An Efficient Framework for Storing and Retrieving Unstructured Patient Health Records in the Cloud

Hanya M.Abdallah, Ahmed Taha and Mazen M.Selim
Computer Science department, Faculty of Computers and Informatics, Banha University, Egypt.

Abstract *Migrating Patient Health Record (PHR) to the cloud plays an essential role in the healthcare field. The patient has become able to maintain, access and share his health record among different specialists anywhere and anytime. However, he still suffers from security issues related to storing PHR in the cloud. Giving security to an immense volume of patient's confidential data with high proficiency is necessary. PHR needs to be anonymized or encrypted before being moved to the cloud due to the sensitivity of these medical data. This paper introduces an encryption based framework for storing and maintaining unstructured PHRs by patients and retrieving them by the authorized users. The contents of the PHR are encrypted with two different techniques. The encrypted files are partitioned into a random number of files. These files are of variable number and variable size. After that, the files are stored in a secured cloud storage. 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.*

Keywords: *Cryptography, Anonymization, Security, Cloud Computing, Chaotic Maps.*

Received December 31, 2018; Accepted March 24, 2019

1. Introduction

Patient Health Record (PHR) is a set of critical information that patients keep about their health. This record can be either structured or unstructured. The structured PHR is stored usually in a relational database which has a definite set of attributes. These attributes can be unique-identifiers (e.g., name, phone number, etc.), quasi-identifiers (e.g., age, gender, etc.), sensitive attributes (e.g., disease name), non-sensitive attributes. On the other hand, the unstructured PHR has an internal structure although it is not structured via a predefined scheme. However, it can be 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.[15, 28]. Actually, physicians need to know the complete history of the patient before they could make a proper diagnosis. However, this information is difficult to be maintained. Each patient may consult a number of different physicians during his lifetime. 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.

Recently, it has been found that cloud computing is 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 [18, 30]. 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 [28]. 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 [26].

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 [2]. 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.

Many researchers have been developing different schemes for storing and accessing PHRs in the cloud [2, 3, 23, 26, 27, and 32]. One scheme is based on one level of security [2, 17, and 27]. That is the use of only the encryption and decryption of PHR. Another scheme depends on two levels of security [3, 23, and 32]. It uses encryption and decryption along with controlling the user's access to the encrypted files, but more security is still needed. 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 Section 5.

2. Related Work

Many techniques for securing and preserving data privacy exist. These techniques can be either cryptographic or non-cryptographic [33]. Based on the number of keys being used, cryptographic techniques come into three categories: Asymmetric-key encryption, symmetric-key encryption, and Hashing. On the other hand, non-cryptographic techniques involve anonymization, data sanitization, and randomization. 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.

2.1. 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. [1]. These techniques use operations of generalization, suppression, bucketization...etc. [18, 30]. The above-mentioned techniques are suitable for structured data [35, 36]. 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.

2.1.1. Structured Data Anonymization

In [35] 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 criteria. 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 [36]. 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 [22]. 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.

2.1.2. Unstructured Data Anonymization

Most of the data released today is unstructured. However, there exist a very less previous work that uses anonymization [16, 24, and 34] and appears to be unfeasible.

Gardner *et al.* introduce a method for de-identifying unstructured health data [16]. 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 [34], 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 [24] 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.

2.2. 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 [21]. They are either traditional such as AES, RSA, DES, and BLOWFISH, etc. [8] or advanced such as Homomorphic Encryption, Identity Based Encryption, and Attribute-based Encryption (ABE) with its variations [21, 31].

2.2.1. Structured Data Encryption

Many researchers use Homomorphic encryption [6, 10, and 38] and Attribute Based Encryption (ABE) techniques [13, 17, 23, 25-27, and 32] 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 [3], Alias *et al.* 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 have 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 [32], Shrestha considers 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 [5], 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.

Chennam *et al.* [11] 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 [10] 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.

2.2.2. Unstructured Data Encryption

A system essentially based on the idea that PHR has two types of data (text data, scanned images) is presented in [2]. 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 *et al.* present a framework in [4] 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 [12, 29, and 37]

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 [29], Reddy K. *et al.* 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 [12] by Dai Y. *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 [4] 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 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.

3. 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:

3.1. 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
Figure 1.a) if his role is "patient" or to stage two (see Figure 1.b) if his role is "Physician or hospital or insurance company".

3.2. Stage 1: PHR Upload to Cloud Storage

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

3.2.1. 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.

3.2.2. Patient Health Records

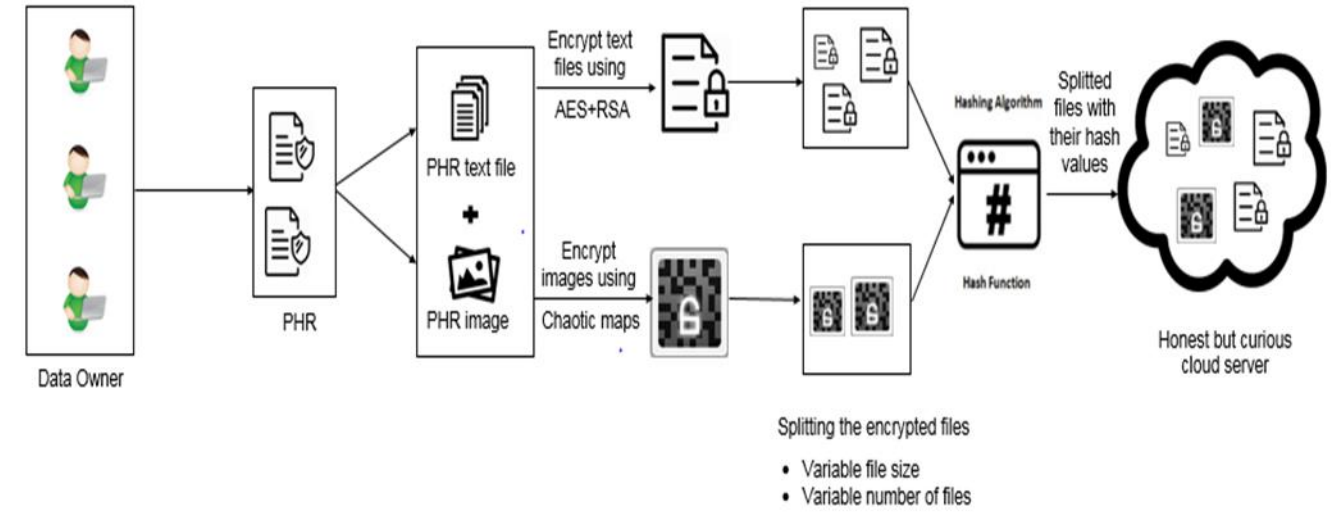
Patient health records are a collection of files that the patient needs to store on cloud. These files may include identification sheet, patient's medical history, laboratory reports, scan reports, X-rays, operative reports, pathology reports, their current medications, progress notes, etc. As can be 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.

3.2.3. Encryption

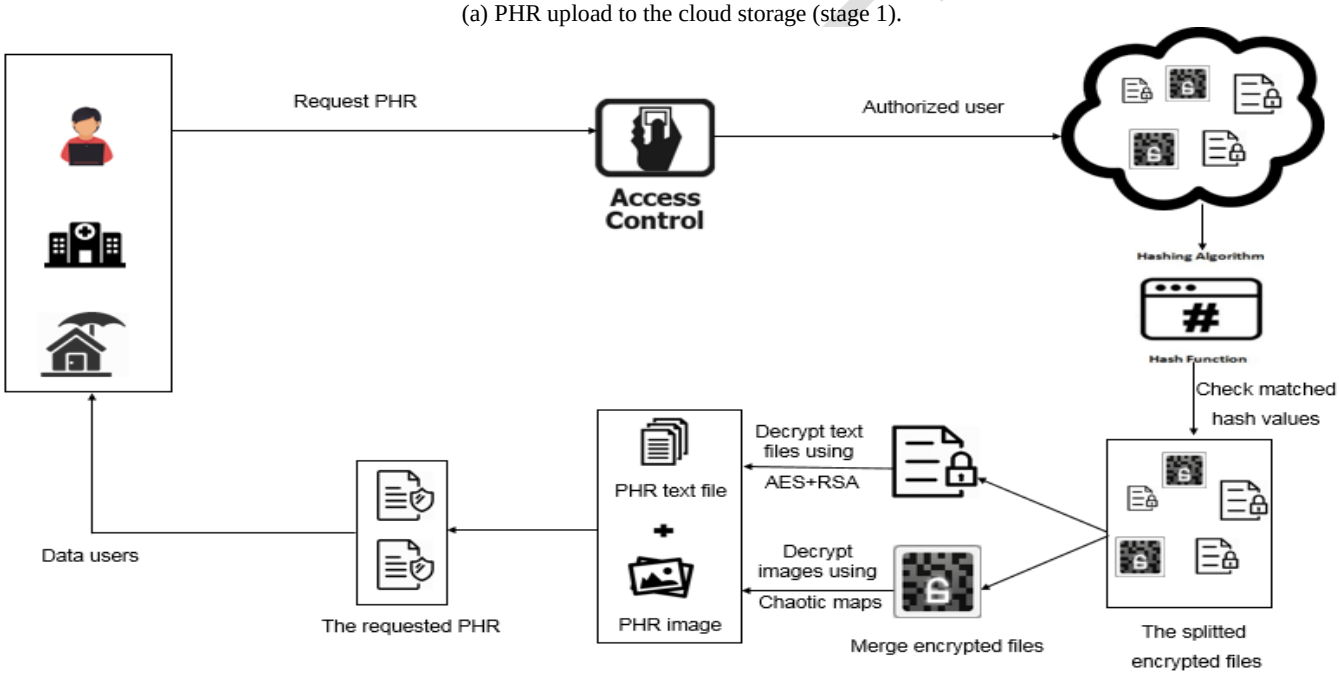
Based on the concept of attribute-based encryption, when the patient tends to encrypt a PHR file before being moved to the cloud, the proposed framework first checks the extension attribute of this file. If it has any of known text data extensions such as 'txt', 'text', 'doc', 'docx', 'pdf', etc., the proposed framework deploys a text-based encryption technique that combines AES and RSA algorithms (see Figure 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 [14] and is applied on colored images but in our work, it is modified to be suitable for gray medical images (see Figure 3).

IAJIT



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



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

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

Text Based Encryption Technique	
Step 1:	Symmetric key of AES with 256 bit length is generated randomly.
Step 2:	Two asymmetric public and private keys of RSA are generated with 1024-bit length.
Step 3:	The uploaded file is encrypted with AES secret key. After encryption, the encrypted file is sent to the split layer before being stored on cloud servers and the original file is unlinked from the temporary directory.
Step 4:	The public key of RSA is used to encrypt the AES key. Then the encrypted AES key is stored to the database.

Figure 2. Text Based Encryption Technique.

3.2.4. Splitting encrypted PHR files

After the 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.

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 [14]

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 [14] for shuffling the rows and columns for each block when: $2.75 < \mu_1 < 3.4$, $2.7 < \mu_2 < 3.45$, $0.15 < \gamma_1 < 0.21$, and $0.13 < \gamma_2 < 0.15$

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

Step 5: Diffuse the pixels of each block using diffusion formula in [14].

$$C_{now} = (P_{now} + Z_{2n} + C_{pre} + P_{pre}) mod 256$$

Figure 3. Chaotic Based Image Encryption Technique.

3.2.5. 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.

3.2.6. 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

3.3. Stage 2: PHRs Retrieval from the cloud storage

As shown in Figure 1.b, this stage involves the following five components:

3.3.1. 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).

3.3.2. Access control

When a data user requests to access a PHR file, the framework 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.

3.3.3. 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.

3.3.4. Merge partitioned files

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

3.3.5. 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. First, 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.

4. 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. All experiments are run on a personal computer with windows 7 with an Intel core i7 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 and split/merge time for individual PHR files. Each one is discussed in the following subsections.

4.1. Dataset

Two datasets are available for patient health records: MIMIC-III [19] and eICU [20]. 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 are collected from [9] and all of size 256×256. It includes Magnetic Resonance Imaging (MRI), Computed Tomography (CT), and X-Ray...etc.

4.2. Security Analysis

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

4.2.1. 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 [4] 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 [4] 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.

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.

4.2.2. 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.

4.2.3. 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.

4.2.4. Availability

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

4.3. Encryption and Decryption Time

Tables 1 and 2 show the results of implementing the proposed framework and the framework presented in [4] 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 [4] 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 [4] 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 [4] 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

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 [4]

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

in the proposed framework is less than the total time for encrypting and decrypting them in scheme [4] by about 20% and 85% respectively. Table 1 also shows 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.

4.4. 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.

5. 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.

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

References

[1] Abouelmehdi K., Beni-Hessane A., and Khaloufi H., "Big healthcare data: preserving security and privacy," *J. Big Data*, vol. 5, no. 1, pp. 1–18, 2018.

[2] Aiswarya R., et al., "Harnessing Healthcare Data Security in Cloud," Proc. Int. Conf. Recent Trends Inf. Technol. ICRTIT 2013, Chennai, India, pp. 482–488, 2013.

[3] Alias A. E. and Roy N., "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.

[4] Arunkumar R. J. and Anbuselvi R., "Enhancement of Cloud Computing Security in Health Care Sector," *Int. J. Comput. Sci. Mob. Comput.*, vol. 6, no. 8, pp. 23–31, 2017.

[5] Au M. H. *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.

[6] Awasthi P., et al., "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*, pp. 509–517, Springer, Singapore, 2019.

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

[8] Bhanot R. and Hans R., "A Review and Comparative Analysis of Various Encryption Algorithms," *Int. J. Secur. Its Appl.*, vol. 9, no. 4, pp. 289–306, 2015.

[9] *Box2016, Box DICOM Sample Studies*. Available at: https://boxdicom.com/samples.html. (Accessed: 20-Apr-2018).

[10] Chandrasekhar S., Ibrahim A., and Singhal M., "A Novel Access Control Protocol Using Proxy Signatures for Cloud-Based Health Information Exchange," *Comput. Secur.*, vol. 67, no.1, pp. 73–88, 2017.

[11] Chennam K. and Mudanna L., "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.

[12] Dai Y., Wang H., and Wang Y., "Chaotic Medical Image Encryption Algorithm Based on Bit-Plane Decomposition," *Int. J. Pattern Recognit. Artificial Intell.*, vol. 30, no. 4, pp. 1–15, 2016.

[13] Deshmukh P., "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.

[14] Elabady N. F., 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.

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

[16] Gardner J. and Xiong L., "An integrated Framework for De-Identifying Unstructured Medical Data," *Data Knowl. Eng.*, vol. 68, no. 12, pp. 1441–1451, 2009.

[17] Gondkar D. A. and Kadam V. S., "Attribute Based Encryption for Securing Personal Health Record on Cloud," *Proc. 2nd International Conference on Devices, Circuits and Systems*, Madrid, Spain, pp. 1–5, 2014.

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

[19] Johnson and E. Alistair, "MIMIC-III, a critical care database," *Sci. data*, 2016.

[20] Johnson AE, et al., "Philips-MIT eICU Collaborative Research Database," *CCM*, 2018.

[21] Kirubakaramoorthi R., Arivazhagan D., and Helen D., "Survey on Encryption Techniques used to Secure Cloud Storage System," *Indian J. Sci. Technol.*, vol. 8, no. 36, pp. 1–7, 2015.

[22] Li J., et al., "A Top-down Approach for Approximate Data Anonymization," *Enterprise Information Systems*, vol.7, no.3, pp. 272–302, 2013.

[23] Li M., Yu S., and Zheng Y., "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.

[24] Li X., and Qin J., "Anonymizing and Sharing Medical Text Records," *Inf. Syst. Res.*, vol.18, no.1, pp. 1–21, 2017.

[25] Qian H., Li J., and Zhang Y., "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.

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

[27] Ramakrishnan N. and Sreerekha B., "Enhancing Security of Personal Health Records in Cloud Computing by Encryption," *Int. J. Sci. Res.*, vol. 4, no. 4, pp. 298–302, 2015.

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

[29] Reddy V. and Fathima A., "Efficient Encryption Technique for Medical X-ray Images using Chaotic Maps," *Proc. IEEE International Conference of Wireless Communications, Signal Processing and Networking*, Chennai, India, pp. 783–787, 2016.

[30] Selvaraj B. and Periyasamy S., "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.

[31] Shabir M. Y., et al., "Analysis of Classical Encryption Techniques in Cloud Computing," *J. Tsinghua Sci. Technol.*, vol. 21, no. 1, pp. 102–113, 2016.

IAJET

[32] Shrestha N. M., et al., "Enhanced E-Health Framework for Security and Privacy in Healthcare System," *Proc. 6th International Conference on Digital Information Processing and Communications*, Beirut, Lebanon, pp. 75–79, 2016.

[33] Singh B., Singh A., and Singh D., "A Survey of Cryptographic and Non-Cryptographic Techniques for Privacy Preservation," *Int. J. Comput. Appl.*, vol. 130, no. 13, pp. 7–10, 2015.

[34] Thavavel V. and Sivakumar S., "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.

[35] Wang W., Chen L., and Zhang Q., "Outsourcing High-Dimensional Healthcare Data to Cloud with Personalized Privacy Preservation," *Comput. Networks*, vol. 88, pp. 136–148, 2015.

[36] Yang J. J., Li J. Q., and Niu Y., "A Hybrid Solution for Privacy Preserving Medical Data Sharing in the Cloud Environment," *Futur. Gener. Comput. Syst.*, vol. 43–44, pp. 74–86, 2015.

[37] Zhang L. and Yang B., "An Efficient Cryptosystem for Medical Image Encryption," *Int. J. Signal Process. Image Process. Pattern Recognit.*, vol. 8, no. 7, pp. 327–340, 2015.

[38] Zhou J., et al., "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.

Hanya M. Abdullah 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 works 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.

IAJET