DOI: https://doi.org/10.31341/jios.44.1.5 UDC 004.76:004.056:61
Open Access Original Scientific Paper

A Cloud Solution for Securing Medical Image Storage

Mbarek Marwan *marwan.mbarek@gmail.com*
Faculty of Sciences, LTI Laboratory
University of Chouaïb Doukkali, El Jadida, Morocco

Abdelkarim Ait Temghart *aittemghart.abdelkarim@gmail.com*
Faculty of Sciences and Techniques, TIAD Laboratory
University of Sultan Moulay Slimane, Beni Mellal, Morocco

Fatima Sifou *fatimasifou@gmail.com*
Faculty of Sciences, LRIT Laboratory
University of Mohammed V, Rabat, Morocco

Feda AlShahwan *fa.alshahwan@paaet.edu.kw*
College of Technological Studies
Public Authority for Applied Education and Training, Shuwaikh, Kuwait

Abstract

Cloud computing is an easy-to-use, affordable solution to manage and analyze medical data. Therefore, this paradigm has gained wide acceptance in the healthcare sector as a cost-efficient way for a successful Electronic Medical Records (EMR) implementation. Cloud technology is, however, subject to increasing criticism because of the numerous security vulnerabilities. In this regard, we propose a framework to protect confidential data through the development of new security measures, including compression, secret share scheme and XOR operation. The primary objective of the proposal is to achieve the right balance between security and usability. To this aim, we divide an image into several blocks and then encrypt each piece separately with different cryptographic keys. To enhance privacy and performance, we suggest DepSky architecture to keep data on various storage nodes. Simulation experiments have been conducted to prove the effectiveness of the proposed methodology.

Keywords: cloud computing, medical image, storage, security, secret share scheme

1. Introduction

Cloud computing was introduced to deal with the scarcity issues of computational resources. This technology uses virtualization technique and pay-per-use concept to make substantial cost savings. More importantly, cloud enables on-demand network access to remote storages, software and services [1]. Therefore, this technolgy

relieves organizations from the expense of building local data centers and managing on-premises applications. Today, a wide range of industrial solutions tailored to meet client's needs are available in the market, such as Google Drive, iCloud, SkyDrive, Amazon S3, Dropbox and Microsoft Azure. Because of this fact, healthcare organizations are increasingly interested in adopting cloud to manage medical records. In this case, a web interface is used to store, share and analyze health records as illustrated in Figure 1.

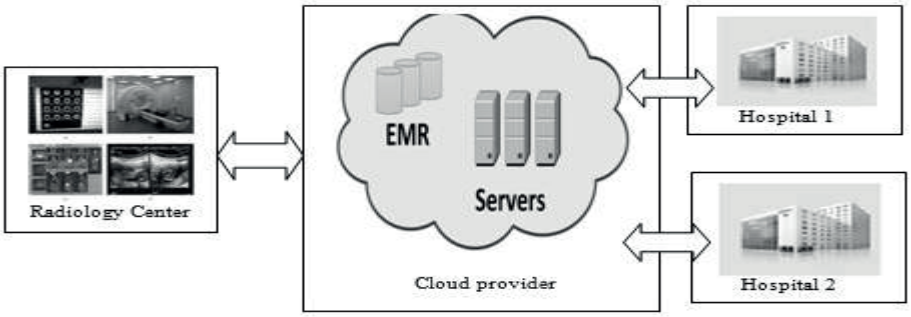

Figure 1. The general architecture of cloud storage in healthcare domain

Cloud involves storing data on virtual servers that are generally managed by third parties. As a result, clients' data are extremely vulnerable to multiple security risks. In light of this fact, we design and develop a secure framework to enhance the data confidentiality in cloud storage. Several methods are proposed for this purpose including Shamir's secret share algorithm, multi-cloud storage, XOR and lossless compression.

The remainder of this paper is organized as follows. Section 2 describes services and deployment models of cloud computing, as well as security challenges. The proposed methodology to meet privacy requirements is presented in Section 3 and 4. Section 5 presents the simulation results of our proposed solution. In section 6, we evaluate the performance and the security of the proposed method. We end this paper in Section 7 and 8 by highlighting the major contributions of this study and discussing future directions.

2. Background Information

In the following section, brief descriptions of different cloud services and deployment models, as well as the state of the art on security challenges, are presented.

2.1. Cloud Services

Cloud computing is introduced to help healthcare organizations for addressing the issue of computational resources shortage. In order to meet increasing customer

demands, cloud providers offer a wide range of cloud services, i.e. software, platform and infrastructure as a service to meet the needs of healthcare organizations. For all cases, clients are charged based on resources utilization, and meanwhile externalize all maintenance tasks. Figure 2 presents the main cloud services in healthcare domain [2], [3], [4], [5].

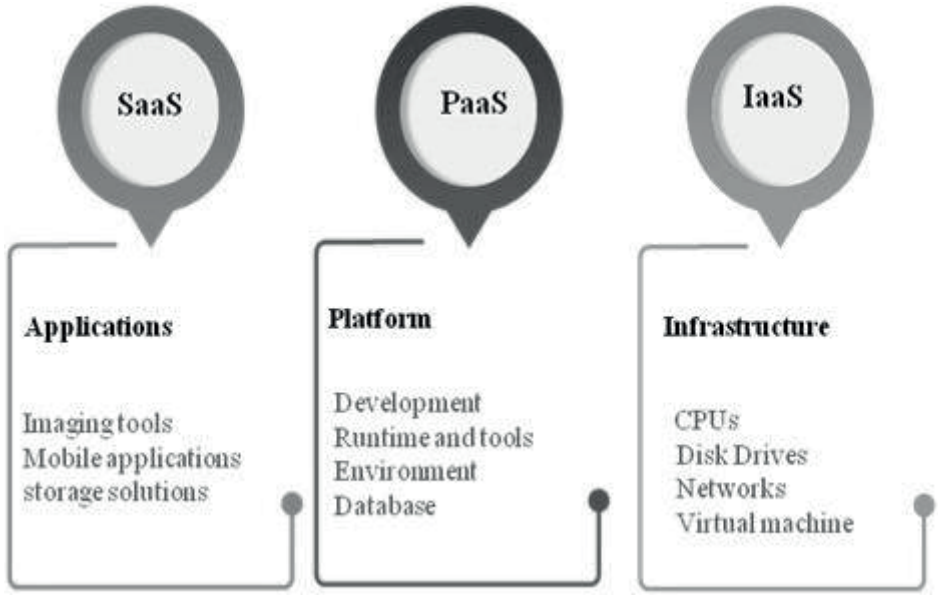

Figure 2. Cloud deployment models

In the SaaS model, clients use remote software and tools to process or safeguard medical data. As an illustration, doctors send a medical image to a cloud provider in order to process it. The third party sends the result to clients through a secure connection to avoid data disclosure. The PaaS model offers necessary tools, such as virtual machine, programming languages and databases, to develop in-house applications. The objective of IaaS is to provide a highly elastic environment that enables on-demand resource capabilities like hardware, storage, servers and network components.

2.2. Security Issues in Cloud Storage

Although the cloud storage has radically revolutionized the way to use IT services, it is still vulnerable to security risks. This is principally due to the fact that data are moved from in-house systems to cloud servers. Besides, cloud computing relies on multiple technologies to offer flexible, on-demand IT services and computational resources to healthcare organizations, including virtualization, Parallel and Distributed System (PDS) and Web 2.0 [6]. However, these techniques are actually the most serious challenges facing cloud computing as they carry security concerns

and risks. As shown in Figure 3, virtualization, data location, storage device and web technology are the main root causes of security breaches.

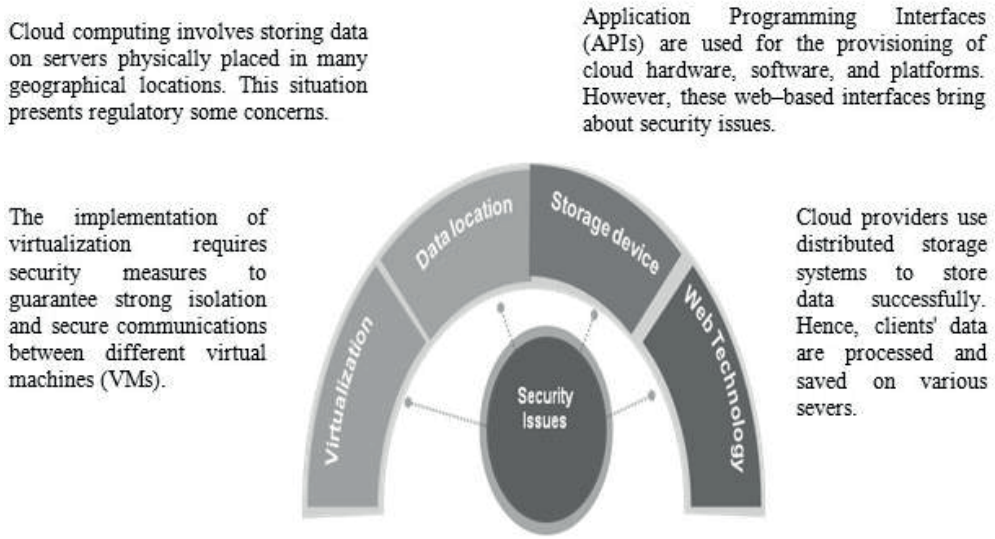

Figure 3. The main security issues in cloud computing

In fact, to incorporate virtualization in cloud infrastructure might pose additional security problems, including VM image sharing, VM isolation, migration and rollback attacks, cross VM side channel attacks, VM escape and hypervisor issues [7], [8]. Besides, it is harder to check the security of outsourced data because data are managed remotely and clients do not often have full control over their data or even worse, they lose control [9]. In cloud storage, critical issues related to management are brought about among which are multi-tenant environment, data recovery vulnerability, improper media sanitization and data backup [10]. Moreover, a malicious user could easily exploit weaknesses in web technology to gain access to clients' data. In this context, many vulnerabilities occur mostly in web-based applications which represent the major obstacles for moving data into the cloud, namely broken authentication and session management, Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), just to name a few [11], [12].

2.3. Privacy Requirements in Cloud Storage

Beside inherent security and privacy problems, clients' data are usually saved and processed in many servers at unknown locations. This leads to additional security issues having to do with access control, privacy, integration, etc. In this section, we formally present different factors that have the most significant influence on the quality of data protection. Basically, there are four crucial factors to consider when outsourcing medical data to an external cloud provider [13,] [14], [15], [16], as shown in Figure 4.

Figure 4. Security and privacy requirements in cloud computing

On the one hand, medical images integrity is seen as top concerns in healthcare sector because they play a vital role in disease diagnosis and treatment. Under these considerations, it is mandatory to implement necessary measures to avoid data corruption and loss. For instance, watermarking is now commonly used to verify the integrity of outsourced data. In a related issue, it is becoming important to encrypt data before transferring them into cloud computing in order to hide personal medical records. In this respect, there are various approaches to cope with data confidentiality, including secret share scheme, visual cryptography, steganography, etc. On the other hand, cloud computing is a multi-domain environment, and then requires different security policies that are tailored for each client. Even worse, heterogeneity and diversity of cloud services necessitate the utilization of fine-grained access control systems. Besides, it is important to implement accounting mechanisms and logs to analyze users and services activities in order to meet the requirements of Service Level Agreements (SLAs). In this regard, identification and authentication ensure that only legitimate subjects can log on to computing resources. Given the distributed nature of cloud computing, the implementation of rigorous authentication tools demands sophisticated methods to capture cloud requirements. For this reason, various techniques are suggested to identify and validate cloud users' identity efficiently, namely Single Sign-On (SSO), Zero Knowledge Proof (ZKP) and Identity-Based Identification (IBI), etc.

3. The Proposed Techniques

As outlined above, the privacy of medical information must be considered during the migration to cloud storage. In this respect, we propose a framework to mitigate security problems in cloud environment. In particular, we suggest an approach to protect the confidentiality of outsourced data to avoid data disclosure. To this point, we propose a multi-cloud architecture, which consists of distributed storage systems, to reduce security problems facing the classical cloud model. To remediate these

weaknesses, medical data will be saved on different nodes to be fully protected against a number of threats and attacks. Technically, our proposed security solution is a combination of a wide range of useful techniques, i.e. compression, secret share scheme, XOR operation and secure socket layer (SSL) protocol. Consequently, this approach is meant to meet security requirements by enforcing confidentiality of medical records.

3.1. Compression

So far, the utilization of health records has grown immensely. In fact, it provides significant support to doctors in ameliorating diagnosis process. Hence, enormous volumes of medical data are daily created to meet the rising demands of healthcare services. For this reason, it is necessary to use compression methods to remove duplicate data, thereby minimizing the total data size. By eliminating unwanted low bit rates, compression techniques enable great savings in bandwidth and storage. However, they might compromise the visual quality of digital records and lead to misdiagnosis. Since medical data accuracy and consistency are critical, it is mandatory to use lossless or near-lossless compression to maintain data integrity and avoid data loss. In this context, the popular DICOM standard encompasses different types of compression algorithms, including JPEG loss and lossless, Run Length Encoding Compression, JPEG-LS, JPEG 2000 and MPEG2 [17]. In this study, we suggest JPEG 2000 algorithm because of its significant features to reduce the required storage capacity and maintain image quality.

3.2. Secret Share Scheme

Basically, distributed storage system implies storing data on multiple virtual servers. In addition, it is important to ensure whether a malicious user can reveal any information about a medical data with an insufficient number of shares. A simple way to achieve this is to use Shamir's Secret Share (SSS) method [18] to distribute a secret among several entities while simultaneously preserving data confidentiality. In this technique, the secret information (M) is partitioned into a predefined number of shares (M_1, M_2,..., M_n) using the threshold scheme A (t, n).

Definition 1. Let m and n be positive integers, $m \leq n$. A m-out-of-n threshold scheme is a technique for distributing a secret among a set of n members in such a way that any m members can reconstruct the value of the secret data, but no group of $k-1$ or fewer can obtain any information about the secret. Simply put, this method satisfies the following condition:

1) Correctness: The secret data s is uniquely computed by any m shares from $\{s_1,\ldots, s_n\}$ and there exists an efficient algorithm to determine s from these m shares.

2) Privacy: Even though the unauthorized users gain access to $m-1$ shares from $\{s_1, \ldots, s_n\}$, they cannot reveal any information about the secret s. In fact, the probability distribution of $m-1$ shares is independent of s.

Definition 2. Let a_0, a_1, …, a_n and y_0, …, y_n be elements of a field $\mathbb{R}$. Then there exists a unique polynomial $g \in F[X]$ with $\deg(g) < n$ such that $g(a_i) = y_i$ for $i = 1, \ldots, n$.

The Lagrange interpolation polynomial can be computed as the following sum [19].

$$g(x) = y_0 b_0(x) + \ldots + y_n b_n(x) \quad (1)$$

In this case, the base polynomials $b_i(x)$ is defined by the following relation.

$$b_i(x) = \prod_{j=1, i \neq k}^{t} \frac{(x - a_j)}{(x_k - a_j)} \quad (2)$$

Normally, this polynomial can be represented as the matrix equation $A \cdot x = y$, where A is the Vandermonde matrix.

Given distinct $x_1, \ldots, x_k \in F$, and the coefficients $a_0, \ldots, a_{k-1}$ of the interpolating polynomial g, where

$$g = \sum_i a_i . x^i \in F[X], \text{ where } g(x_i) = y_i. \quad (3)$$

The interpolation polynomial can be represented by the following matrix equation [20]:

$$\begin{pmatrix} 1 & x_1 & \ldots & x_1^{k-1} \\ 1 & x_2 & \ldots & x_2^{k-1} \\ & & . & \\ & & . & \\ & & . & \\ 1 & x_k & \ldots & x_k^{k-1} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ . \\ . \\ . \\ a_{k-1} \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ y_k \end{pmatrix} \quad (4)$$

Theorem 1. For a Shamir's (t, n)-threshold scheme, any subset of up to *t-1* shares does not reveal any information about the original value.

Proof

Suppose we have *t-1* shares (x_i, y_i), each potential candidate secret *s'* is represented by a unique polynomial f (.) of degree *t-1* satisfying the condition: $f(0) = s'$, for $s' \in \mathbb{Z}_p$. Based on the representations of the Lagrange polynomial, there is an equal probability of $Pr[s = s']$ for all $s' \in \mathbb{Z}_p$.

3.3. Secure Socket Layer

Generally, it is necessary to secure communication between healthcare organizations and cloud storage providers to avoid data disclosure. To this aim, we typically use Secure Socket Layer (SSL), named also Transport Layer Security (TLS), to maintain data protection during the transfer and storage process. This scheme uses a set of security services to ensure that information cannot be intercepted. Among these services are confidentiality, accuracy, integrity and non–repudiation [20]. This is achieved by using symmetric encryption cryptosystem to maintain data privacy. It also relies on digital signatures to enable non–repudiation mechanisms. Besides, this protocol uses message digest as an integrity checking tool. Practically, we implement this protocol using Java Secure Socket Extensions (JSSE) that is included in the JDK software.

3.4. Multi-cloud Environment

In general, moving medical data to cloud storage would reduce expenses and improve efficiencies. Beside these remarkable advantages, the utilization of cloud services exposes medical data to numerous threats and risks, though. Hence, confidentiality and privacy have to be addressed when implementing cloud in healthcare domain. To some extent, multi-cloud environment can drastically reduce the risk presented by cloud computing [21]. Since data are stored on different nodes, this new model is meant to enhance confidentiality and reliability, as well as reducing the risk of data corruption and loss. Recently, due to the increasing utilization of cloud services, many multi-cloud models have been developed. This solution uses a set of different techniques and mechanisms to improve security, effectiveness and performance. For example, Byzantine Fault Tolerance (BFT) achieves high security level by using replication method, while Redundant Array of Cloud Storage (RACS) uses the RAID5 technique to prevent data loss. Furthermore, DepSky, InterCloud Storage (IC-Store) and High Availability and Integrity Layer (HAIL) rely on distributed storage systems to efficiently spread out data over multiple storage pools. This approach ensures security and prevents lock-in vendor issue. Based on the implemented security mechanisms, DepSky is the most efficient model to protect the confidentiality of clients' data in off-site storage systems [22]. As illustrated in Figure 5, DepSky offers four secure storage pools and relies on two algorithms, DEPSKY-A and DEPSKY-CA, to achieve a high level of security [23]. In such a model, the first algorithm is used to duplicate data stored in cloud computing. Meanwhile, the second one performs data encryption.

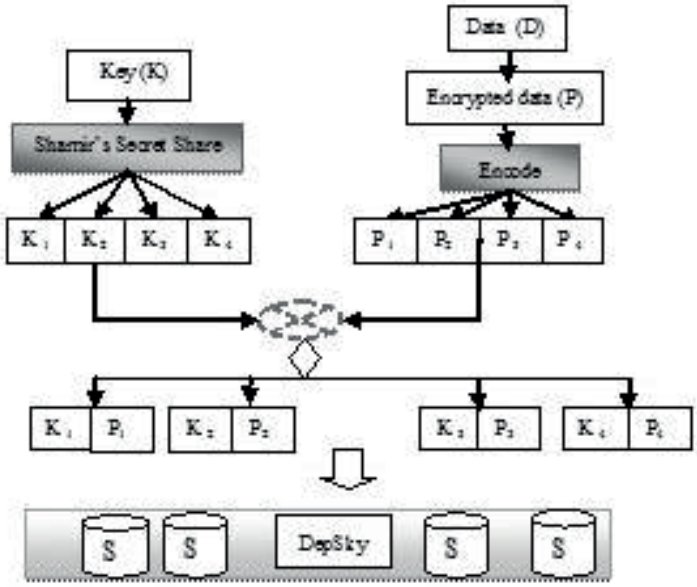

Figure 5. Fundamentals of the DepSky architecture

4. Proposed Framework

Broadly speaking, data security has consistently been a major issue in cloud storage. In fact, security problems are not addressed enough in this new paradigm. In existing frameworks, many encryption methods are used to reduce risks, vulnerabilities and potential consequences. Nevertheless, these methods have a negative impact on the performance during the utilization of cloud storage. Currently, conventional cryptographic techniques, such as RSA, DES, AES, Triple DES and Blowfish, are used to protect medical data. Unfortunately, all of these techniques involve CPU and Memory Intensive operations to perform data encryption. In this paper, we propose a simple and efficient approach to reduce threats and weaknesses of cloud storage. Such a solution can offer healthcare organizations on-demand resources to save information contained within electronic health records. It will undoubtedly assist doctors and nurses in making decisions.

4.1. An Overview of the Proposed Framework

In order to protect data, we propose an architecture that consists of three main components, i.e. Clients, CloudSec and CloudSlave. In such a scenario, CloudSlave provides a scalable storage system to save medical data remotely. In parallel, CloudSec module is used as a trusted interface between clients and service providers. This proxy component offers necessary security mechanisms to prevent data disclosure during the storage process. Essentially, CloudSec module allows clients to upload and download their data, as well as managing metadata. Furthermore, we use a MasterIndex database hosted in the CloudSec to safeguard patients' identifying information and ensure that these sensitive data remain protected and accessible only by authorized users. This is meant to ensure data privacy and more specifically to achieve a certain degree of anonymity and unlinkability. Thus, we use pseudonym techniques to map the real username to prevent the ability to link between the owners and associated medical records. Besides, this module is responsible for authenticating the cloud consumers, typically by using login and password. To this objective, we suggest the SSL protocol for enabling a secure connection between clients and CloudSlave. The CloudSec module offers necessary security mechanisms to deal with threats and vulnerabilities of cloud storage. In this regard, we combine several security techniques, such as secret share scheme, XOR and compression, to avoid data disclosure when storing data in the CloudSlave. For security purpose, we suggest DepSky architecture as a distributed storage system.This solution guarantees that outsourced data are spread across four storage pools in order to enhance data confidentiality. The third component of our proposed architecture refers to healthcare organizations and imaging centers which produce or use digital records. In such a scenario, the role of the CloudSec module is twofold. (1) It sends medical records to the CloudSlave system, and then saves them safely. Basically, this is achieved by dividing data into a number of portions and then encrypted them. (2) It offers access control as a

service to prevent unauthorized access to clients' medical records. Figure 6 gives an overview of the proposed architecture for securely outsourcing data storage.

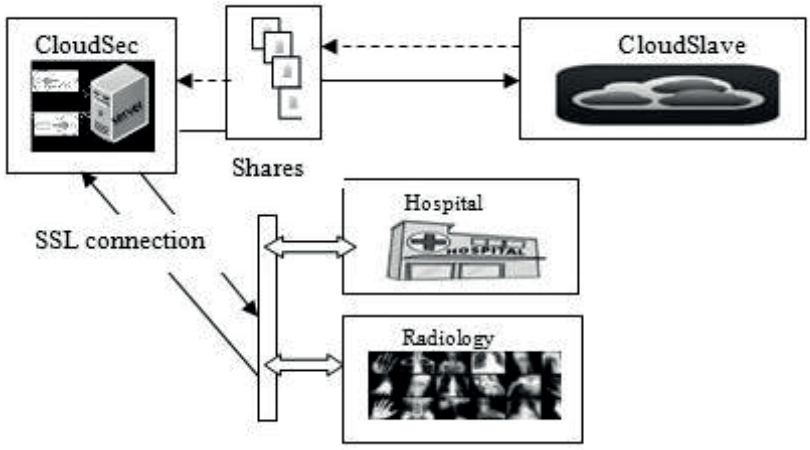

Figure 6. General overview of the proposed architecture

4.2. The Fundamentals of the Proposed Framework

In the proposed methodology, CloudSec is an essential element of the suggested architecture to deal with cloud security concerns. Indeed, it is responsible for encrypting data before storing them into a set of cloud servers, i.e., CloudSlave. For this reason, CloudSec uses compression algorithm to reduce the amount of physical space occupied by medical records. In this respect, we use JPEG200 algorithm [24] as a lossless method to carry out compression process. Next, CloudSec splits the secret medical image into four pieces to enhance security. To provide perfect secrecy, each share is encrypted using XOR operation and the encryption key. The latter is generated from an input key. In practice, we rely on Shamir Secret Share (SSS) scheme to generate four sub-keys from an initial key. Note that for every image, we compute the corresponding encryption keys relying on client's SSN (Social Security Number), thereby creating sub-keys. Since the management of public keys remains a burden, we avoid the need for public-key authority in our proposal thanks to XOR and SSS method. For example, radiology center creates a patient DICOM image, and then, send it to the CloudSec securely using SSL connection. In this case, clients use a web interface to store, retrieve and modify data from the cloud storage according to their access illegibility. In this architecture, CloudSec converts the medical image from DICOM format to JPEG through JPEG 2000 compression algorithm. Subsequently, we split each digital record into four pieces to improve security. Furthermore, each portion is encrypted by using XOR operation to enhance data protection and prevent data disclosure. In the same line, CloudSec module creates the index associated to each piece. This is useful to facilitate the retrieval and storage of medical images. The choice of this approach is motivated by several reasons. First, we benefit from an easy key management

technique since it permits deriving encryption key from SSN without need to implement and deploy a Public Key Infrastructure (PKI). Second, this system uses simple methods for data protection to avoid heavy computational costs. In fact, healthcare professionals should have the ability to access remote medical data rapidly, especially during emergency scenarios. The core principles of the proposed approach are illustrated in Figure 7.

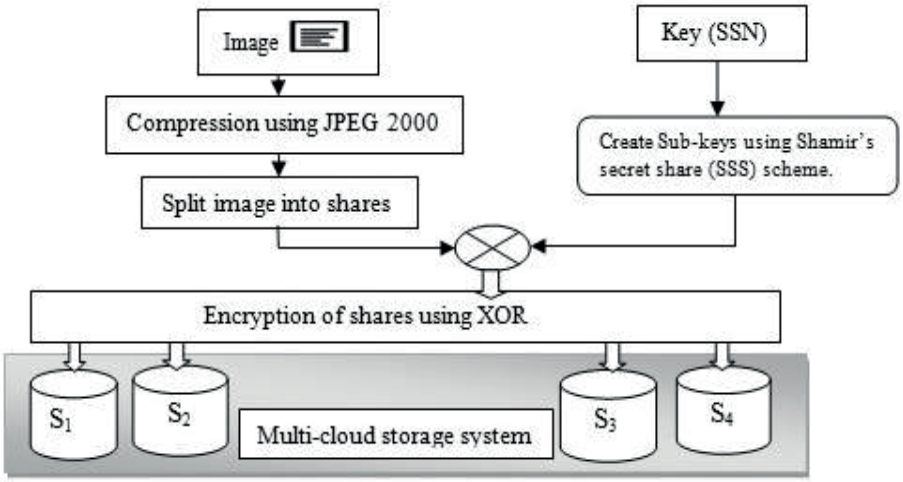

Figure 7. The proposed methodology for data protection in cloud storage

To summarize, the proposed methodology provides an efficient and simple method to promote cloud storage in the healthcare domain. This would help medical providers to cut costs and improve operational effectiveness. After successfully authenticating with the cloud provider, the consumers start the storage process by encrypting their data as presented in Algorithm 1.

Algorithm 1. EncryptImage

1: Input: M, SSN, where M is DICOM medical image and SSN is the social security number.
2: Output < S_1, S_2, S_3, S_4, $index_1$, $index_2$, $index_3$, $index_4$ >
3: $M_1 \leftarrow$ ConvertToJpeg (M) //Convert DICOM images into JPG format.
4: $M_2 \leftarrow$ Compress (M_1) //Set compression parameter.
5: {M_{21}, M_{22}, M_{23}, M_{24}} $\leftarrow$ Split (M_2)
6: {K_1, K_2, K_3, K_4} $\leftarrow$ SSS (SSN) //Create sub-keys using SSS method.
7: S_1 = Encrypt (M_{21}, K_1), S_2 = Encrypt (M_{22}, K_2), S_3 = Encrypt (M_{23}, K_3), S_4 = Encrypt (M_{24}, K_4).
8: {$index_1$, $index_2$, $index_3$, $index_4$} $\leftarrow$ CreateIndex (S_1, S_2, S_3, S_4)
9: MatserIndex $\leftarrow$ {$index_1$, $index_2$, $index_3$, $index_4$, SSN} //Save index in the MasterIndex database
10: Return < S_1, S_2, S_3, S_4, $index_1$, $index_2$, $index_3$, $index_4$>

Moreover, this framework allows healthcare professionals to retrieve their medical records outsourced to cloud storage. For this reason, clients send a request to CloudSec after authentication. The CloudSec module first retrieves the index related to the patient's image. This approach is useful to locate each stored portion in the cloudSlave. Accordingly, the latter module sends four pieces of medical record to the CloudSec. In the decryption process, we apply XOR operation on these pieces with the same sub-keys. Finally, CloudSec merges these portions and decompresses the result to reconstruct the original image. In this architecture, CloudSec uses SSL connection to send back the stored image to the client. In this context, Algorithm 2 is used to decrypt and download encrypted images, and then send them to the owner.

Algorithm 2. DecryptImage

1: Input: SSN, where SSN is the social security number.
2: Output < M >
3: $\{index_1, index_2, index_3, index_4\} \leftarrow$ MasterIndex [SSN] // Retrieve Index from MasterIndex database.
4: $\{K_1, K_2, K_3, K_4\} \leftarrow$ SSS (SSN) //Create Sub-keys for decryption.
5: $\{S_1, S_2, S_3, S_4\} \leftarrow$ CloudSlave [$index_1$, $index_2$, $index_3$, $index_4$] //Retrieve four portions associated to the stored image.
6: M_{11} = Decrypt (S_1, K_1), M_{12} = Decrypt (S_2, K_2), M_{13} = Decrypt (S_3, K_3), M_{14} = Decrypt (S_4, K_4)
7: M_2 = Merge (M_{11}, M_{12}, M_{13}, M_{14})
8: M = Decompress (M_2)
9: Return <M>

4.3. Key Generation Procedure

For encrypting data, this method uses essentially a polynomial function to create a fixed number of shares. Basically, this involves selecting (t-1) random, independent coefficients f_1, f_2,..., f_{t-1}. As illustrated in Algorithm 3, the couple (x_i, y_i) typically represents each share, where $x_i = 1,\ldots, n$ and $y_i = f(x_i)$.

Algorithm 3. CreateShares

1: Input: s, k, n, where *s* is secret data $s \in \mathcal{F}$ finite field, *k* is threshold and *n* is the number of shares.
2: Output: $s_1, s_2, \ldots, s_n$, where s_i are shares
3: Set $f_0 = s$
4: Generate uniform coefficients $f_1, \ldots, f_{k-1} \in \mathcal{F}$
5: Construct the polynomial $f(x) = f_0 + f_1 x + \ldots + f_{k-1} x^{k-1}$
6: Evaluate the polynomial $s_i = f(i)$, $(i = 1,\ldots, n)$
7: Return < s_i >

In practice, since we use Lagrange polynomial and (k, n)-threshold scheme to recover the secret data, this scheme requires at least *t* shares. In the present work, we rely on Algorithm 4 to recover the secret data successfully.

Algorithm 4. ReconstructionShares

1: Input: shares s_{t_1}, ..., s_{t_k} where $s_t \in \mathcal{F}$ finite field and $t \in 1, \ldots, n$.
2: Output: secret data *s*
3: Calculate coefficients β_i using relation 2
4: Determine $f(0) = s_{t_1} \beta_{t_1} + \ldots + s_{t_k} \beta_{t_k}$
5: Return $s = f(0)$

5. Simulation and Results

To illustrate the proposed method, we design and develop a cloud application that provides remote storage of digital records. The primary purpose of this framework is to comply with the security and privacy requirements.

5.1. Generation of Cryptographic Keys

The proposed framework relies on Shamir's Secret Sharing (SSS) method in the process of generating keys for cryptography. In this case, we divide the Social Security Number (SSN) of each patient into many shares to create sub-keys. For example, we generate four cryptographic keys from the SSN='4598679' using the SSS technique with the threshold (2, 4). For this reason, we implement the SSS algorithm using Java, as depicted in Figure 8.

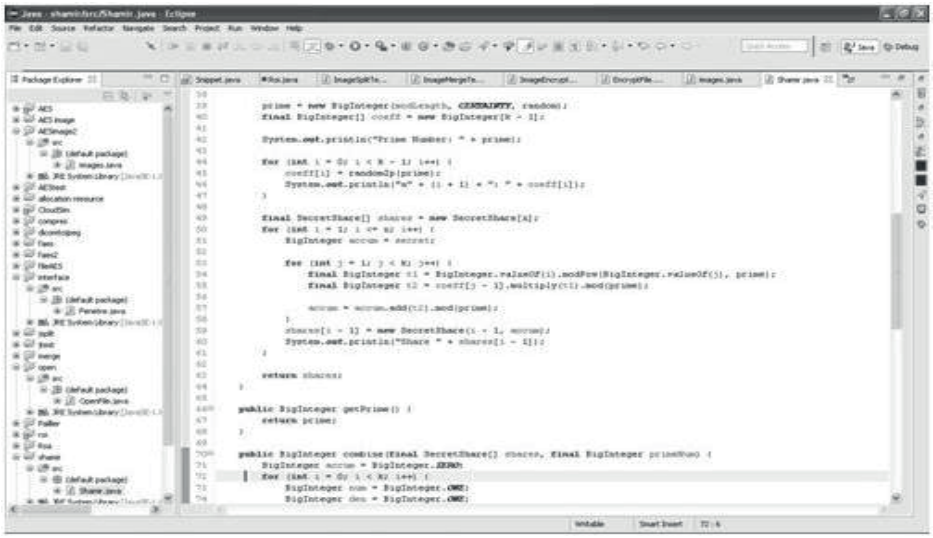

Figure 8. Creating cryptographic keys using the SSS method

Accordingly, we get the following cryptographic keys using the SSS method with the threshold (2, 4) and the prime number: 12164513.

- key 1= 4584532
- key 2= 4570385
- key 3 = 4556238
- key 4 = 4542091

In the same line, to get the secret value, we combine only two shares by using Algorithm 4.

5.2. Data Encryption Approach

The proposed image encryption scheme is implemented using Java as a programming language and MySQL as a database. In this application, clients need to enter their medical information for a successful authentication, as shown in Figure 9.

Figure 9. Client's medical information

In the same line, users have to select a medical image that they want to send to off-site storage systems. For this reason, they commonly rely on the web interface presented in Figure 10.

Figure 10. Selecting an image to be uploaded into the cloud storage

Fortunately, Java programming language technically encompasses powerful libraries to process and handle images. In this regard, Table 1 summarizes the main java functions, which are used to develop the proposed framework.

Functions	**Objective**
ImageIO.getImageReadersByFormatName	This function returns an iterator containing all currently registered ImageReaders that claim to be able to decode the named format.
DicomImageReadParam	This function constructs the ImageReadParam object for the DicomImageReader with the default settings.
SetCompressionType("JPEG2000") SetEncodingRate(2.0) SetLossless(true)	These functions compresse an image using JPEG2000 and set parameter rate.
BufferedImage imgs[] = new BufferedImage[chunks] imgs[count] = new BufferedImage(chunkWidth, chunkHeight, image.getType()) Gr.drawImage(image, 0, 0, chunkWidth, chunkHeight, chunkWidth * y, chunkHeight * x, chunkWidth * y + chunkWidth, chunkHeight * x + chunkHeight, null)	These functions split an image into multiple shucks.
FinalImg.createGraphics().drawImage(buffImages[num], chunkWidth * j, chunkHeight * i, null)	This function creates sub images.
Final BigInteger t1 = BigInteger.valueOf(i).modPow(BigInteger.valueOf(j), prime) Final BigInteger t2 = coeff [j - 1]. multiply(t1). mod(prime) accum = accum.add(t2).mod(prime)	These functions calculate Shamir's coefficients and generate shares.

Table 1. The essential Java functions and their main roles

In order to validate effectiveness of the proposed method, we use the DICOM image named OT-MONO2-8-hip as a sample, which is downloaded from [25]. Therefore, there is a need to convert DICOM image to JPEG format. In this respect, we apply JPEG 2000 (JP2), which is a lossless image compression standard. The key idea of this technique is to reduce the size of medical images. To achieve this, we use the Java program to implement JPEG 2000 algorithm to reduce image file size without

reducing image quality, as shown in Figure 11. Indeed, the used method is a lossless tool to compress medical data, and hence maintains a great visual quality. The primary purpose of implementing 2000 algorithm is to achieve a balance between security and usability.

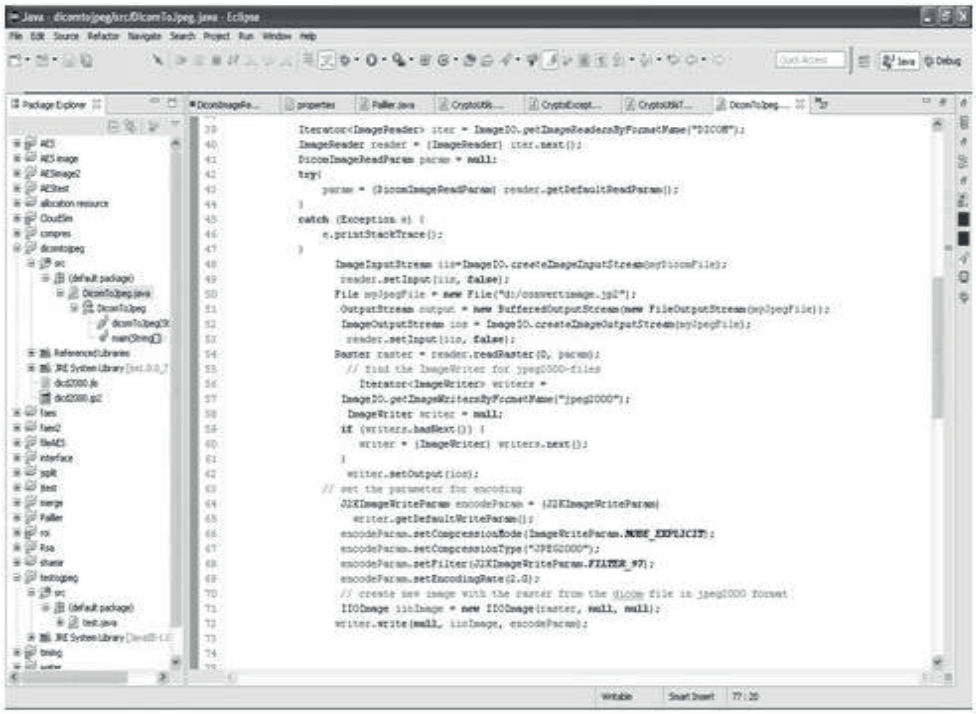

Figure 11. JPEG2000 compression using Java functions

Next, we use Honey view tool to display the converted image, as illustrated in Figure 12.

Figure 12. Display of the compressed image using Honey tool

Subsequently, we divide the medical image into four sub-images to enhance data security. Figure 13 presents the source code used for accomplishing this objective.

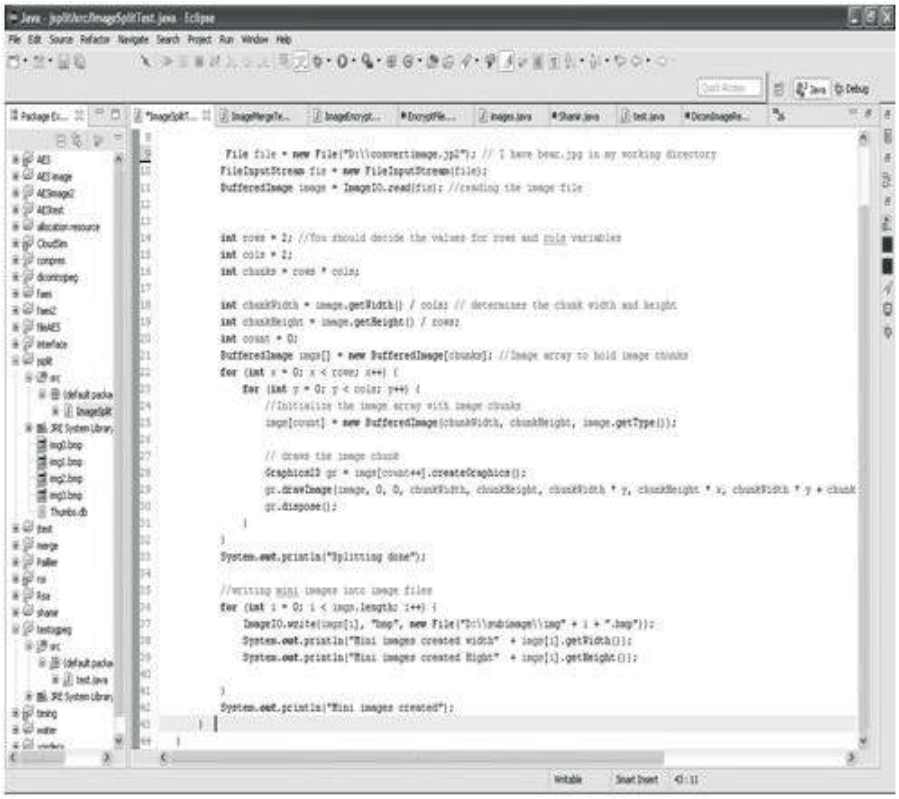

Figure 13. Creating four pieces from an image using Java

As a result, the secret image is split into four images, as depicted in Figure 14. Of course, this method prevents unauthorised users to read any information about the original image from one piece of medical data.

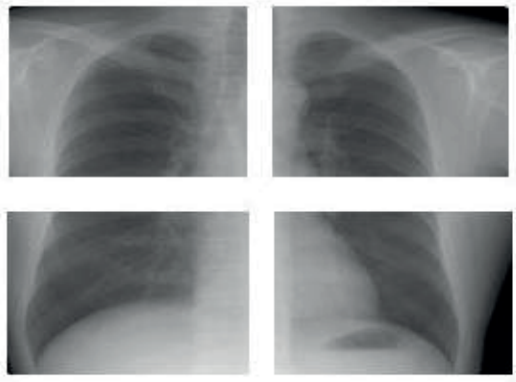

Figure 14. The generated sub images from the secret image

For the security purpose, we apply XOR operation on generated sub-images. In this study, we set 'mbarek' as password to generate sub-keys using SSS method. To this task, we use the Java program presented in Figure 15.

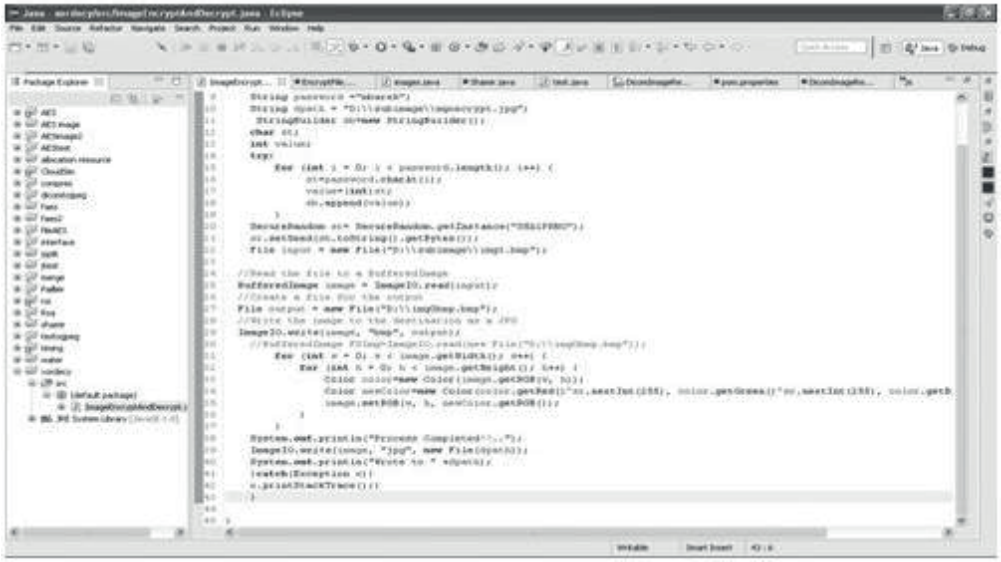

Figure 15. Applying XOR operation on an image

After encryption process, we get four encrypted images. Referring to Figure 16, the simulation results prove that the proposed method protects medical data against authorized users. In fact, this method is able to dissimulate the visual information of the original image, thereby ensuring confidentiality.

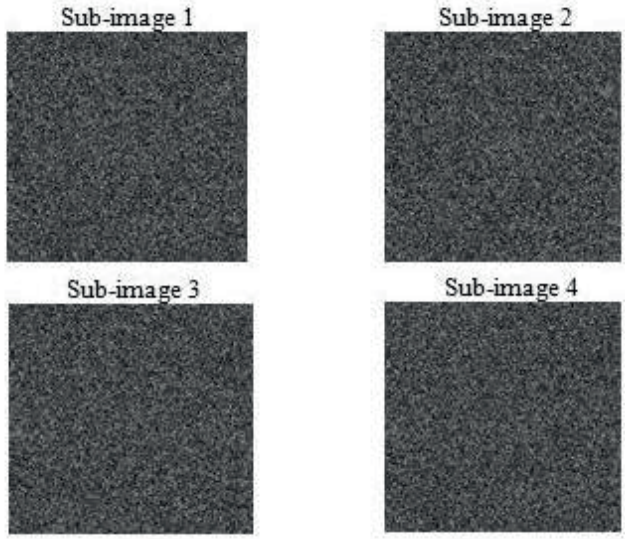

Figure 16. Encrypted image using XOR technique

Similarity, we use private key to decrypt each segment independently. After all shares are encrypted, the original image can be recovered only when four segments are combined together, as shown in Figure 17.

Figure 17. The recovered image

Furthermore, we perform a more comprehensive evaluation of the proposed technique. Specifically, we apply the proposed encryption method on two medical images as shown in Figure 18 and Figure 19.

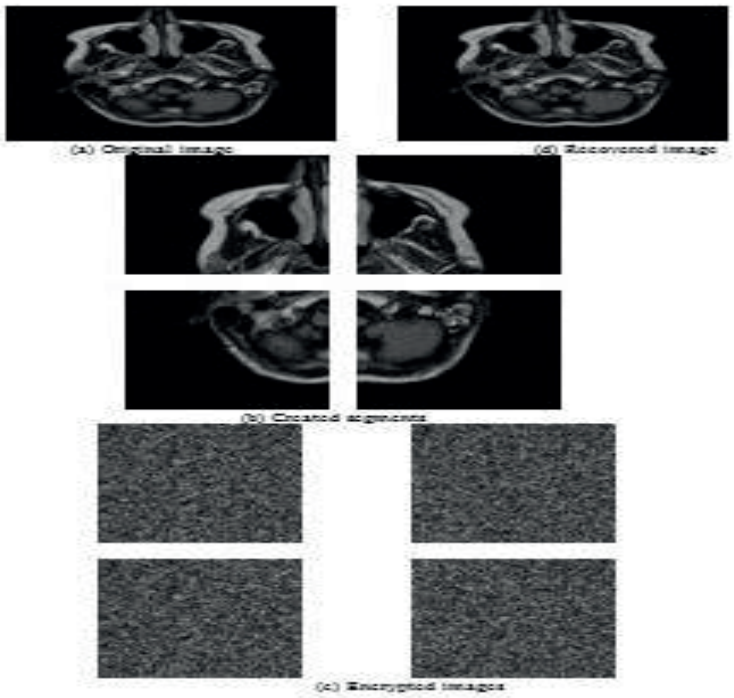

Figure 18. Using the proposed method for encrypting medical image 1

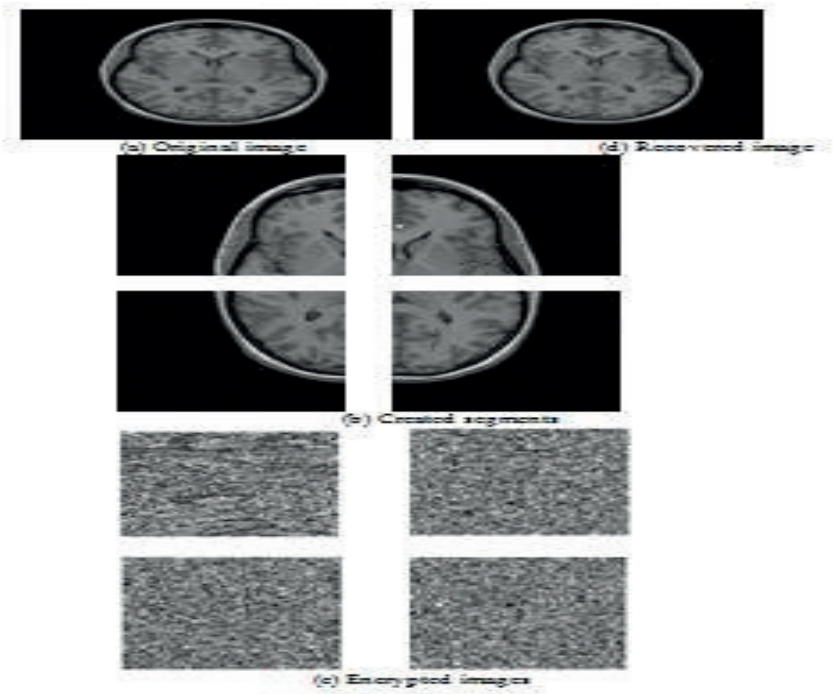

Figure 19. Using the proposed method for encrypting medical image 2

Simulation results show that the method proposed in this paper can provide the appropriate confidentiality level required for the security requirements in the cloud computing.

6. Security and Performance Analysis

This section provides the methodology to evaluate the proposed method and measure performance in order to support theoretical approaches. In this respect, we use running time complexity and histogram to demonstrate that our algorithm improves performance with respect to privacy constraints.

6.1. Theoretical Analysis

The proposed method provides a lightweight scheme that combines data fragmentation and XOR operations. In this regards, we give the following formal definition of this method.

Definition 3. Let S be the secret image. Now, if S is divided into n different shares $E_1,\ldots, E_n$ and then XORed with private keys $K = \{K_1,..,K_n\}$, the following conditions must be met:

- By XORing all n shares with the private K keys $K_1,.., K_n$, we create the encrypted image $S' = \{E_1 \oplus K_1, E_2 \oplus K_{2, \ldots,} E_n \oplus K_n\}$.

- The secret cannot be disclosed by insufficient number of shares.
- The XORed up result of any *n* shares gives no clue about the secret.
- The secret image is visually recovered by XORing all *n* shares with the right keys K_1,.., K_n.

Lemma 1. The XORing up result of every share with the private key K gives no clue about the secret image XOR $((E_1 \oplus K_1, \ldots, (E_t \oplus K_t))$, t = n.

Proof

Let *S* be an image split into *n* shares then XORed with a private key K as S' = $\{E_1 \oplus K_1, E_2 \oplus K_{2,\ldots,} E_n \oplus K_n\}$. In this case, the decryption process requires that the keys be applied in reverse order:

$$\text{XOR } ((S'_1 \oplus K_1, \ldots, (S'_t \oplus K_t)), \text{ t = n.}$$

Therefore, the secret *XOR (S, K)* cannot completely be recovered without applying the required keys.

6.2. Histogram

The primary objective of the histogram analysis is to represent the confusion and diffusion properties in encrypted image. In fact, it indicates the number of pixels in an image at each different intensity value and statistical features of pixels value distribution. In general, the histogram of the encryption image is always uniform for a secure encryption method. Figure 20a is the histogram of image 2, and Figure 20b–e is the histograms of encrypted shares.

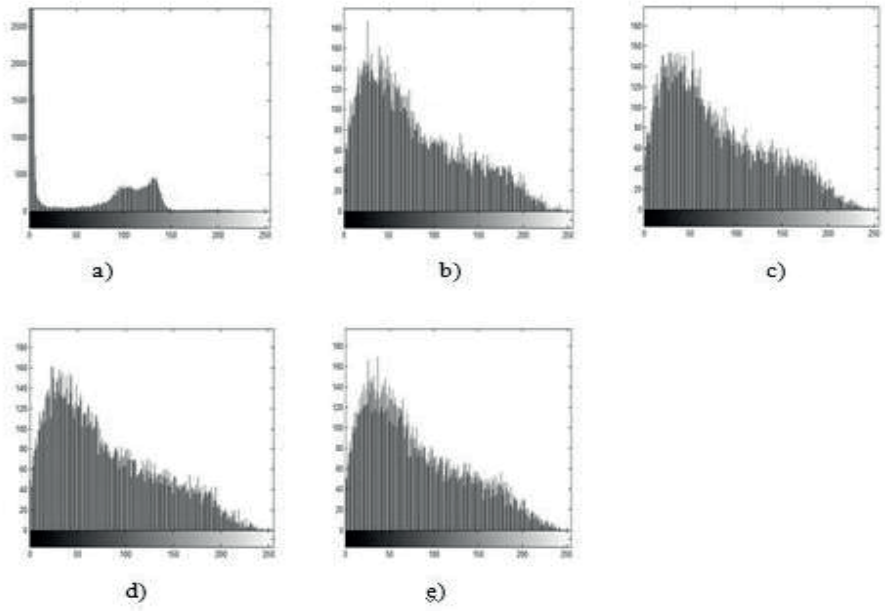

Figure 20. Histograms: (a) plain images; (b-e) encryped shares

The Simulation results show that all the histograms of Figure 20b–e are distributed uniformly, which are totally different from image 2's. Hence, the proposed method can effectively resist the statistical attacks.

6.3. Performance

The execution of our algorithm is composed of two phases. In the first phase, we split an image into four shares. In the second phase, we use a private key and XOR operation to encrypt the secret image. In spite of the proposed method including two phases, it does not require complex mathematical operation for data encryption. Hence, XOR-based approach can outperform the classical encryption techniques. For instance, we implement our proposed method and AES (Advanced Encryption Standard). Basically, there exist three variants of AES that are based on different key sizes (128, 192, and 256 bits). In this study, we use AES-128 to encrypt the secret image. Practically, the plain image is split into data blocks of size 128 bits. In the same line, we choose the encryption key with size 128 bits for both XOR and AES encryption. We implement these algorithms using Java in a PC DELL LATITUDE E6500 with the following specifications: Intel(R) Core(TM) 2 Duo CPU P8400 @ 2,226 GHZ with 1984 MB RAM. For AES cryptosystem, we use the following Java libraries and functions, as shwon in Table 2.

Libraries	java.security javax.crypto
Functions	cipher cipher=cipher.getInstance("AES") keygen=KeyGenerator.getInstance("AES") key key=keygen.generateKey() cipher.init(Cipher.ENCRYPT_MODE, key) cipher1.init(Cipher.DECRYPT_MODE, key);

Table 2. Java functions for AES implementation

Here, we have taken five random image samples for encryption by using AES and XOR algorithm. Table 3 shows the execution time required by different image sizes.

	AES (ms)	**XOR-based method (ms)**
Image (1024x1024)	96	80
Image (768x768)	87	65
Image (512x512)	75	56
Image (256x256)	52	40
Image (225x225)	49	30

Table 3. The execution time for image encryption using AES and XOR-based method

At the same time, we use the Figure 21 to present the findings in a comprehensible format so that they can easily be analysed.

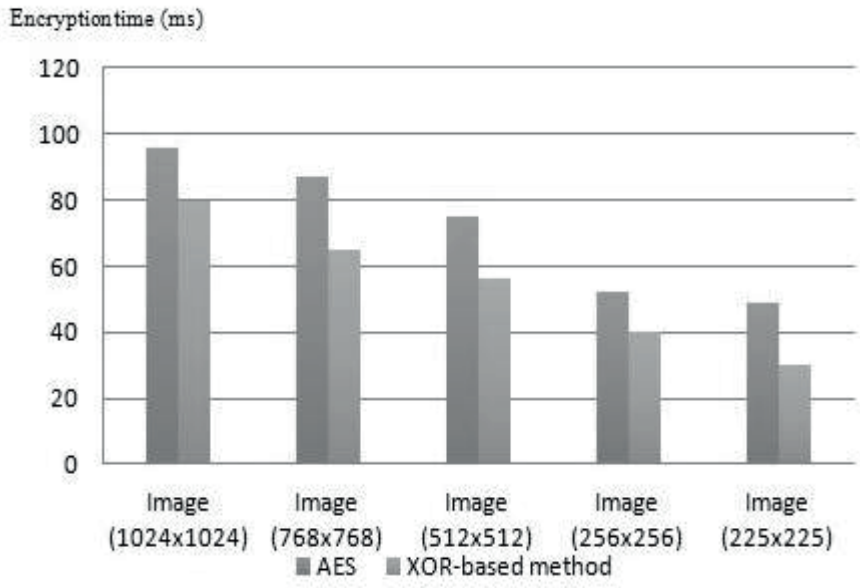

Figure 21. Encryption time for different images

Expectedly, the proposed method is significantly faster than the AES algorithm. Consequently, the XOR-based method is an efficient approach to balance the tradeoff between security and performance.

7. Discussion

Cloud computing is a new model for using IT services without installing in-house hardware and software. Basically, remote resources are rapidly provisioned and reassigned according to the consumer's needs. The main features of this technology are automated updates on software, flexibilty in capacity and availability. However, as mentioned above, security has been one of the most challenging issue to be considered for successful implementation of cloud computing. In this context, notable efforts have been made to increase security measures in cloud services. Many of these efforts have focused on the adoption of traditional cryptographic techniques for protecting confidential data. Although these methods are effective for data protection, many challenges and difficulties still face the utilization of these techniques in cloud environments. In this respect, we propose a lightweight cryptography method for providing data confidentiality in cloud storage. The proposal relies only on lossless compression and XOR operation for preventing accidental data disclosure. This strategy aims at addressing the trade-off between security and performance in cloud storage, especially for medical images. Table 4 presents a brief comparison between the method presented in this paper and existing security measures.

8. Conclusion

Cloud based services have gained notoriety in the latest technology innovations and emerging trends in Information Technology (IT) due to its various appealing features and advantages. These have been fuelling its integration in different sectors to increase effectiveness, especially in the medical field. Theoretically, this new concept promises to deliver endless storage systems over the Internet, which are often controlled, allocated and managed by an external third party. In the same line, consumers are generally charged based on the usage of cloud resources for cost savings.

Technique	Advantages	Disadvantages
AES, RSA, DES and 3DES	These algorithms offer the highest level of security and protection. In fact, these cryptosystems are robust to a wide range of attacks.	The utilization of these algorithms in cloud faces two main issues. Normally, these methods are designed to deal with numerical data and textual data. Therefore, these encryption approaches can be considered inappropriate in cases of medical images. Besides, these cryptosystems pose a number of problems around managing keys and searching through encrypted data.
Homomorphic	The central idea behind these methods is that we can process data without having the decryption key. With this encryption method, client data are safe and secure from an untrusted cloud.	While homomorphic has been widely recognized as a promising technique, it also brings new challenges in terms of service performance and availability. In fact, these algorithms generally have high computational costs.
Steganography	By using this technology, we can easily hide secret data inside images. Generally, the cover-media appears to be a meaningful image. Thus, it may help distract the attacker's attention since data appear to be still plaintext.	Despite the fact that steganography techniques are simple and effective ways to protect data, they face many challenges, such as embedding capacity and image quality degradation.

The proposed approach	The use of XOR method presents a strategy to reduce the computational costs required for data encryption. This approach ensures usability without compromising security.	The multi-cloud architecture requires many nodes to store client's data. This approach generally costs more than classical architecture.

Table 4. Comparison between the proposed method and existing techniques

Additionally, this solution is meant to promote and facilitate interoperability and secure exchange of clinical data among healthcare professionals by using shared storage pools. Nevertheless, the implementation of this solution poses security problems due to vulnerabilities and threats associated with different technologies involved in implementing cloud technology. For example, virtualization, web technologies and storage pools are the major source of cloud storage challenges. In this respect, we suggest a framework to protect medical images when using off-site storage pools. The main purpose of this proposal is to outsource data storage to help achieve cost savings, scalability and performance. Particularly, this framework is designed to prevent security breaches in cloud storage, especially insider attacks and risks. To achieve this goal, our proposed architecture relies on multi-cloud environment for preserving privacy of digital data stored on off-site solutions. More precisely, a medical image is split into four pieces before sending it to the cloud storage named CloudSlave. In addition, we introduce CloudSec for enforcing the data protection and avoid the disclosure of personal health information. Since medical images are sensitive data, we use lossless compression algorithms to retain the image quality after the compression, i.e., JPEG 2000. Furthermore, we propose XOR method to encrypt data and SSS method for generating encryption keys. The simulation results demonstrate that our suggested framework provides a secure cloud application to save data. In future work, we plan to analyze the proposed solution in terms of security and robustness using different security parameters, including the Number of Pixels Change Rate (NPCR) and correlation coefficient. Meanwhile, we intend to use watermarking techniques to resolve the authentication issue.

References

[1] P. Mell and T. Grance, "The NIST definition of cloud computing," Technical Report, National Institute of Standards and Technology, vol. 15, 2009.

[2] L. Qian, Z. Luo, Y. Du, and L. Guo, "Cloud computing: an overview," in Proceedings of the 1st International Conference on Cloud Computing, Beijing, China, pp. 626–631, 2009.

[3] E. AbuKhousa, N. Mohamed, and J. Al-Jaroodi, "E-health cloud: opportunities and challenges," Future Internet, vol. 4, no. 4, pp. 621–645, 2012.

[4] B. Calabrese and M. Cannataro, "Cloud computing in healthcare and biomedicine," Scalable Computing: Practice and Experience, vol. 16, no. 1, pp. 1–18, 2015.

[5] M. Marwan, A. Kartit, and H. Ouahmane, "A framework to secure medical image storage in cloud computing environment," Journal of Electronic Commerce in Organizations, vol. 16, no. 1, pp. 1-16, 2018.

[6] M. Marwan, A. Kartit, and H. Ouahmane, "Security in cloud-based medical image processing: requirements and approaches," in Proceedings of the 2nd international Conference on Big Data, Cloud and Applications (BDCA'17), 2017, article no. 6.

[7] A. Mazhar, S.U. Khan, and A.V. Vasilakos, "Security in cloud computing: opportunities and challenges," Information Sciences, Elsevier, vol. 305, pp. 357–383, 2015.

[8] T. Radwan, M.A. Azer, and N. Abdelbaki, "Cloud computing security: challenges and future trends," International Journal of Computer Applications in Technology, Inderscience, vol. 55, no. 2, pp. 158 -172, 2017.

[9] S. Iqbal, M.L.M. Kiah, N.B. Anuar, B. Daghighi, A.W.A. Wahab, and S. Khan, " Service delivery models of cloud computing: security issues and open challenges, " Security and Communication Networks, vol. 9, no. 17, pp. 4726-4750, 2016.

[10] D.A.B. Fernandes, L.F.B. Soares, J.V. Gomes, M.M. Freire, and P.R.M. Inácio, "Security issues in cloud environments: a survey," International Journal of Information Security, vol. 13, no. 2, pp. 113–170, 2013.

[11] N.K. Bajwa, H. Singh, and K.K. De, "Critical success factors in electronic health records (EHR) implementation: an exploratory study in north India," International Journal of Healthcare Information Systems and Informatics, vol. 12, issue 2, pp. 1-17, 2017.

[12] M. Marwan, A. Kartit, and H. Ouahmane, "A cloud based solution for collaborative and secure sharing of medical data, " International Journal of Enterprise Information Systems (IJEIS), vol. 14, no. 3, pp. 128-145, 2018.

[13] F. Shirazi, A. Seddighi, and A. Iqbal, "Cloud computing security and privacy: an empirical study," Lecture Notes in Computer Science, vol. 10272, Springer, Cham: Proceedings of the International Conference on Human-Computer Interaction, pp. 534-549, 2017.

[14] P. Ravi Kumar, P. Herbert Raj, and P. Jelciana, "Exploring security issues and solutions in cloud computing services: a survey," Cybernetics and Information Technologies, vol. 17, no. 4, pp. 3–31, 2017.

[15] B. Yüksel, A. Küpçü, and Ö. Özkasap, "Research issues for privacy and security of electronic health services," Future Generation Computer Systems, vol. 68, pp. 1–13, 2017.

[16] G. Ramachandra, M. Iftikhar, and F.A. Khan, "A comprehensive survey on security in cloud computing," Procedia Computer Science, Elsevier, vol. 110, pp. 465–472, 2017.

[17] A. Shamir, "How to share a secret," Communications of the ACM, vol. 22, no. 11, pp. 612–613, 1979.

[18] M. Marwan, A. Kartit, and H. Ouahmane, "A cloud-based framework to secure medical image processing," Journal of Mobile Multimedia, vol. 14, no. 3, pp. 319-344, 2018.

[19] A. Cheraghi, "Sharing several Secrets based on Lagranges interpolation formula and cipher feedback mode," International Journal of Nonlinear Analysis and Application, vol. 5, no. 2, pp. 60–66, 2014.

[20] Eric Rescorla, "SSL and TLS: designing and building secure systems," Publisher: Addison Wesley, 2001.

[21] M. Marwan, A. Kartit, and H. Ouahmane, "A secure framework for medical image storage based on multi-cloud," in Proceedings of the international Conference on Cloud Computing Technologies and Applications, 2016, pp. 88-94.

[22] M. Tebaa and S. El hajji, "From single to multi-clouds computing privacy and fault tolerance," IERI Procedia, Elsevier, vol. 10, pp. 112-118, 2014.

[23] A. Bessani, M. Correia, B. Quaresma, F. André, and P. Sousa, "DepSky: Dependable and secure storage in a cloud-of-clouds," ACM Transactions on Storage (TOS), vol. 9, no. 4, article 12, 2013.

[24] D.S. Taubman and M.W. Marcellin, "JPEG2000 image compression fundamentals, standards and practice," Lecture Notes in Engineering and Computer Science, Taubman et al., Eds., 2002, ISBN 978-1-4615-0799-4.

[25] S. Barr, Medical image samples, [Online], Available: http://www.barre.nom.fr/medical/samples/. [Accessed 06 May 2020].