed, the PHR files are of different content. Some of them may include text data while others may include images. Hence, different cryptographic techniques are needed to ensure the privacy and confidentiality of these records. Once patient tries to upload a file to his health record on cloud, the file is first cached into a temporary directory and after being encrypted with the appropriate encryption technique, the original file is unlinked from the temporary directory.

3) Encryption

Based on the concept of attribute-based encryption, when the patient tends to encrypt a PHR file before being moved to the cloud, the proposed framework first checks the extension attribute of this file. If it has any of known text data extensions such as 'txt', 'text', 'doc', 'docx', 'pdf', etc., the proposed framework deploys a text-based encryption technique that combines AES and RSA algorithms (see Fig. 2). In fact, AES is symmetric encryption algorithm (i.e. uses the same key for encryption and decryption). However, RSA is asymmetric encryption algorithm (i.e. uses two different keys, one is public and the other is private). The proposed framework ensures double integrity and confidentiality as it first encrypts the PHR text file using AES algorithm with a randomly generated 256-bit key length then encrypts the AES key using RSA public key of 1024-bit. Through this hybridization (i.e. the AES key is encrypted based on RSA), the framework can accredit the access control to the data owner.

On the other side, if a PHR file has any of known image extensions such as 'png', 'bmp', 'jpg', 'jpeg', etc., the proposed framework deploys an image encryption algorithm based on chaotic maps under an access policy which is (file original name appended with MD5 of its uploading date/time) or (file size). A chaotic based image encryption algorithm is introduced in [35] and is applied on colored images but in our work, it is modified to be suitable for gray medical images (see Fig. 3)

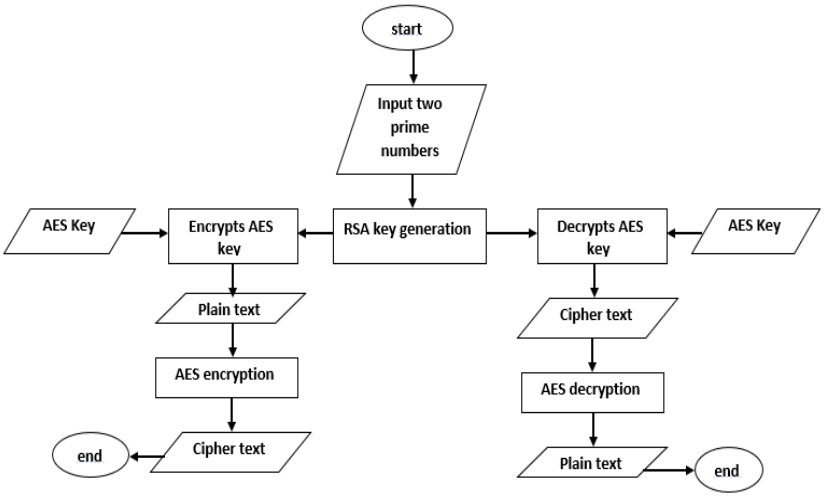


Fig.2. Flowchart of text-based encryption technique

Chaotic Based Image Encryption Technique

Step 1: Read the image A, and get its size M ×N.

Step 2: Divide the image into a matrix of (U, V) blocks .each block is 8 ×8 pixel.

Step 3: Convert the matrix into one dimensional vector and rearrange the vector using a transformation map generated from a chaotic formula (1) in [35]

When: $1.41 < \mu < 1.59$

$$x_{n+1} = \mu x_n (1 - x_n)(2 + x_n) \quad (1)$$

Step 4: Generate two transformation maps using a chaotic formula (2) in [35] for shuffling the rows and columns for each block. when $2.75 < \mu_1 < 3.4$, $2.7 < \mu_2 < 3.45$, $0.15 < r_2 < 0.21$ and $0.13 < r_2 < 0.15$

$$\begin{aligned} x_{i+1} &= \mu_1 x_i (1 - x_i) + \gamma_1 y_i^2 \\ y_{i+1} &= \mu_2 y_i (1 - y_i) + \gamma_2 (x_i^2 + x_i y_i) \end{aligned} \quad (2)$$

Step 5: Diffuse the pixels of each block using diffusion formula in [35].

$$c_{now} = (p_{now} + z_{2n} + c_{pre} + p_{pre}) \bmod 256$$

Fig.3. Chaotic based image encryption technique

4) Compressing Medical Images

After the images being encrypted, Discrete Cosine Transform algorithm (DCT) is used, which compresses the image with a good compression ratio. As shown in Fig. 4, this process can be completed in five steps.

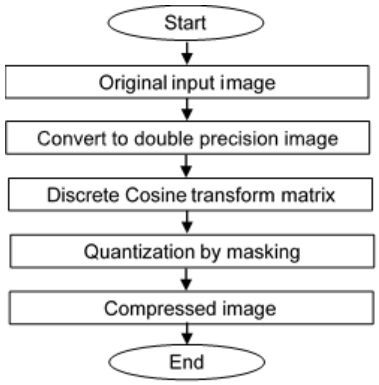


Fig.4. Flowchart of the compression process

5) Splitting Encrypted PHR Files

After compression of medical images and encryption process is completed, the encrypted file is partitioned based on its size into a random number of files with variable sizes. The partitioned files that are of different sizes and names make the process of attacking so difficult to unauthorized users. This is one of the most important distinguishing features of the proposed framework. The randomization of the number of partitioned files for every encrypted file forms a second level of security to the PHR data after the first one embedded in the encryption process.

6) Applying a hash function

After the splitting process is completed, a hash function is applied to generate a unique hash value for each partitioned one. This hash value can be used later for checking the integrity of the files.

7) Cloud storage

The partitioned encrypted PHR files are stored in the cloud storage. Each file outsourced to the cloud servers has a unique name which is the original file name concatenated with the uploading date/time stamp after being encrypted with MD5. Now, the patient can manage his health record either by deleting any file or by authenticating certain physicians to access specific files.

Stage 2: PHRs Retrieval from the cloud storage

As shown in Fig. 1.b, this stage combines three entities and four processes. The three entities are data owner, data user and cloud storage while the five processes include access control, check matched hash values, merge partitioned files, decompressing the image files, and decrypt merged files. A brief description of each one of them is given below.

1) Data Users

Data users or physicians are authorized persons who can access patient ' s PHRs to provide the medical diagnosis. Once they have the right to access the PHR data, they can see all PHR files that they have permission to access. Even if the files belong to more than one patient, they still can access it. In fact, the data user has to request a private key required to decrypt and access a certain file from the patient (the data owner).

2) Access Control

When a data user requests to access a PHR file, the data owner first checks the user's eligibility to access the PHR. If he is an authorized one, he can view all PHR files he is allowed to access. The proposed framework makes the data owner the only person who controls access to his health record. In other words, he is capable of revoking the access to his record to prohibit a user to access them.

3) Check Matched Hash Values

Once the data user is authenticated to access a certain file, a hash function is applied on each part of it. The computed hash value of each part of the requested file is compared to the hash value previously stored with each one. If the two values are matched, it means that attackers have not changed the content of the file.

4) Merge Partitioned Files

Once the data intgerity is assured, the partitioned file he requests is identified and its parts are merged togther. Thus, the requested file is reassembled but is still

encrypted.

5) Decompressing the Image Files

After the partitioned parts of image files are merged, the steps of compression process (see Fig. 4) are reversed to reconstruct the original encrypted image from the compressed one. Usually this process is faster than compression.

6) Decrypt merged files

Like encryption, the decryption technique probes for the extension attribute of the file. That is, if the file has an extension of the well-known extensions of the text files, the decryption process is done based on AES+RSA. It asks the data user to provide his secret key. Fisrt, the file is decrypted with the AES key using the RSA secret key entered by the authorized user. Second, the retrieved AES key is used to decrypt the PHR merged file. On the other hand, if the file extension is one of the images extensions, the system applies the reverse process of the chaotic map based on the encryption algorithm after ensuring that the access policy associated with the data user matching the access policy of the requested file.

IV. Experimental Results

To evaluate the performance of the proposed framework, several experiments have been conducted. First, the system is fully implemented using PHP 5.5 and MYSQL except the image encryption technique. It is implemented using MATLAB R2016b. All experiments are run on a personal computer with windows 7 64-bit operating system with an Intel® core™ i7-4510U CPU (2.6 GHz) and 8 GB of RAM. The performance of the proposed framework is evaluated in terms of security analysis, encryption/decryption time of different size PHRs, split/merge time for individual PHR files, and performance of the compression technique used. Each one is discussed in the following subsections.

A. Dataset

Two datasets are available for patient health records: MIMIC-III [36] and eICU [37]. The two datasets contain detailed information regarding the clinical care of ICU patients. Unfortunately, we cannot access these datasets because it needs to complete the CITI "Data or Specimens Only Research" course. Therefore, we turn to create our own PHR dataset to evaluate the performance of the proposed framework.

Our own PHR dataset contains 100 patient health records ranging in size from 0.122 MB to 12.1 MB depending on the patient's health status. Each record consists of 26 text files, 2-7 lab reports and 2-7 medical images. Text files give information about patient's entering to the hospital, patients' routine vital signs and any extra information related to their health. Lab reports can be lab test results or reports on medical radiography. Medical images of size 256×256 are collected from [38] and can be Magnetic Resonance Imaging (MRI), Computed Tomography (CT), and X-Ray...etc.

B. Security Analysis

This section illustrates how the proposed framework complies with the intended security requirements such as confidentiality, integrity, access control, and availability.

a. Confidentiality

First, one of the most important distinction points of the proposed framework is that it does not deal with the PHR as a single unit but rather it distinguishes its content. The PHR can contain medical images and text files. The proposed framework deals with each case with a different technique within the same record. There is no doubt that using more than one encryption technique at the same time will serve to upgrade the security level. On the other side, the framework presented in [31] do not differentiate between the content of a medical record and use the same encryption technique (AES) regardless of its content.

Second, the proposed framework encrypts the PHR text files using a hybrid technique of AES and RSA. When using keys as 256 bit AES and 1024 bit RSA, detecting the private key is impossible even if the attacker owns the generated public keys. Furthermore, the proposed framework encrypts the medical images using chaotic maps based algorithm. This type of algorithms are characterized by better security, simple computation and high speed. On the other side, the frameworks presented in [33] use only AES technique for encrypting the different types of PHR data. AES requires more time for encrypting the medical images due to its complex encryption process. Surely, employing different techniques increases the security level of the proposed framework.

Table 1. Encryption and Decryption Time of the Proposed Framework

Record ID	Record Size (MB)	No. of Images	Images Encryption Time	Images Decryption Time	No. of text files	Text files Encryption Time	Text files Decryption Time	Record Encryption Time	Record Decryption Time
26	0.51	5	0.5024	0.55435	31	0.0218	0.057	0.5242	0.61135
546	1.04	6	0.60288	0.66522	32	0.0279	0.059	0.63078	0.72422
346	2.03	7	0.70336	0.77609	33	0.0389	0.0798	0.74226	0.85589
283	4.31	3	0.30144	0.33261	29	0.071	0.1226	0.37244	0.45521
286	8.32	4	0.40192	0.44348	30	0.09085	0.13825	0.49277	0.58173

Table 2. Encryption and Decryption Time of Scheme Presented in [31]

Record ID	Record Size (MB)	No. of Images	Images Encryption Time	Images Decryption Time	No. of text files	Text files Encryption Time	Text files Decryption Time	Record Encryption Time	Record Decryption Time
26	0.51	5	2.508585	0.589515	31	0.03099	0.1134	2.539575	0.702915
546	1.04	6	3.010302	0.707418	32	0.08201	0.1401	3.092312	0.847518
346	2.03	7	3.512019	0.825321	30	0.11341	0.1734	3.625429	0.998721
283	4.31	3	1.505151	0.353709	29	0.2257	0.1903	1.730851	0.544009
286	8.32	4	2.006868	0.471612	30	0.4014	0.2098	2.408268	0.681412

Third, there are several techniques available to hack the decryption keys including key search technique, brute force attack, crypt analysis and systems-based attack. However, the proposed framework is robust against these attacks by providing a higher level of security. In this level, the encrypted files are partitioned into a variable number of files with variable sizes. Therefore, in case that the user forgets to close his session and the attacker tries to download and to decrypt data, he is asked to enter the private key and cannot decrypt the files unless they are merged successfully.

b. Integrity

As shown in stage 2 of the framework, after the user is authorized, the computed hash value of each part of the requested file is compared to the hash value previously stored with each one. If the two values are matched, it means that the accuracy and consistency of the file has not been changed by attackers

c. Access Control

The proposed framework makes the data owner the only person who controls access to his health record. He can update his record by adding or deleting files. In addition, he can grant or revoke access to his record files to prohibit unauthorized users to access them.

d. Availability

Usage of cloud to store and retrieve PHRs ensures the availability of them when wanted. Also, it provides on demand accessing of PHRs.

C. Encryption and Decryption Time

Table 1. and Table 2. show the results of implementing the proposed framework and the framework presented in [31] respectively for storing and accessing different medical records. The results here are illustrated by encryption and decryption time of PHRs of different sizes ranging from 0.51 MB to 8.32 MB. Note that the framework presented in [31] use only AES technique for encrypting the whole PHR.

From the tables, it is observed that the encryption and decryption time of the PHRs mainly depend on the number and the size of medical images in the record. The encryption and decryption time of medical images in the proposed framework is less than the encryption and decryption time in scheme presented in [31] by about 20% and 94% respectively. In addition, the encryption and decryption time of text files in the proposed framework is less than the encryption and decryption time in scheme [31] by about 25% and 50% respectively. All of this is reflected on the total encryption and decryption time of the PHR. Therefore, the total time for encrypting and decrypting the PHRs in the proposed framework is less than the total time for encrypting and decrypting them in scheme [31] by about 20% and 85% respectively. Tables 1 and 2 also show that the decryption time of the PHR is greater than the encryption time in the proposed framework. This is due to the process of recovering the encrypted AES key on the server for being used in the text files decryption process.

Table 3. Time to split and merge different sizes PHR files

File size (kb)	No. of partitioned files	Split time (sec)	Merge time (sec)
64	8	0.00300002	0.00500106
128	3	0.00199985	0.00300002
256	8	0.00500011	0.00599980
512	4	0.00639994	0.01100110
1024	4	0.00700092	0.021000862

D. Split and Merge Time

As shown in Table 3. , The split and merge time is mainly based on the size of file. For example, consider 64 kb and 1024 kb files, the two files are partitioned into eight and four files respectively. Although the number of partitioned files of 64 kb file is double the number of partitioned files of 1024 kb file, the split and merge time of 1024 kb file is much greater than of 64 kb file.

E. Compression Performance

To evaluate the performance of the compression algorithm, medical images of different types (MRI and CT) and size (256×256 and 512×512) are used as shown in Fig. 4. Also, several metrics are calculated such as image compression ratio, Mean squared error (MSE), Peak signal-to-noise ratio (PSNR), and compression time.

1) **Image compression ratio percentage (CRP)**: is calculated from the ratio between the original image size and compressed image size as follows:

$$CRP = \left(\frac{1}{\text{original image size/ compressed image size}} \right) \times 100\% \quad (3)$$

2) **Mean squared error (MSE):** indicates an error between the original image and compressed image.

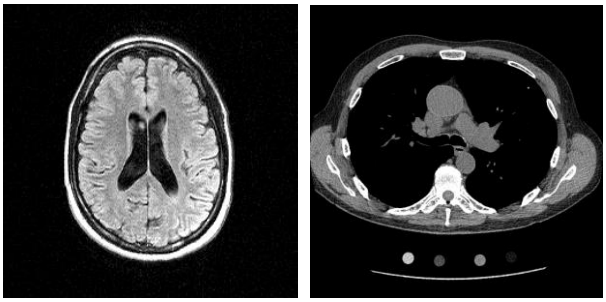
It should be as small as possible. It can be calculated as follows where m, n are the number of rows and columns.

$$MSE = \frac{\sum_{m,n} \left(uncompressed(m,n) - compressed(m,n) \right)^2}{m \times n} \quad (4)$$

3) **Peak signal-to-noise ratio (PSNR):** is related to MSE and it gives the amount of noise in acompressed image. PSNR should be as high as possible. R is the maximum possible pixel value of image

$$PSNR = 10 \log_{10} \left(\frac{R^2}{MSE} \right) \quad (5)$$

4) **Compression time:** is the time that algorithm takes to convert the original image to a compressed one.



(a) MRI_brain.png (b) CT_Lung.png

Fig.4. Medical Images of Different Types and Size.

Table 4. Performance of the compression technique used

Metric / Image	MRI_brain.png		CT_Lung.png	
Resolution	**256×256**	**512×512**	**256×256**	**512×512**
Original Size(KB)	65.053	178.08	35.523	257.053
Compressed Size(KB)	5.9570	20. 5322	5.1347	16.0811
CRP	9.2%	11.5%	14.45%	6.3%
MSE	0.004	0.00051	0.0022	0.00201
PSNR	72.1	83.89	74.628	74.9
Compression time	0.4875	1.956	0.5701	1.928

The quality of the image after the decompression is ascertained using the MSE and PSNR. Table 4. Shows that the small values of MSE and the large values of the PSNR indicates the good quality of the compressed images. In addition, the table shows that the speed of the algorithm is directly proportional to the size of the image. It takes about 1.9 sec to compress 512×512 images. For example, if we need to store the MRI_brain.png image of size 512×512 PX, the table shows that we will save 88.5% of the cloud storage.

V. Conclusion

This paper presents a secure framework for storing and retrieving PHR files in the cloud. The framework is characterized by its high security because it maintains four levels of protection: the use of two different encryption techniques based on the type of PHR files, splitting the encrypted files randomly in the number and the size, granting access control to the users, and finally decrypting the files on client side not on the server side.

Experimental results compare the proposed framework with existing frameworks. The results suggest that the proposed framework is more secure and achieve better encryption and decryption time. In addition, the time needed to split and merge the files is very small compared to the return we get in increasing the security. We believe that our research will serve as a base for future studies on securing PHRs in the cloud environment. As a future work, we suggest that the framework supports multi-keyword fuzzy search in order to allow doctors to provide the proper medical diagnosis.

References

[1] H. Elmogazy, "Towards Healthcare Data Security in Cloud Computing," 8th Int. Conf. Internet Technol. Secur. Trans., pp. 363–368, 2013.

[2] P. R. M. Rao, S. M. Krishna, and A. P. S. Kumar, "A Case Study on Privacy Threats and Research Challenges in Privacy Preserving Data Analytics," *Proc. Int. Conf. Electron. Commun. Aerosp. Technol. ICECA 2017*, vol. 2017–Janua, pp. 185–188, 2017.

[3] M. Jayabalan and M. E. Rana, "Anonymizing Healthcare Records: A Study of Privacy Preserving Data Publishing Techniques," *Adv. Sci. Lett.*, vol. 24, no. 3, pp. 1694–1697, 2018.

[4] B. Selvaraj and S. Periyasamy, "A Review of Recent Advances in Privacy Preservation in Health Care Data Publishing," *Int. J. Pharma Bio Sci.*, vol. 7, no. 4, pp. 33–41, 2016.

[5] M. P. Radhini, P. Ananthaprabha, and P. Parthasarathi, "Secure Sharing of Medical Records Using Cryptographic Methods in Cloud," *Int. J. Comput. Sci. Mob. Comput.*, vol. 3, no. 4, pp. 514–521, 2014.

[6] R. Aiswarya, R. Divya, D. Sangeetha, and V. Vaidehi, "Harnessing Healthcare Data Security in Cloud," *2013 Int. Conf. Recent Trends Inf. Technol. ICRTIT 2013*, pp. 482–488, 2013.

[7] S. Belguith, A. Jemai, and R. Attia, "Enhancing Data Security in Cloud Computing Using a Lightweight Cryptographic Algorithm," in *11th International Conference on Autonomic and Autonomous Systems*, 2015, pp. 98–103.

[8] N. Ramakrishnan and B. Sreerekha, "Enhancing Security of Personal Health Records in Cloud Computing by

Encryption," *Int. J. Sci. Res.*, vol. 4, no. 4, pp. 298–302, 2015.
[9] N. M. Shrestha, A. Alsadoon, P. W. C. Prasad, L. Hourany, and A. Elchouemi, "Enhanced E-Health Framework for Security and Privacy in Healthcare System," in *6th International Conference on Digital Information Processing and Communications*, 2016, pp. 75–79.
[10] M. Li, S. Yu, and Y. Zheng, "Scalable and Secure Sharing of Personal Health Records in Cloud Computing Using Attribute-Based Encryption," IEEE Trans. PARALLEL Distrib. Syst., vol. 24, no. 1, pp. 131–143, 2013.
[11] A. E. Alias and N. Roy, "Improve Security of Attribute Based Encryption for Secure Sharing of Personal Health Records," *Int. J. Comput. Sci. Inf. Technol.*, vol. 5, no. 5, pp. 6315–6317, 2014.
[12] B. Singh, A. Singh, and D. Singh, "A Survey of Cryptographic and Non-Cryptographic Techniques for Privacy Preservation," *Int. J. Comput. Appl.*, vol. 130, no. 13, pp. 7–10, 2015.
[13] K. Abouelmehdi, A. Beni-Hessane, and H. Khaloufi, "Big healthcare data: preserving security and privacy," *J. Big Data*, vol. 5, no. 1, pp. 1–18, 2018.
[14] W. Wang, L. Chen, and Q. Zhang, "Outsourcing High-Dimensional Healthcare Data to Cloud with Personalized Privacy Preservation," *Comput. Networks*, vol. 88, pp. 136–148, 2015.
[15] J. J. Yang, J. Q. Li, and Y. Niu, "A Hybrid Solution for Privacy Preserving Medical Data Sharing in the Cloud Environment," *Futur. Gener. Comput. Syst.*, vol. 43–44, pp. 74–86, 2015.
[16] J. Li, J. Yang, Y. Zhao, and B. Liu, "A Top-down Approach for Approximate Data Anonymization," in *Enterprise Information Systems*, 2013, pp. 272–302.
[17] X.-B. Li and J. Qin, "Anonymizing and Sharing Medical Text Records," *Inf. Syst. Res.*, no. April, pp. 1–21, 2017.
[18] V.Thavavel and S.Sivakumar, "A generalized Framework of Privacy Preservation in Distributed Data mining for Unstructured Data Environment," *Int. J. Comput. Sci. Issues*, vol. 9, no. 1, pp. 434–441, 2012.
[19] J. Gardner and L. Xiong, "An integrated Framework for De-Identifying Unstructured Medical Data," *Data Knowl. Eng.*, vol. 68, no. 12, pp. 1441–1451, 2009.
[20] R. Kirubakaramoorthi, D. Arivazhagan, and D. Helen, "Survey on Encryption Techniques used to Secure Cloud Storage System," *Indian J. Sci. Technol.*, vol. 8, no. 36, pp. 1–7, 2015.
[21] R. Bhanot and R. Hans, "A Review and Comparative Analysis of Various Encryption Algorithms," *Int. J. Secur. Its Appl.*, vol. 9, no. 4, pp. 289–306, 2015.
[22] M. Y. Shabir, A. Iqbal, Z. Mahmood, and A. Ghafoor, "Analysis of Classical Encryption Techniques in Cloud Computing," *J. Tsinghua Sci. Technol.*, vol. 21, no. 1, pp. 102–113, 2016.
[23] P. Awasthi, S. Mittal, S. Mukherjee, and T. Limbasiya, "A Protected Cloud Computation Algorithm Using Homomorphic Encryption for Preserving Data Integrity," in *Recent Findings in Intelligent Computing Techniques. Advances in Intelligent Systems and Computing*, Singapore: Springer, 2019, pp. 509–517.
[24] J. Zhou, X. Lin, S. Member, X. Dong, Z. Cao, and S. Member, "PSMPA : Patient Self-Controllable Cooperative Authentication in Distributed m-Healthcare Cloud Computing System," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 6, pp. 1693–1703, 2015.
[25] S. Chandrasekhar, A. Ibrahim, and M. Singhal, "A Novel Access Control Protocol Using Proxy Signatures for Cloud-Based Health Information Exchange," *Comput. Secur.*, vol. 67, pp. 73–88, 2017.
[26] D. A. Gondkar and V. S. Kadam, "Attribute Based Encryption for Securing Personal Health Record on Cloud," in *2nd International Conference on Devices, Circuits and Systems*, 2014, pp. 1–5.
[27] P. Deshmukh, "Design of Cloud Security in the EHR for Indian Healthcare Services," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 28, no. 1, pp. 146–153, 2016.
[28] H. Qian, J. Li, and Y. Zhang, "Privacy-Preserving Personal Health Record Using Multi-Authority Attribute-Based Encryption with Revocation," *Int. J. Inf. Secur.*, vol. 14, no. 6, pp. 487–497, 2014.
[29] M. H. Au *et al.*, "A General Framework for Secure Sharing of Personal Health Records in Cloud System," *J. Comput. Syst. Sci.*, vol. 90, no. March, pp. 46–62, 2017.
[30] K. Chennam and L. Mudanna, "C E A S E : Confidentiality and Access Control for Securing Personal Health Records in the Cloud," *Ann. Comput. Sci. Ser. J.*, vol. 14, no. 2, pp. 37–45, 2016.
[31] R. J. Arunkumar and R. Anbuselvi, "Enhancement of Cloud Computing Security in Health Care Sector," *Int. J. Comput. Sci. Mob. Comput.*, vol. 6, no. 8, pp. 23–31, 2017.
[32] V. P. K. Reddy and A. A. Fathima, "Efficient Encryption Technique for Medical X-ray Images using Chaotic Maps," in *IEEE International Conference of Wireless Communications, Signal Processing and Networking*, 2016, pp. 783–787.
[33] Y. Dai, H. Wang, and Y. Wang, "Chaotic Medical Image Encryption Algorithm Based on ¯ Bit-Plane Decomposition," *Int. J. Pattern Recognit. Arti ¯cial Intell.*, vol. 30, no. 4, pp. 1–15, 2016.
[34] L. Zhang and B. Yang, "An Efficient Cryptosystem for Medical Image Encryption," *Int. J. Signal Process. Image Process. Pattern Recognit.*, vol. 8, no. 7, pp. 327–340, 2015.
[35] N. F. Elabady, et al., "Improving the Security of Image Encryption by using Two Chaotic Maps," *Int. J. Comput. Appl.*, vol. 108, no. 19, pp. 27–32, 2014.
[36] JOHNSON and E. Alistair, "MIMIC-III, a critical care database," *Sci. data*, 2016.
[37] Johnson AE, et al., "Philips-MIT eICU Collaborative Research Database," *CCM*, 2018. [Online]. Available: http://eicu-crd.mit.edu/.
[38] Boxdicom.com, "Box DICOM Sample Studies," 2016. [Online]. Available: https://boxdicom.com/samples.html. [Accessed: 20-Apr-2018].

Authors' Profiles

Hanya M. Abdallah received her BSC in May 2013, she is currently works as teaching assistant at computer science department, Benha University, Egypt. Her current research interests lie in the development of usable cryptographic security solutions to enhance the information security in cloud computing.

Ahmed Taha received his M.Sc. degree and his Ph.D. degree in computer science, at Ain Shams University, Egypt, in February 2009 and July 2015 respectively. He is currently working as assistant professor at computer science department, Benha University, Egypt. His research interests concern: Computer Vision & Image Processing, Digital Forensics, Security (Encryption – Steganography – Cloud Computing), Content-Based Retrieval.

Mazen M. Selim received the BSc in Electrical Engineering in 1982, the MSc in 1987 and PhD in 1993 from Zagazig University (Benha Branch) in electrical and communication engineering. He is now an Associate Professor at the faculty of computers and informatics, Benha University. His aras of interest are image processing, biometrics, sign language, content based image retrieval (CBIR), face recognition and watermarking.

How to cite this paper: Hanya M. Abdallah, Ahmed Taha, Mazen M. Selim, "Cloud-based Framework for Efficient Storage of Unstructured Patient Health Records", International Journal of Computer Network and Information Security(IJCNIS), Vol.11, No.6, pp.10-21, 2019.DOI: 10.5815/ijcnis.2019.06.02