JJEE
Volume 5, Number 3, 2019
Pages 175-190
Jordan Journal of Electrical Engineering
ISSN (Print): 2409-9600, ISSN (Online): 2409-9619

A Novel Framework for Storing and Sharing Medical Information Securely Over Cloud Computing Environment

Jamal N. Bani Salameh*

Computer Engineering Department, Mutah University, Alkarak, Jordan
E-mail: jbanisal@mutah.edu.jo

Received: April 22, 2019 *Accepted: May 30, 2019*

***Abstract*—** Medical information is usually stored in a centralized database; its recently fast growth makes it even harder for the traditional local database center to handle over time. Investing in an effective data center locally is not a cost efficient; therefore, high-tech platforms are needed to manage these digital records. Today, cloud computing has become a promising paradigm for medical institutions to store and share their medical information since it offers large storage capacity and huge computing capabilities for a limited cost. On the other hand, it still has potential risks and challenges. However, outsourcing medical information to a public cloud storage provider causes the patient's privacy and medical image security to become a critical concern. In this research, we proposed a novel and secure framework used for storing and sharing medical information over a public cloud based environment. Our approach relies on cryptography and steganography to protect medical information not only during transmission but also when it is stored on the cloud server that is managed by an untrusted party. For cryptography, our encryption algorithm MJEA (for Modified Jamal Encryption Algorithm) has been used; it is a symmetric block encryption algorithm with 64-bit block size and 120-bit key size. This algorithm is used to encrypt the medical image. The encrypted form of the medical image has been used as a cover image to hide the patient's information and the encryption key. As a result, the security techniques used in the proposed framework ensure confidentiality, integrity and authenticity for medical information before outsourcing it to be stored on the cloud server.

***Keywords*—** Medical image; Patient's information; Medical information; Cloud computing; Cryptography; Steganography.

1. INTRODUCTION

Medical information often includes medical images and medical records. Electronic medical record systems may contain medical information such as: Electronic Patient Records (EPRs), Electronic Medical Records (EMR) and Electronic Health Records (EHRs). Those records may include: clinical examinations, diagnosis explanations, physical examination results and other findings. Medical records and medical images are sensitive information; and they are considered to be crucial components in the healthcare sector because they are the key tools for diagnosis and treatment of patients [1]. Usually, medical information is stored locally in centralized database centers. Nowadays healthcare providers generate numerous amounts of medical information daily; and its fast growth makes the job for a traditional data center harder to handle over time. It will not be cost effective to invest in a local database center to solve this problem; therefore, new platforms are needed to manage these huge amounts of digital records [2]. In order to reduce this cost, many medical institutions nowadays decide to store their medical information in an unreliable third party (i.e. cloud server), specially designed for data storage and sharing. Cloud computing provides an efficient solution for healthcare institutions to store, share,

* Corresponding author

access, and view medical information online. Cloud computing is a new paradigm that serves as an alternative to other traditional models. Cloud computing is the most efficient solution that enables consumers to take advantage of information technology without investing and implementing local data centers [3]. Furthermore, the cloud environment provides an on-demand, elastic and scalable pool of services that are accessible from anywhere at any time. As shown in Fig. 1, there are three main service models that are offered through cloud computing:

a) Software-as-a-Service (SaaS): It allows customers to run their applications remotely from the cloud, but they have no control over the underlying cloud infrastructure.
b) Platform-as-a-Service (PaaS): It includes operating systems, storage and required services for a certain application. The user does not control or manage the underlying cloud infrastructure.
c) Infrastructure-as-a-service (IaaS): It refers to computing resources as a service. The user does not control or manage the underlying cloud infrastructure but he has control over storage, operating systems, and applications.

One more service that could be provided by the cloud provider is data-Storage-as-a-Service (dSaaS). The user gets the required storage capabilities as needed, but he does not have control over the underlying cloud infrastructure [4].

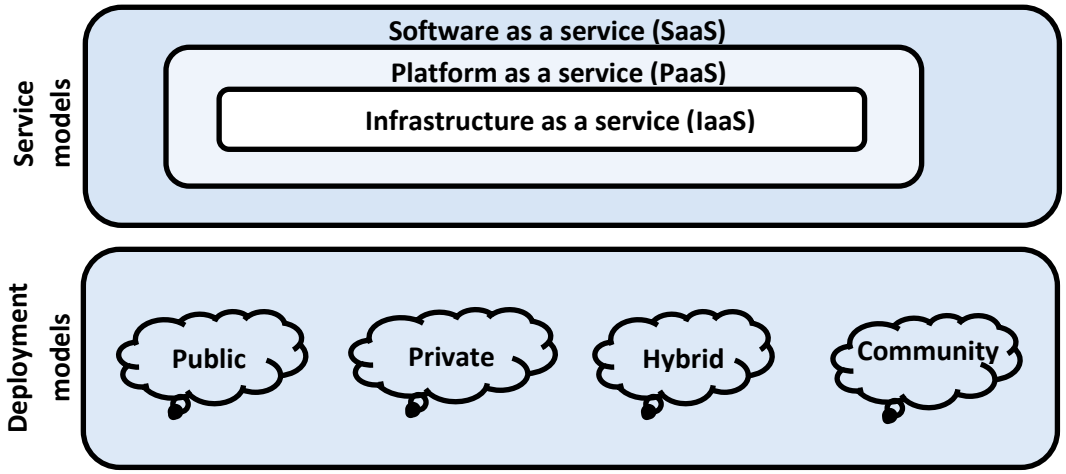

Fig. 1. Service and deployment models for cloud computing.

When the medical data owner (Clinic) considers storing and sharing his electronic health records over the cloud environment, he has the choice of selecting the required model among four common types of cloud deployment models [4]:

a) The private cloud infrastructure: It is operated, managed and controlled solely by the customer; and it may exist on or off premise. In this model, the Cloud Service Provider (CSP) provides the required security and privacy protection.
b) The community cloud infrastructure: It is shared by different organizations; and it supports a particular community with common concerns. It is managed by a third party or the customer; and it may be found on or off premise.
c) The public cloud infrastructure: It is made available to the public or a large business group; and is owned by a CSP. In this model, users have a full responsibility for ensuring patients' privacy and security.

d) The hybrid cloud: It is a combination of at least one private cloud and a public cloud. Although blocks of a hybrid cloud are related, they provide unique hybrid cloud services for several models simultaneously.

However, cloud computing presents a variety of on-demand services and benefits, but it still faces many challenges. In this regard, privacy and security are the main issues that need to be guaranteed before transferring medical data toward the cloud environment. Therefore, when moving medical data over the cloud environment, the risk should be minimized as much as possible as it has a very sensitive and critical data about the patient. A high level of protection has to be maintained to get over any attacking attempts that may interrupt these data when it is exchanged throughout the untrusted network [5]. In order to get secure data transferring and sharing over the cloud based environment, the cloud service provided for users should reach all the security needs such as: availability where service must be available all the time, reliability of the system and network, authentication of customers, data confidentiality and integrity [6].

2. RELATED WORK

Cloud computing offers medical organizations a cost-effective platform for outsourcing medical information, but this approach still faces several issues related to privacy and security. A great deal of the research conducted contributes to enhancing good solutions for security threats in this field. In this section, we will review some related work that handles medical data storing and sharing on the cloud environment. Initially, symmetric key cryptography uses the same key for encryption and decryption processes. The most used symmetric key encryption algorithm with the E-medicine is the advanced encryption standard (AES) [7]. Furthermore, steganography (hiding data inside a cover type of data without changing the original form) has been used for securing medical data. AES was used alongside image steganography to hide an encrypted prescription from the doctor to the pharmacist [8]. Also, the AES was used together with Attribute-Based Encryption (ABE) in E-health. AES has been used to encrypt files and upload them to the hospital system. It helps get advantage from the key policy ABE to give access privileges to customers according to their attributes [9]. Another type of hybrid cryptography is using the RSA along with the AES; the RSA will be used for digital signature whereas the AES is used for encryption. This would maintain data security and integrity through encryption and user authentication through digital signature [10]. Mbarek et al. [11] have proposed a secure platform for medical image storage based on a multi-cloud environment to improve data privacy by allowing healthcare providers and patients to store their medical images over the cloud. They used a secret share scheme to enhance data confidentiality and reversible watermarking technique based on the Thodi algorithm to verify the integrity of medical images. Deshmukh [12] has proposed a technique for storing medical records and accessing them by physicians and patients as authorized by the key control method. The authors of [13] created a common framework on the cloud for hosting electronic health records and services of an entire country. Ahmed et al. [14] suggested providing cloud services for maintaining Electronic Patient Records. Pirretti et al. [15]

have proposed a scheme to periodically re-cipher data by the owner using ABE algorithm. In addition, the data owner has to redistribute the new key to the authorized users each time he re-encrypts his data. The work in [16] proposed the idea of Secure Electronic Medical Record Sharing Mechanism in Cloud Computing Platform. Yiwen et al. [17] introduces an active and secure medical information service framework based on a distributed cloud and block chain technique. Within their proposed framework, medical information is stored on the distributed cloud after encryption. The authors of [18] proposed a novel framework to enhance the protection of DICOM file that contains medical images and privacy of patient information; they encrypt the image and upload it with patient information to the cloud. The cloud will store this information inside an oracle database and the encrypted image inside a file. The keys of this encrypted image will be saved in a database. If the client (doctor) registered in a cloud requests to download any medical image, the cloud will perform the steganography method using least significant bit (LSB) in order to embed the patient's information and the key inside the encrypted medical image. The cloud sends the encrypted image with steganography information to the doctor. The authors of [19] presented two security methods to ensure secure sharing of medical images over the cloud -based environment by providing the mean of trust management between the authorized entities of these data. Besides, it allows sharing the Electronic Patients' Records between those parties. The first technique applies spatial watermarking while the second method implements hybrid techniques. Pan et al. [20] proposed a secured public cloud platform dedicated to medical image sharing by adopting a security policy so as to control different security mechanisms. This policy is based on a risk assessment they conducted so as to identify security objectives with a special interest for digital content protection. These objectives are gathered of different security mechanisms like usage and access control policy, watermarking and partial-encryption. As shown above, many proposals were introduced in the literature that deals with storing, exchanging and sharing medical information in such a way that guarantees security and verifies data confidentiality, integrity and availability. Cloud technology simplifies storage and access to the patients' medical information. That is the reason why cloud storage has been recently adopted by healthcare institutions. Despite its numerous advantages, the shift to cloud storage faces many challenges. In this regard, privacy and security are the main issues that need to be solved before outsourcing medical information to a public cloud storage provider. There are several options available to avoid security issues. One option is to use a private cloud instead of a public cloud. Even the user can fully control this model. Although it is more secure than the public cloud, users are responsible for securing their sensitive data; this can be done by using a suitable encryption technique. Another option is to use the public cloud model. The public cloud model provides the customer with all cloud privileges in terms of shared resources and services, but it is managed and controlled by the provider. In this case, the user is not confident with this kind of service since he does not trust the provider, which makes him worry about his data being revealed and abused. The big challenge in this case is how to create trust between the provider and the owner of the data. In this paper, we proposed a secure framework for storing and

sharing medical information over a public cloud platform. Our approach relies on a hybrid system that combines cryptography and steganography methods to secure medical information not only during transmission but also when it is stored in a cloud server. Our proposed system will include many techniques such as: end-to-end encryption, authentication, authorization and access control. The rest of the paper is organized as follows: Section 3 shows an overview about the proposed framework. Section 4 gives a detailed description of the proposed framework. Section 5 shows some results and discussion about the efficiency of the proposed system. Finally, section 6 concludes this work and suggests some remarks for a future work.

3. OVERVIEW OF THE PROPOSED FRAMEWORK

In this research, we proposed a novel system to provide the required security for storing and sharing Medical Images (MI) and Patient's Information (PI) over a public cloud based environment. More specifically, we are focusing on protecting the content of the exchanged medical information while transmission and during storage on the cloud server. Our approach applied a hybrid system composed of cryptography and steganography; those security techniques will be enough to ensure confidentiality, integrity and authenticity for medical information before outsourcing it to the public CSP. In this section, we will give a full description of the proposed framework.

3.1. Entities Involved in the Proposed Framework

Fig. 2 shows a block diagram for the proposed system; as we see in this figure, the proposed framework permits sharing of medical information between the following three entities:

3.1.1. Medical Data Owner (MDO)

MDO is the owner or producer of the medical data. This could be: a clinic, a radiology center or a hospital where a medical image for a certain patient is taken for the first time. In addition, the required information about the patient and about his case is gathered and stored in an EMR. MDO will be responsible for:

a) Protecting the medical data for his patients before being outsourced to the public CSP.
b) Uploading the encrypted form of the medical information to the CSP to be stored on the cloud server (CS).
c) MDO has a full access to the CS; he can upload, download, add, modify or delete medical records.
d) Assigning a unique encryption key for every medical case.
e) Assigning a unique ID for every remote doctor.
f) MDO has the authority to determine who is allowed to access the cloud server and retrieve medical information for a certain patient.

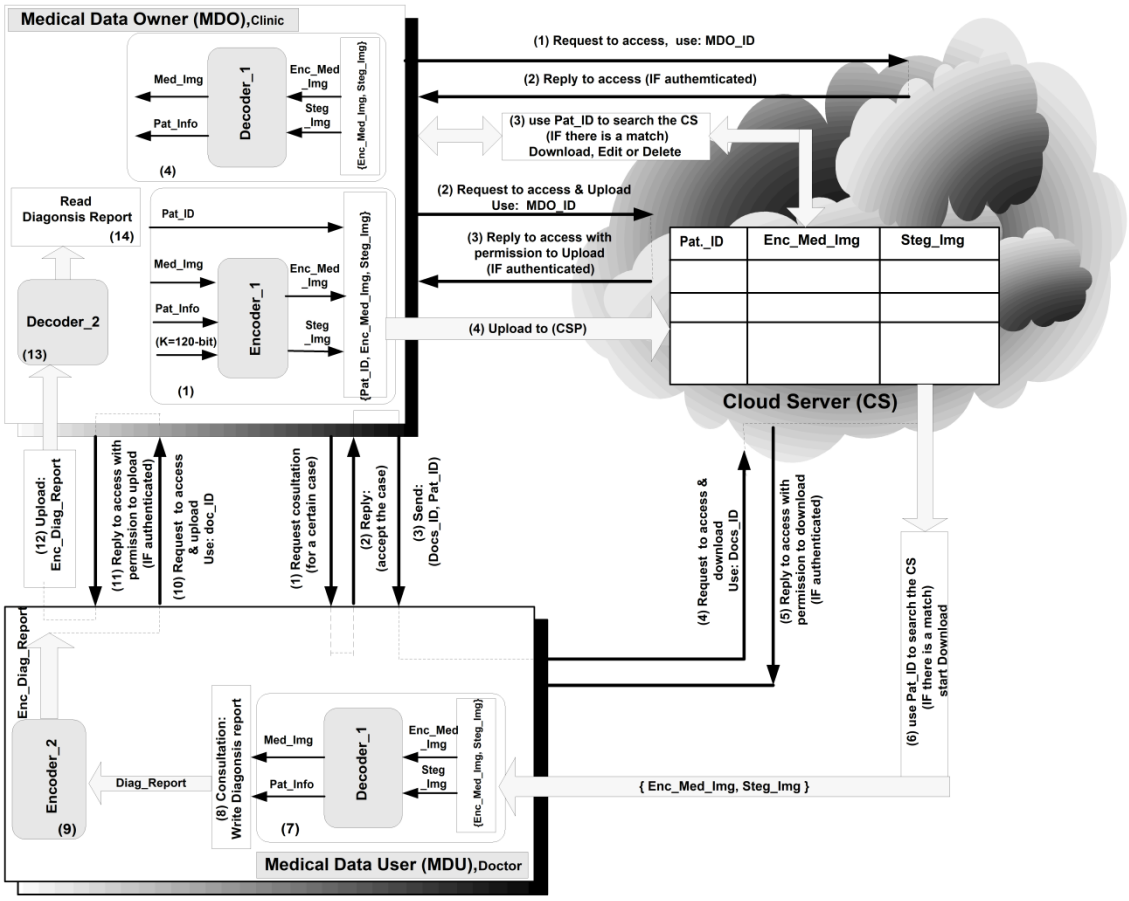

Fig. 2. Block diagram of the proposed framework.

3.1.2. Medical Data User (MDU)

MDU is the one who is authorized to download from CSP, retrieve and use the medical record for a certain patient. MDU could be anyone of the following entities:

a) The remote doctor: He is considered as an outsource-doctor for the clinic. He has been used for consultation and other diagnosis purposes. Each remote doctor has to be registered with the MDO in the doctor's database. Therefore, MDO assigns him a unique ID used for authentication purposes; and he gives him the authority to access the cloud server to download and use medical information for a certain patient.

b) The clinic (MDO): The owner of the medical data is allowed to access the cloud server by using his unique ID with CSP to download and use the medical information for a certain patient.

3.1.3. CSP

CSP is the provider for the public cloud computing environment who offers various services for clients (i.e. storing medical information on the CS and sharing it between MDO and MDU).

3.2. Medical Authentication and Login-IDs Used in the Proposed Framework

The proposed framework creates three types of identifiers to be used for authentication purposes:

a) MDO_ID: the public CSP assigns a unique identifier (MDO_ID) for every registered MDO that will be used for authentication purposes.
b) Docs_ID: the CSP assigns a unique identifier (Docs_ID) and sends it to the MDO. This ID will be used by the MDO to control the access for the CS. It could be sent to any registered doctor with the MDO. Therefore, the doctor can use it with the CSP for authentication purposes to access the CS and download a medical record for a certain patient.
c) Pat_ID: the MDO assigns a unique identifier (Pat_ID) for every patient entering the clinic for medication. This ID could be used in the following scenarios:
   - It is used by the MDO as an identifier for uploading a medical information record of a certain patient to the CSP.
   - It is used by the MDU or MDO as an identifier for downloading a medical information record of a certain patient from the CSP.

3.3. The Proposed Framework Scenarios

As shown in Fig. 2, the proposed system has the following scenarios:

3.3.1. Uploading Scenario:

This scenario is responsible for uploading medical records to the CS. The parties involved in this scenario are the MDO and the CSP. MDO is concerned about uploading a secured form of medical data to the CSP. This process is accomplished within 4-steps as shown in Fig. 2 and described below:

- Step (1): Encoder_1 is responsible for protecting or encoding medical information before sending it to the CSP; the output of this step will be a bundle that includes (Pat_ID, Enc_Med_Img, Steg_Img) to be uploaded to the CSP.
- Step (2): MDO sends a request using SSL connection to the public CSP which asks for permission to access the account in order to upload medical data. His ID along with the request for authentication purposes will be sent.
- Step (3): If MDO is registered with CSP and his ID is valid, the CSP will send a reply for the MDO by which he is permitted to access his account to upload his medical data.
- Step (4): MDO starts uploading the secured data that he got out of step (1) to the CSP.

3.3.2. Consultation Scenario:

The parties that are involved in this scenario are the MDO, MDU and the CSP. This scenario is concerned about getting a consultation from a remote doctor for a medical case of a certain patient; this process is accomplished within 14-steps as shown in Fig. 2 and described below:

- Step (1): The MDO sends a request to the remote doctor asking him for a consultation about a medical case of a certain patient.
- Step (2): If there is no objection, the remote doctor will notify the MDO that his request has been accepted.

- Step (3): The MDO will send the remote doctor two ID's (Pat_ID and Docs_ID) to be used once he is connecting with the CSP.

The remote doctor needs to see the medical image and the patient's information in order to do his consultation. To get that, he will do the following:

- Step (4): the remote doctor will send a request to the CSP through which he asks to access the cloud server to download the required information; he will send the Docs_ID for authentication purposes.
- Step (5): If this ID is valid, that the doctor is authenticated; and the CSP will send a reply - for the doctor - by which he is permitted to access the cloud server and download medical data.
- Step (6): The doctor will use the Pat_ID to search the database of the cloud sever; if there is a match then he will start downloading that record which contains all the required medical information about the patient.

The information retrieved from the CSP is in a ciphered form. It is not readable unless it is decoded. To get that, the doctor needs to do the following:

- Step (7): Decoder_1 will be used for decoding medical information and return it back to the original form. Two outputs have been generated out of this step: the medical image and the patient's information that were originated at MDO side.
- Step (8): Once the doctor viewed the medical image and the patient's information, he will be able to diagnose the case and write his diagnosis report (Diag_Report).
- Step (9): The proposed system requires that medical information should be encrypted before being exchanged between communication parties; for this reason, the remote doctor needs to use Encoder_2 to encrypt his consultation before transmitting it to MDO. The output for this step will be (Enc_Diag_Report).

Now, the remote doctor needs to access his account with the MDO to upload his encrypted diagnosis report; in order to accomplish this, he needs to do the following:

- Step (10): the remote doctor will send a request to the MDO by which he is asked to access his account to upload his report; he will send his (doc_ID) for authentication purposes.
- Step (11): If this ID is valid, the doctor is authenticated; and the MDO will send a reply for the doctor by whom he is permitted to access his account and upload the report.
- Step (12): The remote doctor will upload a bundle which includes {Pat_ID, Enc_Diag_Report}.
- Step (13): The MDO can easily decrypt the consultation (Enc_Diag_Report) by using Decoder_2 to get the original Diag_Report.
- Step (14): Finally the MDO can read the report carefully and prescribe the suitable medication for the patient according to the specialist doctor's consultation.

3.3.3. Downloading Scenario:

This scenario is responsible for downloading medical records from the CSP. Furthermore, the proposed framework gives MDO the permission to download, edit, or even delete existing medical records stored on the cloud server. This process is accomplished within 4-steps as shown in Fig. 2 and described below:

- Step (1): The MDO will send a request to the CSP by which he is asked to access his account and send the (MDO_ID) for authentication purposes.
- Step (2): If this ID is valid, he is registered and authenticated. The CSP will send a reply by which he is permitted to access the cloud server.

Since the MDO has a full access to his account, then he can download, edit or delete any record from the cloud server.

- Step (3): If the choice for the MDO is to download the medical information of a certain patient, he will use the Pat_ID to search the database of the cloud sever; if there is a match then he will start downloading that record which contains all the required medical information about the patient.

The information retrieved from the CSP is in a ciphered form; and it is not readable unless it is decoded. To get that, the MDO needs to do the following:

- Step (4): Decoder_2 will be used for decoding medical information and return it back to the original form. Two outputs have been generated out of this step: the medical image and the patient's information that were originated at MDO before.

4. DETAILED DESCRIPTION OF THE PROPOSED FRAMEWORK

As shown in Fig. 2, the block diagram of the proposed framework has many building blocks. In this section, we will describe those blocks in details:

4.1. Detailed Description of Encoder_1

This encoder is used by the MDO; and it is responsible for encrypting and using the encrypted image as a cover image to hide the patient information and the encryption key before uploading them to the CSP. Fig. 3 shows the block diagram for this encoder.

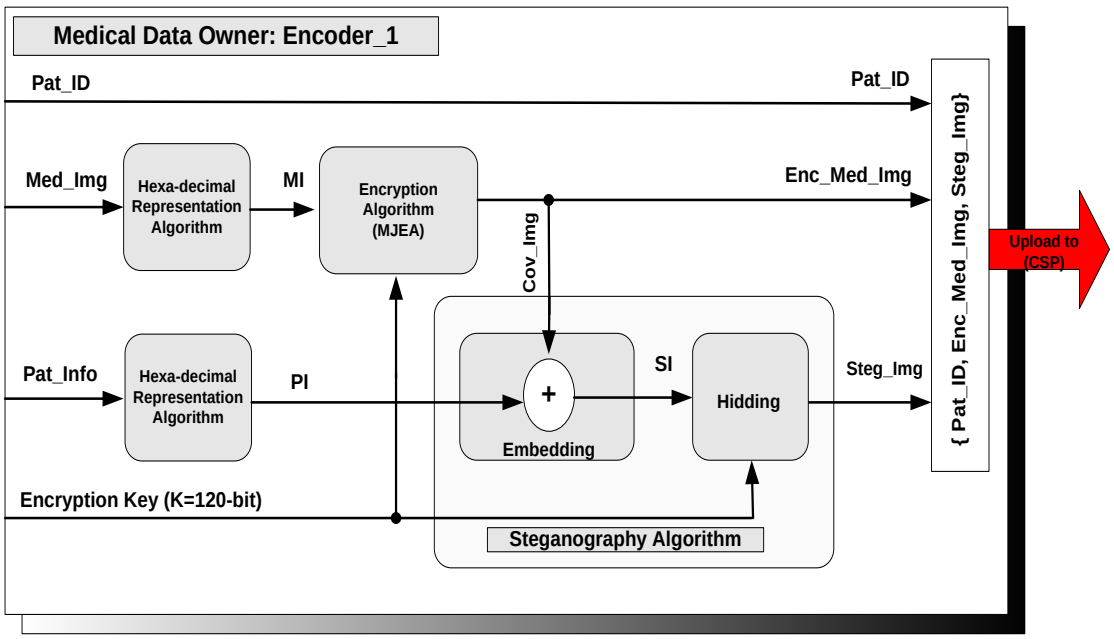

Fig. 3. Architecture of Encoder_1.

The figure shows that this encoder has four inputs {Pat_ID, Med_Img, Pat_Info, K (120-bit)}, and three outputs {Pat_ID, Enc_Med_Img, Steg_Img}. This encoder works as follows:

Firstly, the hexadecimal representation algorithm will be used to convert the (Med_Img) and (Pat_Info) into 8-bit binary representation and store the result in arrays called (MI) and (PI); each one of those arrays has 8-columns and S/8 rows, where S represents the number of pixels for the medical image. After that, the encryption algorithm (MJEA) will be used by the help of K (120-bit) to encrypt (MI) and store the result in an array called (Enc_Med_Img) that represents the encrypted form of the medical image, which will be used as a cover image for the steganography algorithm. The steganography algorithm is simple; and has two steps: The first step will mix the two input arrays together; this can be done by doing bit-by-bit XORing between (MI) and (PI) and storing the result in an array called (SI) that represents the steganography image. The second step will embed or hide the encryption key (K) in the last 120-bit of SI; and the output of this step will be stored in an array called (Steg_Img). The location where to hide the encryption key is only known by the MDO and the MDU as a shared secret. We believe this is a good method to hide the encryption key since the (Enc_Med_Img) that will be used as a cover image is a set of pseudorandom bits. Hiding the pseudorandom bits representing the encryption key will not be traceable by the attacker. The bundle uploaded to the CSP will be as follows: {Pat_ID, Enc_Med_Img, Steg_Img}. As a final note for this encoder, there was no capacity limitation to the patient's information that will be embedded in the encrypted medical image; the only limitation is the size of the patient's sensitive information which should not exceed the size of the image.

4.2. Brief Description of MJEA

In this research, we adopted our MEJA algorithm to fulfill the required medical information's encryption and decryption in the proposed framework. MJEA is a novel symmetric-key block encryption algorithm; it has a 64-bit block size, 8-rounds, and 120-bit key. MJEA has been analyzed thoroughly as both a plain text encryption algorithm and an image encryption algorithm. MJEA divides the plain text message or the image into 64-bit blocks. Then it encrypts each block separately. In MJEA, all operations are Xored on 8-bit words. Each 64-bit block of the plain text (Pt) goes to one end of the algorithm; and then it runs into 8-rounds under the control of the encryption key to produce the 64-bit block of the ciphered text (Ct). The decryption process in MJEA is the same as the encryption, but it flaws in the reverse direction; the S-boxes' linear transformation of the algorithm and the order of the sub-keys must be used in a reverse order. The decryption process of MJEA starts with Ct as an input; and ends with Pt as an output. Ct is divided into 64-bit blocks; then each block is decrypted separately. Every Ct-block goes to one end of MJEA; then the algorithm runs in 8-rounds under the control of the same 120-bit encryption key that has been used at the sender side to produce the Pt-block at the end. The design's algorithm is easy; and its performance is good according to the Avalanche Effect [21, 22].

4.3. Detailed Description of Decoder_1

As we said before, the entities authorized to download medical information from the CSP are the MDU and the MDO; and the downloaded information is in a ciphered form that is not readable unless it is decoded. Decoder_1 could be used by MDU or by MDO to do this job and return the ciphered medical information back to the original form. Fig. 4 shows a block diagram of this decoder which has two inputs {Enc_Med_Img, Steg_Img} and three outputs {Med_Img, Pat_Info, K (120-bit)}. This decoder works as follows: The first step will rearrange the received Enc_Med_Img and Steg_Img into arrays with 8-columns and S/8 rows, where S represents the number of pixels in the medical image. Furthermore, the steganography algorithm will be used in the reverse direction to extract the encryption key (K) from the last 120-bits of the Steg_Img; and the remaining image will be the SI. After that, the digital form of the patient's information PI will be extracted by doing bit-by-bit XORing between SI and the Cov_Img. Moreover, the digital form of the medical image MI will be formed by using MJEA and the extracted-K to decrypt the Enc_Med_Img. Finally, the hexadecimal representation algorithm will be used in the reverse direction to convert the arrays MI and PI by the help of Matlab functions to retrieve the original medical image (Med_Img) and the patient's information (Pat_Info). Now, when the consultant (outsourcing doctor) views the medical information (Med_Img and Pat_Info) of a certain patient, he will analyze it carefully and send back his diagnosis about the case to the clinic (MDO). The proposed system requires that any transmission between any communication entities should be encrypted, so the doctor needs to use Encoder_2 that is shown below to encrypt his report before sending it back to the MDO.

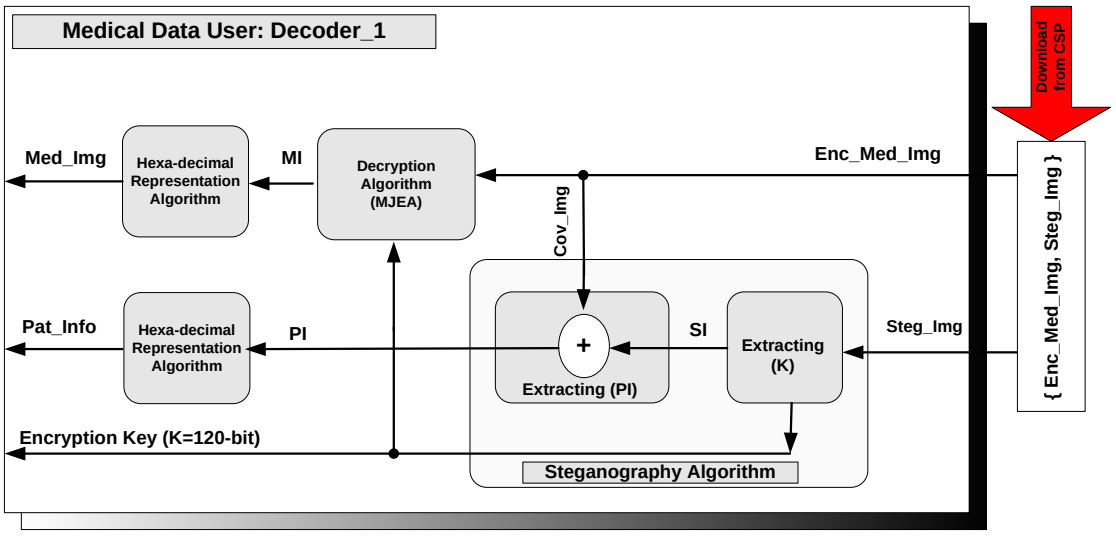

Fig. 4. Architecture of Decoder_1.

4.4. Detailed Description of Encoder_2

This encoder is used by MDU; and it is responsible for encrypting the consultant action report before sending it to the MDO because it will be sent over an unsecured channel. Fig. 5 shows a block diagram of this encoder. The figure shows that this encoder has three inputs: {Pat_ID, Diagnosis Report, K (120-bit)} and two outputs

{Pat_ID, Enc_Diag_Report}; this encoder works as follows: Firstly, the hexadecimal representation algorithm will be used to convert the Diagnosis Report into 8-bit binary representation and store the result in an array called (Diag_Report). After that, the encryption algorithm (MJEA) with the help of K (120-bit) will be used to encrypt (Diag_Report) and store the result in an array called (Enc_Diag_Report) that represents the encrypted form of the Diagnosis Report. The bundle that will be sent securely to the MDO will contain {Pat_ID and Enc_Diag_Report}.

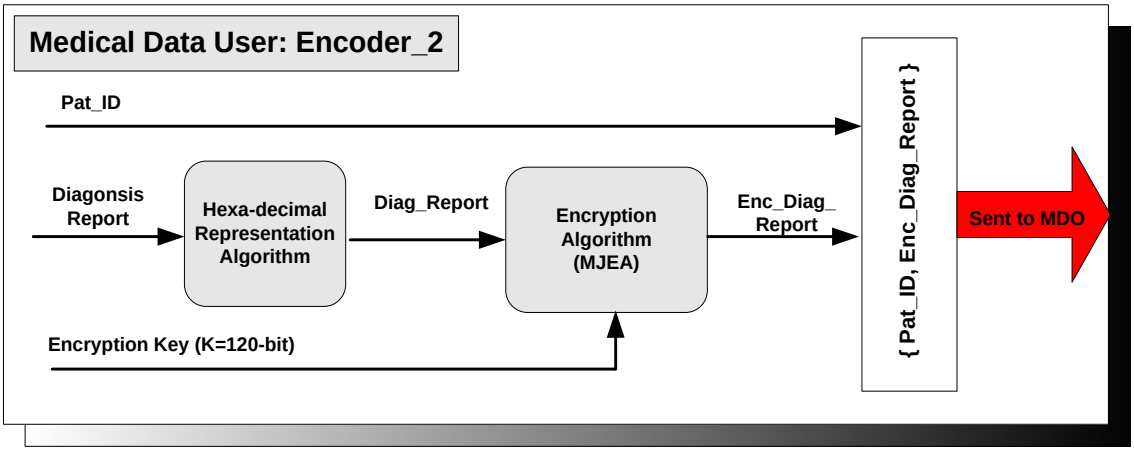

Fig. 5. Architecture of Encoder_2.

4.5. Detailed Description of Decoder_2

This decoder is used by MDO; and it is responsible for decrypting the Enc_Diag_Report that comes from the remote doctor (consultant). Fig. 6 shows an architecture diagram of this decoder. The figure shows that the bundle that is received as an input for this decoder contains: Pat_ID that will be used to identify the patient and get his own encryption key from the MDO's db. Simply this decoder MJEA with the help of K (120-bit) is used to decrypt the Enc_Diag_Report to get Diag_Report. Finally, the (hexa-decimal representation algorithm) will be used in the reverse direction to convert this file to a readable report.

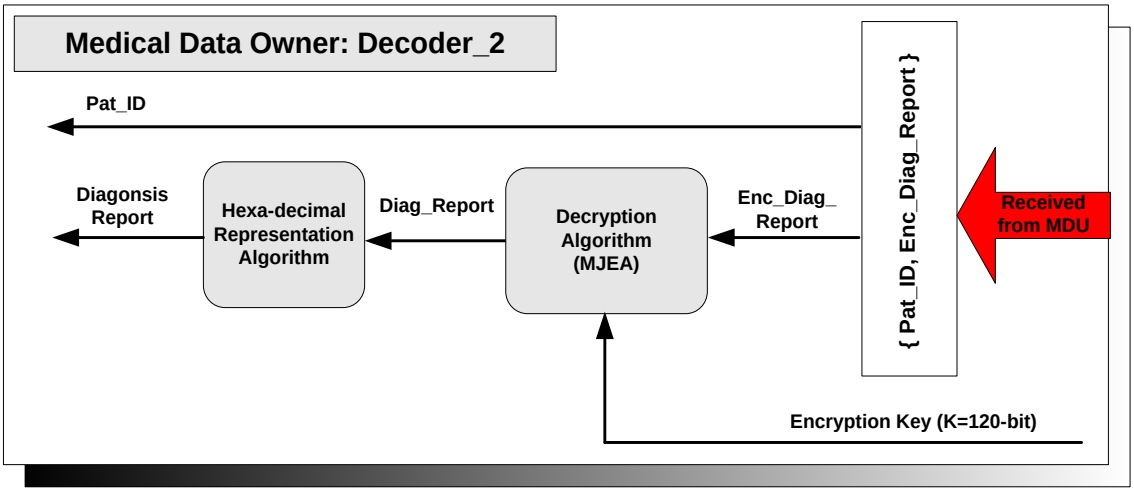

Fig. 6. Architecture of Decoder_2.

5. RESULTS AND ANALYSIS

We applied the proposed technique at the MDO side to protect some selected medical images (Iris and Chest) before outsourcing them to the cloud environment. Both Figs. 7 and 8 show a sample of the results of this experiment: Figs. 7(a) and 8(a) show original medical images (Org_Img) that should be encrypted before uploading them to the CSP. Figs. 7(b) and 8(b) show the medical images after they were encrypted by using MJEA and the encryption key Enc_Img. Figs. 7(c) and 8(c) show the Steg_Im g that hides the patient's information and the encryption key. Furthermore, Figs. 7(b, c) and 8(b, c) form the bundle that will be sent securely to the cloud service provider; there is no risk sending it over an unsecured-channel because it is protected against man-in-the-middle attack. Once the CSP has received those images, he will store them in the cloud server. So, the encrypted form of the medical information will be stored on the cloud sever. The cloud provider has no control over encryption and decryption for the stored medical information. Therefore, if the cloud provider or any attacker is able to get access to the stored medical information, he will get nothing; the retrieved data will be meaningless for them since they are encrypted. He needs to know how to decrypt those data in order to use them accordingly. However, we believe that the proposed framework was able to minimize the risk of any attacking attempts that may face medical information when transferring it over the cloud environment. Furthermore, it achieves all security requirements (i.e. authentication of users, data confidentiality and integrity) that are needed for storing and sharing medical information over cloud environments with other clients.

The proposed framework relies on MJEA to provide the required protection for medical information; we believe it is a secured algorithm because it has been analyzed considerably as a text and as an image encryption algorithm in former research. All experimental results showed the strength of MJEA to encrypt plain text and digital images. The performance evaluation of the algorithm achieves good security properties: it thoroughly scrambles the plain text with the 120-bit key when running for 8-rounds. It achieves a good Avalanche Effect as quickly as possible and completes the encryption/decryption process with a high speed [21]. MJEA was able to achieve a minimum correlation coefficient, high quality and embedding capacity of an encoded image. In addition, it was able to replace and transform all pixels in the original image. The histogram uniformity in the results ensures the success of MJEA in achieving the required randomness. On the receiver side, there was no loss of image quality after performing the decryption process [22]. Furthermore, in a previous work, we designed a new approach for securing medical images and patient's information by using a hybrid system adopting MJEA. We used different simulation metrics for evaluating the performance of this work such as: Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR), and histogram distribution analysis. All experimental results proved the strength of the proposed system [23].

As a final note, the time needed for encoders and decoders to perform their operations actually depends mainly on MJEA. The design philosophy behind MJEA was simplicity of the design which yields an algorithm that is easier to implement and achieve a good Avalanche Effect as quickly as possible. Further evaluation of the

proposed framework will be adopted in the future work to measure the time needed for encoders and decoders to perform their operations.

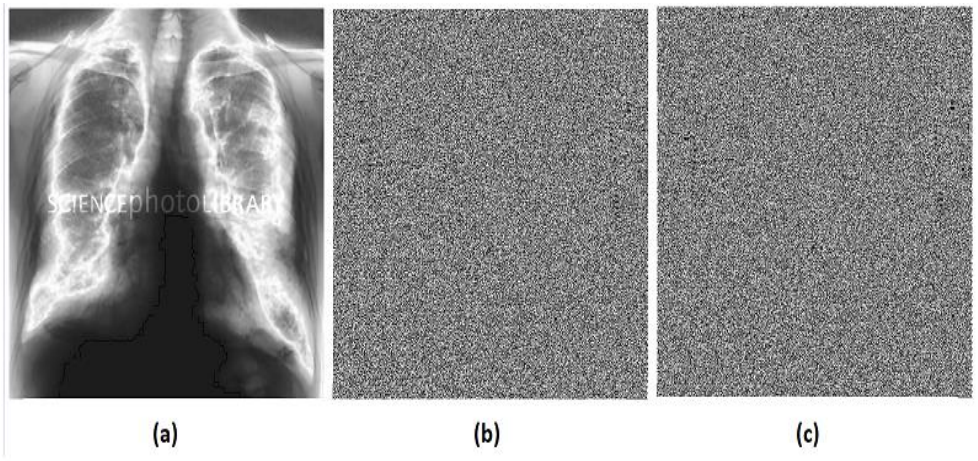

Fig. 7. Protecting chest-image at the MDO side.

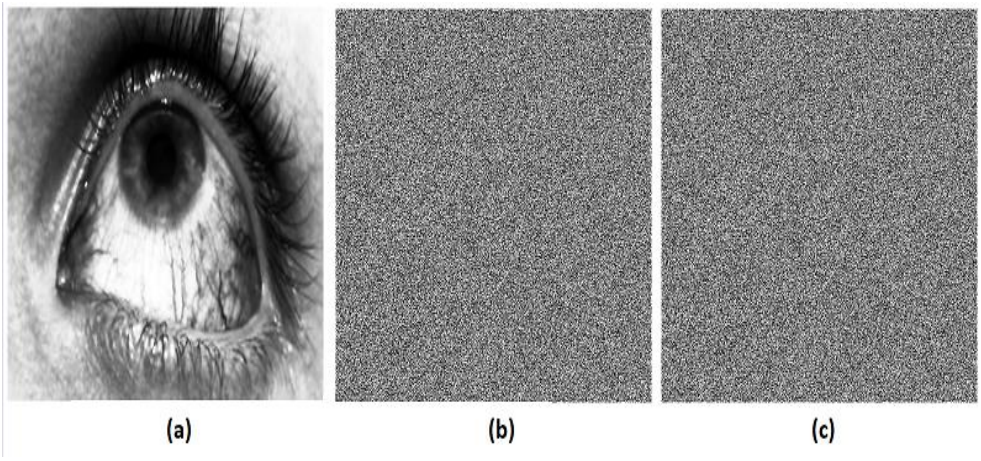

Fig. 8. Protecting iris-image at the MDO side.

6. CONCLUSIONS AND FUTURE WORK

In this research, we proposed a novel framework to provide the required security for storing and sharing (MI) and (PI) over the cloud-based environment. Our proposed technique relies on cryptography and steganography to protect the medical data while uploading them to the CSP over an unsecured-channel. For cryptography, we used our encryption algorithm (MJEA) to encrypt the (MI). (Enc_Med_Img) is used as a cover image to hide the patient's information before uploading it to the CSP. In order to hide (PI) into the cover image, a simple technique representing the steganography algorithm has been used. It does bit-by-bit Xoring between (PI) and (Enc_Med_Img) in order to get the SI. Furthermore, the second step in this process will hide the encryption key (K) in the last 120-bit of SI to get (Steg_Img). The location where to hide the encryption key is only known by the MDO and the MDU as a shared secret. We believe this is a good method to hide the encryption key since using (Enc_Med_Img) as a cover image is a set of pseudorandom bits; therefore to hide the pseudorandom bits representing the encryption key will not be traceable by the attacker. The bundle that will be uploaded

to the CSP will contain the following: {Pat_ID, Enc_Med_Img, Steg_Img}. The proposed framework has good access control options in which MDO creates four types of Login-IDs that are used for authentication purposes. This process improves the patient's privacy and the security of sensitive information (i.e. medical data). The proposed framework achieves two main types of authentication in this cloud environment framework: the first one is authentication between the MDO and the MDU by using a unique ID for each remote doctor; and the second one is between the CSP and MDO by using a unique ID for the medical data owner. As a result, the proposed framework protects the privacy of patients and ensures confidentiality, integrity and authenticity for medical information before outsourcing it to be stored on the cloud server. In our future work, we will implement our framework by using Java and MySQL; and it will be tested in a real cloud environment. Additionally, we will do more performance evaluation experiments under different metrics such as: up-time and down-time ratio, system throughput, and response time; then we will conduct more analysis for the proposed framework to get better performance.

REFERENCES

[1] B. Cyganek, M. Graña, B. Krawczyk, A. Kasprzak, P. Porwik, K. Walkowiak, M. Wozniak, "A Survey of big data issues in electronic health record analysis," *Applied Artificial Intelligence*, vol. 30, no. 6, pp. 497-520, 2016.

[2] P. Sanjay, M. Sindhu, J. Zambrano, "A Survey of the state of cloud computing in healthcare," *Network and Communication Technologies*, vol. 1, no. 2, pp. 12-21, 2012.

[3] P. Melland, T. Grance, "The NIST definition of cloud computing," *National Institute of Standards and Technology*, vol. 15, pp. 1-3, 2009.

[4] R. Buyya, J. Broberg, A. Goscinski, *Cloud Computing: Principles and Paradigms*, John Wiley & Sons, 2011.

[5] R. Asija, R. Nallusamy, "A Survey on security and privacy of healthcare data," *Proceedings of the 3rd Annual global Healthscare Conference*, Singapore, 2014.

[6] R. Sumathi, E. Kirubakara, "SCEHSS: secured cloud based electronic health record storage system with re-encryption at cloud service provider," *International Journal of Computer and Communication Engineering*, vol. 2, no. 2, pp. 162-171, 2013.

[7] A. Jammu, H. Singh, , "Improved AES for data security in e-health," *International Journal of Advanced Research in Computer Science,* vol. 8, no. 5, pp. 2016-2020, 2017.

[8] A. Omotosho , O. Adegbola, O. Mikail, J. Emuoyibofarhe, "A Secure electronic prescription system using steganography with encryption key implementation," *International Journal of Computer and Information Technology*, vol. 3, no. 5, pp. 980-986, 2015.

[9] R. Kalaiselvi, K. Kousalya, R. Varshaa, M. Suganya, "Enhanced secure sharing of personal health records in cloud computing," *Gazi University Journal of Science*, vol. 29, no. 3, pp. 583-591, 2016.

[10] M. Sadikin, R. Wardhani, "Implementation of RSA 2048-bit and AES 256-bit with digital signature for secure electronic health record application," *Communication & Information Technology Journal*, vol. 10, no. 2, pp. 63–69, 2016.

[11] M. Mbarek, A. Kartit, H. Ouahma, "A Secure framework for medical image storage based on multi-cloud," *Proceedings of the 2nd IEEE International Conference on Cloud Computing Technologies and Applications*, 2016.

[12] P. Deshmukh, "Design of cloud security in the EHR for Indian healthcare services," *Journal of King Saud University Computer and Information Sciences*, vol. 29, pp. 281-287, 2017.

[13] E. Hendrick, B. Schooley, C. Gao, "Cloud health: developing a reliable cloud platform for healthcare applications," P*roceedings of the IEEE Consumer Communications and Networking Conference*, pp. 887-891, 2013.

[14] S. Ahmed, A. Abdullah, "E-healthcare and data management services in a cloud," *Proceedings of High Capacity Optical Networks and Enabling Technologies*, pp. 248-252, 2011.

[15] M. Pirretti, P. Traynor, P. McDaniel, B. Waters, "Secure attribute-based systems," *Journal of Computer Security*, vol. 18, no. 5, pp. 799–837, 2010.

[16] Z. Li, E. Chang, K. Huang, F. Lai, "A Secure electronic medical record sharing mechanism in the cloud computing platform," *Proceedings of the IEEE 15th International Symposium on Consumer Electronics*, 2011.

[17] D. Yiwen, L. Jianwei, G. Zhenyu, F. Hanwen, "A Medical information service platform based on distributed cloud and blockchain," *Proceedings of 2018 IEEE International Conference on Smart Cloud*, 2018.

[18] S. Kurnaz, A. Jasim, "Cloud system for encryption and authentication medical images," *Journal of Computer Engineering*, vol. 20, no. 1, pp. 65-75, 2018.

[19] F. Elgamal, N. Hikal, F. Abu-Chadi, "Secure medical images sharing over cloud computing environment," *International Journal of Advanced Computer Science and Applications*, vol. 4, no. 5, pp. 130-138, 2013.

[20] W. Pan, G. Coatrieux, D. Bouslimi, N. Prigent, "Secure public cloud platform for medical images sharing," *European Federation for Medical Informatics*, pp. 251-255, 2015.

[21] J. Salameh, "A New symmetric-key block ciphering algorithm," *Middle-East Journal of Scientific Research*, vol. 12, no. 5, pp. 662-673, 2012.

[22] J. Bani Salameh, "An investigation of the use of MJEA in image encryption," *WSEAS Transactions on Computers*, vol. 15, pp. 12-23, 2016.

[23] J. Bani Salameh, "A new approach for securing medical images and patient's information by using a hybrid system," *International Journal of Computer Science and Network Security*, vol. 19, no. 4, pp. 28-39, 2019.