IJCSE International Journal of Computer Sciences and Engineering Open Access
Survey Paper Vol.-7, Issue-4, April 2019 E-ISSN: 2347-2693

# Survey Paper on Quaternion-Based Encryption

**Garima Mathur[1*] , Anjana Pandey[2]**

[1]Department of Computer Science, UIT, RGPV, Bhopal, India
[2]Department of Information Technology, UIT, RGPV, Bhopal, India

*Corresponding Author: garima41mathur@gmail.com, Tel. : 8871168663*

DOI: https://doi.org/10.26438/ijcse/v7i4.978990 | Available online at: www.ijcseonline.org

Accepted: 18/Apr/2019, Published: 30/Apr/2019

***Abstract***— Nowadays, delivering sensitive digital multimedia contents confidentially over vulnerable public networks is a matter of high importance and encryption is a technique which is widely used for secure communication. An improved and extended version of a lossless encryption technique used for digital images is quaternion based encryption. The quaternion based encryption scheme, significantly improves speed of images encryption in comparison with those originally embedded advanced encryption standard (AES) and triple data encryption standard (3-DES) algorithms. It utilizes extraordinarily properties of quaternion's to perform rotations of information in 3D space for each of the cipher rounds. In this paper it has been surveyed about existing works on quaternion based encryption technique for both gray toned and color images as well as its application for secure transmission of medical images.

***Keywords*** **-** Quaternion rotation, lossless scheme, security, DICOM, image processing, key sensitivity.

## I. INTRODUCTION

As we know that secure transmission of digital images over internet is of high importance. Cryptography is a technique widely used for secure communication. The problem received much attention, however the number of possible methods which enable the transmitted data interception, is increasing rapidly. Quaternion based encryption is one of them which not only eliminates weak keys from the system but also ensures robustness to brute-force attacks. Quaternion encryption uses the unique properties of quaternion's in order to rotate vectors of data in three-dimensional space [1][5][9]. Quaternion's are hyper-complex numbers of rank 4 and thus often applied to mechanics in three-dimensional space. Because of their unique structure, quaternion's are commonly used in place of matrices to perform rotations. In a proposed method, a data vector rotation is used as its encryption.

Due to specific quaternion algebra, the encryption based on a quaternion rotation will be computed much faster than encryption based on a matrix multiplication [12]. Additionally, when encrypting a color image in RGB representation, it is possible to increase the encryption efficiency even further.

### 1.1. Quaternion Calculus

Quaternions are hyper-complex numbers of rank 4 and have two parts-a scalar part and a vector part, which is an ordinary vector in a three dimensional space R3 [3]. A quaternion $q$ is defined by [2][12] -

$$q = w+xi+yj+zk ,$$

Here $w$, $x$, $y$, $z$ are real coefficients of quaternion $q$, and $i$, $j$, $k$ are imaginary units having following properties

$$i^2 = j^2 = k^2 = ijk = -1,$$
$$ij = -ji = k ,$$
$$jk = -kj = i ,$$
$$ki = -ik = j ,$$

A quaternion could also be written as a transposed vector or as a composition of scalar part w and vector part $v$.

$$q = [w\ x\ y\ z]^T \text{ or } q = (w,v) = (w, [x\ y\ z]^T)$$

The sum of two quaternion $q_1$, $q_2$ is defined by adding the it's corresponding coefficients, i.e. in the same manner as for complex numbers

$$q_1+q_2 = (w_1+w_2) +(x_1+x_2)\ i+ (y_1+y_2)\ j+ (z_1+z_2)\ k$$

The product of two quaternion's is more complex due to its anti-commutative of the imaginary units during the multiplication process. The product of the two quaternion's $q_1$, $q_2$ consist scalar and vector products

$$q_1 .q_2 = (w_1w_2- v_1.v_2,\ w_1.v_2 + w_2.v_1 +v_1 \text{ x } v_2)$$

Furthermore, it is important to define the other properties: A conjugate $q^*$, a norm $||q||$ and an inverse $q^{-1}$ of a quaternion $q$:

$$q^* = w-xi-yj-zk$$

$$||q|| = \sqrt{w^2+x^2+y^2+z^2}$$

$$q^{-1} = \frac{q^*}{||q||} = \frac{w-xi-yj-zk}{w^2+x^2+y^2+z^2}$$

It is essential to see that on account of a unit quaternion, for which the standard is equivalent to 1, there is the following relation: $\mathbf{q^{-1} = q^*}$

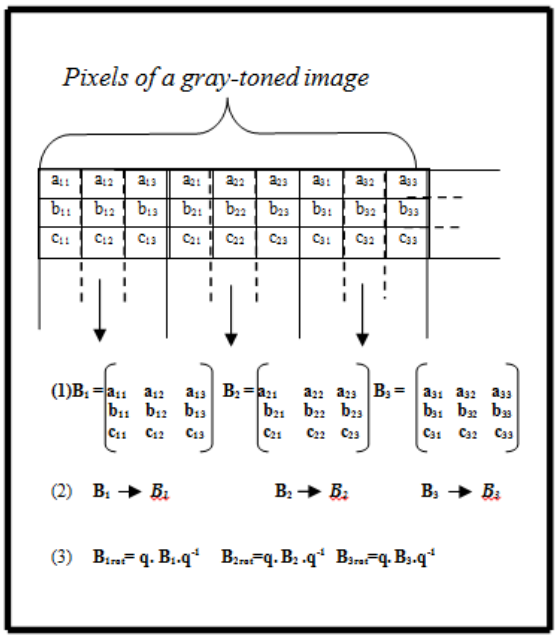



**Fig.1 Encryption of a gray-tone image using the quaternion method**

### 1.2 Quaternion Encryption

The encryption scheme is based on quaternion rotation by creating a quaternion (key) around will be rotating another quaternion (data).

Let's consider two quaternion's $q = [w\ x\ y\ z]^T$ and $P = [0\ a\ b\ c]^T$, where a vector $[a\ b\ c]^T$ ,which represents a vector part of the quaternion $P$, will store data to rotate around a unit quaternion $q$ (key). The obtained quaternion $P_{rot}$ will be a spatial mapping of the pivoted information vector $[a\ b\ c]^T$. The quaternion rotation is written as:

$$\boldsymbol{P_{rot} = q.P.q^{-1}}$$

There are two possible ways of implementing a proposed quaternion encryption. First approach called quaternion method focuses on the Eq. and it is entirely based on quaternion calculus.

The quaternion method is entirely based on the quaternion rotation shown in Eq .

It is possible to optimize the rotation process by extending the vector part of the quaternion $P$ in order to obtain a new quaternion $B$, as it is shown

$$\mathbf{P=}\begin{bmatrix}0, \begin{bmatrix}a\\b\\c\end{bmatrix}\end{bmatrix} \rightarrow \mathbf{B=}\begin{bmatrix}0, \begin{bmatrix}a_1 & a_2 & a_3\\ b_1 & b_2 & b_3\\ c_1 & c_2 & c_3\end{bmatrix}\end{bmatrix}$$

The encryption and decryption process for the quaternion method with the new extended quaternion B (meant to store data information) is shown respectively

$$\boldsymbol{B_{rot} = q\ .B.\ q^{-1}} \textbf{ and } \boldsymbol{B = q^{-1}\ B_{rot}\ .\ q}$$

Where, $B_{rot}$ is the rotated quaternion $B$.

It is also possible to compute a rotation matrix in order to obtain quaternion's of higher order that can be treated as subsequent encryption keys and therefore improve the encrypted data security. The number of computed higher order quaternion-keys is equal to $3^n$, where $n$ denotes the order's size.

### 1.3 A Matrix Representation for Quaternion Multiplication [2]

$q_1 = w_1+x_1i+y_1j+z_1k$

$q_2 = w_2+x_2i+y_2j+z_2k$

$$\begin{aligned} q_{1*}\ q_2 = &(w_1\ w_2 - x_1\ x_2 - y_1\ y_2 - z_1\ z_2)\\ &+ (w_1\ x_2 + x_1\ w_2 + y_1\ z_2 - z_1\ y_2)\ i\\ &+ (w_1\ y_2 - x_1\ z_2 + y_1\ w_2 + z_1\ x_2)\ j\\ &+ (w_1\ z_2 + x_1\ y_2 - y_1\ x_2 + z_1\ w_2)\ k \end{aligned}$$

If we examine each term in this product, we can see that each term depends directly on the coefficients for $\mathbf{q_1}$.Also each term depends straightly on the coefficients for $\mathbf{q_2}$. Along these lines, we can compose the result of two quaternion's in terms of a matrix multiplication.

When the matrix $\mathbf{L_{row}\ (q_1)}$ multiplies a row vector $\mathbf{q_2}$, the result is a row vector representation for $\mathbf{q_1 * q_2}$ .

When the matrix $\mathbf{R_{row}\ (q_2)}$ multiplies a row vector $\mathbf{q_1}$, the result is also a row vector representation for $\mathbf{q_1 * q_2}$.

$$\mathbf{q_1 * q_2 = q_2 L_{row}(q_1) = [\ x_2\ y_2\ z_2\ w_2]} \begin{bmatrix} w_1 & z_1 & -y_1 & -x_1\\ -z_1 & w_1 & x_1 & -y_1\\ y_1 & -x_1 & w_1 & -z_1\\ x_1 & y_1 & z_1 & w_1 \end{bmatrix}$$

$$q_1 R_{row}(q_2) = [\, x_1 \; y_1 \; z_1 \; w_1 \,] \begin{bmatrix} w_2 & -z_2 & y_2 & -x_2 \\ z_2 & w_2 & -x_2 & -y_2 \\ -y_2 & x_2 & w_2 & -z_2 \\ x_2 & y_2 & z_2 & w_2 \end{bmatrix}$$

### 1.4Computing Rotation Matrices from Quaternion's

The matrix form for left-multiplication by **q** is given as-

$$P.\, L_{row}(q) = [\, x_p \; y_p \; z_p \; 0 \,] \begin{bmatrix} w_q & z_q & -y_q & -x_q \\ -z_q & w_q & x_q & -y_q \\ y_q & -x_q & w_q & -z_q \\ x_q & y_q & z_q & w_q \end{bmatrix}$$

and a matrix form for right-multiplication by q'.

$$q^{-1} = q' = [-x_q \; -y_q \; -z_q \; w_q]$$

$$P.\, R_{row}(q^{-1}) = [\, x_p \; y_p \; z_p \; 0 \,] \begin{bmatrix} w_q & z_q & -y_q & x_q \\ -z_q & w_q & x_q & -y_q \\ y_q & -x_q & w_q & z_q \\ -x_q & -y_q & -z_q & -w_q \end{bmatrix}$$

The resultant rotation matrix is the product of these two matrices.

$$q * P * q' = q * (P * q') = q * (P\, R_{row}(q'))$$
$$= (P\, R_{row}(q'))\, L_{row}(q) = P\, (R_{row}(q')\, L_{row}(q))$$
$$= P\, Q_{row}(q)$$
$$Q_{row} = R_{row}(q^{-1}) \cdot L_{row}(q)$$

$$= \begin{bmatrix} w_1 & z_1 & -y_1 & x_1 \\ -z_1 & w_1 & x_1 & y_1 \\ y_1 & -x_1 & w_1 & z_1 \\ -x_1 & -y_1 & -z_1 & w_1 \end{bmatrix} \begin{bmatrix} w_1 & z_1 & -y_1 & -x_1 \\ -z_1 & w_1 & x_1 & -y_1 \\ y_1 & -x_1 & w_1 & -z_1 \\ x_1 & y_1 & z_1 & w_1 \end{bmatrix}$$

$$= \begin{bmatrix} w^2+x^2-y^2-z^2 & 2xy+2wz & 2xz-2wy & 0 \\ 2xy-2wz & w^2-x^2+y^2-z^2 & 2yz+2wx & 0 \\ 2xz+2wy & 2yz-2wx & w^2-x^2-y^2+z^2 & 0 \\ 0 & 0 & 0 & w^2-x^2-y^2+z^2 \end{bmatrix}$$

So using this matrix, compute $P_{rotated}$ :

$$P_{rotated} = P\; Q_{row}$$

### 1.5 Key Generation Scheme

The scheme is based on a rotation matrix calculated from a quaternion-key *q* .The idea of the proposed scheme is to treat elements of each column of the rotation matrix as coefficients *x, y, z* of subsequent quaternion's, from which other rotation matrices can be calculated. To obtain the coefficient *w* of subsequent quaternion's, we compute the arithmetic mean from the six remaining elements of the rotation matrix that were not used to determine coefficients *x, y, z* .

Let us assume that the first quaternion-key *q* from which a rotation matrix was calculated is called the initial quaternion key $q_0$. After grouping the elements of the initial rotation matrix ($q_0$) we will be able to obtain three quaternion-keys of the 1st order: $q_{11}$, $q_{12}$ and $q_{13}$.

From these quaternion's we can calculate three rotation matrices from which we will be able to calculate nine quaternion-keys of the 2nd order. If we assume *n* as a rotation order, then the number of obtained quaternion-keys for the appropriate order *n* is equal to $3^n$. Before generating rotation matrices from quaternion keys, we first rotate every quaternion-key around a different quaternion *f*. Quaternion's *f* are calculated from a random quaternion Julia set. These quaternion's are comprised of four non-zero components, in contrast to where each quaternion *f* was a pure quaternion.

We used the Quat generator for visualization of quaternion Julia sets in 3D space as color images. The fractal image is divided into smaller fragments. The number of fragments depends on the size of the required key space, e.g. for order = 2 we will produce 9 quaternion-keys of the 2nd order, thus we will need to divide the fractal image into $(9 + 3) \cdot 4 = 48$ fragments (9 for quaternion-keys of the 2nd order, 3 for quaternion-keys of the 1st order and 4 for the components *w, x, y, z* of quaternion *f*). From each of the four fragments a different quaternion *f* is calculated.

The below figure shows the process of calculating 1st order quaternion-keys from an initial rotation matrix

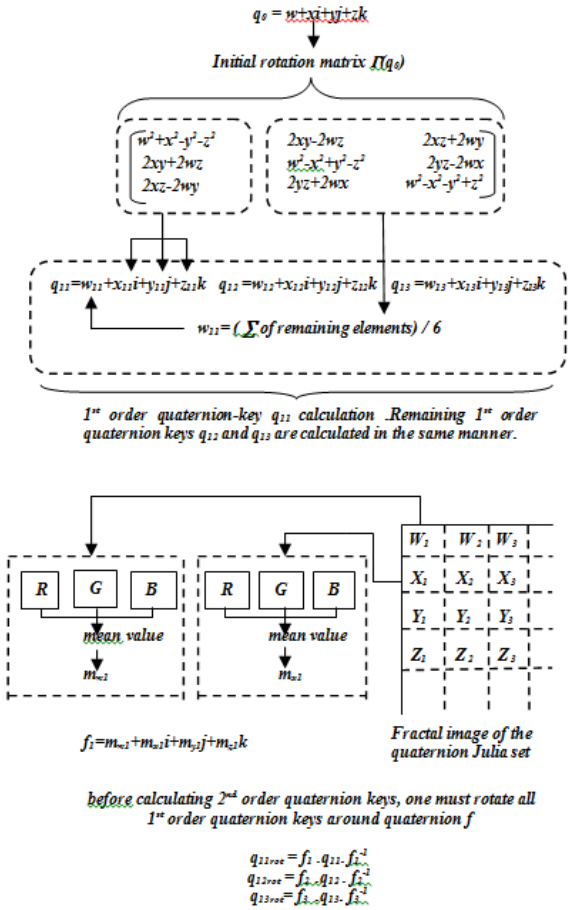



**Fig.2 Process of calculating 1st order quaternion-keys from an initial rotation matrix**

## II. LITERATURE REVIEW

### 2.1 Secure Quaternion Feistel Cipher for DICOM Images [3]

Secure quaternion Feistel cipher (S-QFC) algorithm holds the idea of a modified Feistel network with modular arithmetic and the utilization of exceptional properties of quaternion's to perform rotations of sequences of data in 3D space for each of the cipher rounds. A new more secure key generation scheme which is based on quaternion Julia sets is utilized and both-sided modular matrix multiplications are used for the encryption and decryption process. DICOM (Digital Imaging and Communications in Medicine) is the worldwide standard for medical images and related data. It characterizes the configurations for medical images that can be exchanged with the data and quality necessary for clinical use. The following are its key components-

#### *A. Computation Speed*

The S-QFC algorithm is slower than the F-QFC algorithm, yet at the same time a lot quicker than the AES algorithm. The minimization of the computation speed of the S-QFC algorithm is essentially caused by additional matrix multiplications presented in the encryption and decryption processes.

#### *B. Avalanche Effect*

To quantify the avalanche effect of the F-QFC and S –QFC algorithms, we compared a default cipher text (encrypted DICOM image) with differently modified cipher texts. These Modified cipher texts were obtained by changing 1 bit in the underlying key $q_0$, 1 bit in any quaternion-key $q_i$ , 1 bit in the key matrix **K** (only for the S-QFC algorithm) and 1 bit in plaintext **B**. A strict avalanche effect happens when a solitary piece change in plaintext or in a key can change half of the bits in the cipher text image.

For the S-QFC algorithm, a 1 bit change in plaintext or any key will deliver a cipher text that yields a nearly 50% (i.e. range of 49,973%-50,019% for 512 × 512 DICOM image) difference in its binary form with reference to the default cipher text. For the F-QFC algorithm the avalanche effect is absent when changing 1 bit in the plaintext (due to the lack of key matrix **K**).

#### *C. Key Sensitivity*

Key sensitivity shows how much an encrypted image is sensitive to a change in the key, with higher sensitivity being desirable. For any secure cryptosystem, decryption algorithm should not decrypt a cipher text image correctly, even if there is only a 1 bit difference between the provided key and the genuine key [10]. The outcomes demonstrates that the S-QFC algorithm possesses strong key sensitivity

#### *D. Key Space*

For the S-QFC algorithm let us consider three different scenarios. Let us assume that for each scenario we encrypt data with a single key matrix **K** of size $m \times m$ and $n$ higher order quaternion-keys. Each element of the matrix **K** as well as all four components of each quaternion-key are integers in the range of 0-255 and can be represented by an 8-bit binary number. In the first scenario, a potential attacker only knows the number $n$ of encryption keys and the size $m$ of matrix **K**. The possible key space can be written as-

$$2^{8\cdot(m\wedge 2+4n)} = 10^{2.4\cdot(m\wedge 2+4n)}$$

#### *E. Known Plaintext Attack*

The matrix multiplication on both-side by the key matrix **K**, introduced in the S-QFC algorithm, provides robustness against known plaintext attack. The proposed plan could additionally be enhanced by presenting an option, pseudorandom division of a fractal image or by adapting the algorithm to match a different Clifford algebra.

#### *F. Encryption and Decryption Processes*

Because of 16-bit input data (DICOM image) it is necessary to first decompose the image into two 8-bit gray-tone images of the same dimensions. Each image is stored as a matrix **B**. Each element of this matrix is an integer in the range of 0-255. With the end goal of the algorithm, the matrix **B** should be rewritten as a matrix with $m$ rows and $2m$ columns. The representative matrix **B** ($m \times 2m$) is split into a pair of square matrices, $\mathbf{L_0}$ and $\mathbf{R_0}$, where both matrices are of the similar size $m \times m$. These matrices are then composed as quaternion's $\mathbf{L_0}$ **and** $\boldsymbol{R_0}$.

It might be noticed that the symbol || is used to place the vector part of a quaternion $L$ adjacent to the vector part of quaternion $R$. The value $n$ demonstrates the number of rounds in the cipher. Each round is encrypted with a different quaternion-key $q_i$ where $i = 1, 2, \ldots, n$. The multiplication by a key matrix **K** is performed on all three components of quaternion $R_{i-1}$. The matrix **K** is of size $m \times m$ and is filled with random integers in the range of 0-255.
The general structure of the S-QFC algorithm is demonstrated as follows.

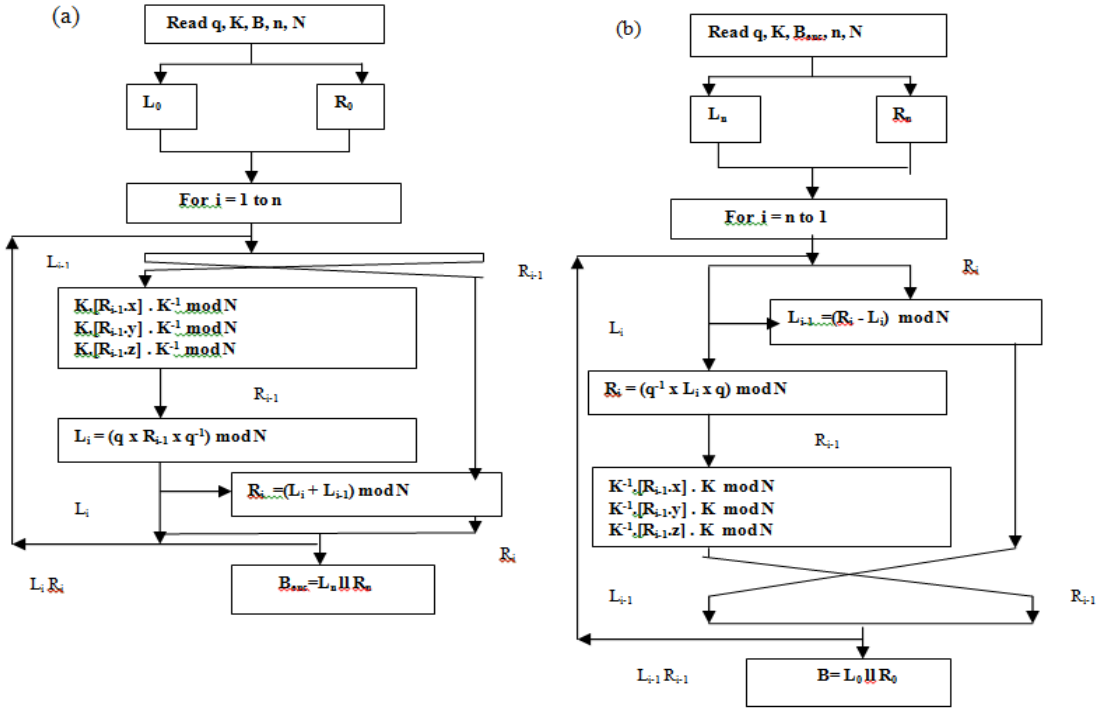



**Fig.3 Process of encryption (a) and decryption (b) for the S-QFC algorithm**

The modular inverse of matrix **K** is calculated based on the Gauss-Jordan elimination algorithm, for which the matrix **K** cannot be singular, modulo *N*. This means that the modular determinant of matrix **K** must be co-prime to *N*. In our case the modulus value equals $N = 257$ which is a prime number, thus the probability that a randomly chosen key matrix will be singular is significantly reduced. Due to the both-sided matrix multiplication, each output element (the result of the matrix multiplication) is related to every element of the input data stored in the $R_{i-1}$ quaternion. Thus, a strong avalanche effect will be present in the encrypted data when changing a single bit of the input data.

S-QFC algorithm successfully removes several security flaws of the F-QFC algorithm such as:

• ***Weak keys and insufficient key space***- both addressed by a new key generation scheme based on quaternion Julia sets

• ***No diffusion property and vulnerability to the known plaintext attack***- both addressed by a both sided modular matrix multiplication by a key matrix **K** during the encryption and decryption processes. It maintains a satisfactory computation speed, especially when compared to AES-ECB.

### 2.2 A New Quaternion-Based Encryption Method for DICOM Images [4]

The Author have scrutinized and slightly modified the concept of the DICOM network to point out the best location for the proposed encryption scheme, which significantly improves speed of DICOM images encryption in comparison with those originally embedded into DICOM advanced encryption standard (AES) and triple data encryption standard algorithms (3-DES).The proposed algorithm decomposes a DICOM image into two 8-bit gray-tone images in order to perform encryption. It uses special properties of quaternion's to perform rotations of data sequences in 3D space for each of the cipher rounds. The images are written as Lipschitz quaternion's, and modular arithmetic was implemented for operations with the quaternion's. This quaternion algorithm is designed to encrypt images (both color and gray-tone), but it can also be used to encrypt textual data.

### *A. DICOM Network*

Security of Medical images/data is an issue that is evolving simultaneously along with technical development. Due to the rising importance of information, secured data transfer and storage have become a serious problem. Although the DICOM standard defines security in several fields, security in commonly used solutions is focused on outside data transferring. The exemplary DICOM solution makes use of TLS, HTTPS, and VPN protocols (defined in the Secure Transport Connection Profiles) for data transmission and data access, which obviously correspond to sharing data through the Internet (Fig.4.a). The model introduced here accepts that both the internal network, i.e. the network which provides communication between the DICOM storage server and medical equipment, and the storage itself are safe..

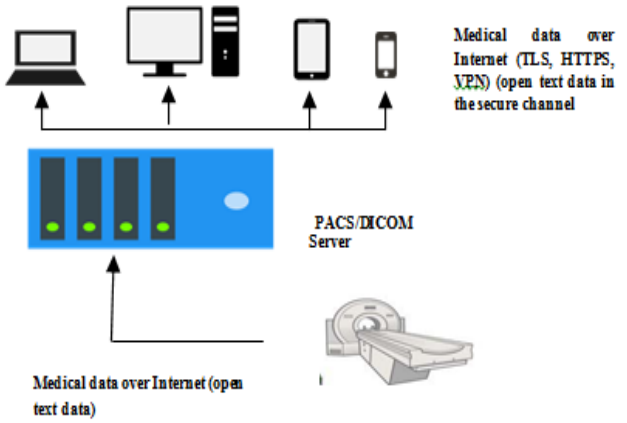
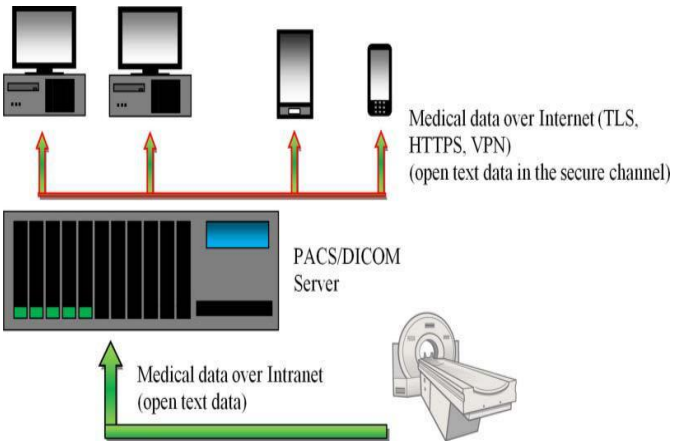



**Fig.(4.a) Exemplary DICOM network.**




With the end goal to enhance security of the information they proposed utilizing an encryption server (a front-end server) (Fig.4.b) whose primary task is to catch, encrypt (with the proposed quaternion method) and hand-off the medical data to the storage server without composing data on any media storage. The utilization of a front-end server eliminates the basic weakness of the standard DICOM network model, i.e. the possibility of accessing some unsecured data on the storage server. Furthermore, the front-end server can be utilized for access control and, if safe and necessary, for decrypting data, e.g. for VPN users.

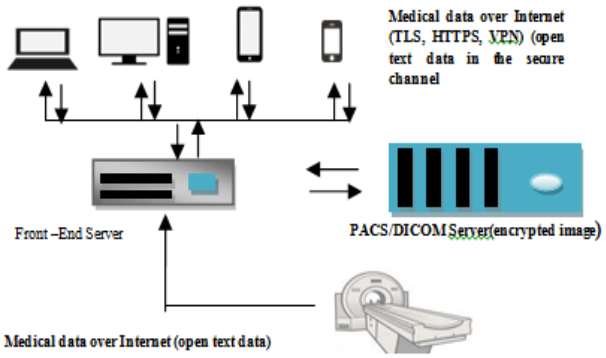



**Fig.( 4.b) Proposed DICOM network equipped with a front-end server**

### *B. Encryption and Decryption process*

In order to encrypt a 16-bit DICOM image of size $r \times c$ we first need to decompose it into two 8-bit gray-tone images, each of size $r \times c$ (Fig. 4). The two images that are obtained are treated as input data for our algorithm. Each image of the pair will be encrypted separately. Decomposition is necessary because of the current limitation of our algorithm, i.e. 8-bit input data. Thus, in case of 32-bit image data decomposition into four 8-bit gray-tone images would be necessary.

Let us now consider a plaintext **B**, which will refer to the one of the two obtained gray-tone images. This plaintext can be written as matrix **B** of equal size ($r \times c$). Each element of matrix **B** is a value in the range of 0-255. For the purpose of the algorithm, matrix **B** should be rewritten as a matrix with m rows and $2m$ columns. The obtained new matrix **B** ($m \times 2m$) is split into a pair of square matrices, $\mathbf{L_0}$ and $\mathbf{R_0}$, where both matrices are of the same size $m \times m$.

These matrices are then written as quaternion's $L_0$ and $R_0$. It may be noted that the symbol || is used to place the vector part of a quaternion $L$ adjacent to the vector part of quaternion $R$. The value $n$ indicates the number of rounds in the cipher. Each round is encrypted with a different quaternion-key $q_i$ where $i = 1, 2, \ldots, n$. The unique, round keys are provided by the rotation matrix algorithm.
For encryption/decryption of each 8-bit image a unique pre-defined initial quaternion-key is needed to generate a unique key space. At the end of decryption process a quaternion $B$ is obtained. Its vector part stores information about the decrypted 8-bit image.

To reconstruct matrix **B** one must write the vector part of the decrypted quaternion $B$ as a vector of size $1 \times 2m^2$ to discard the additional elements (needed to form the matrix $m \times 2m$ on the sender's side). Then, the vector is rewritten into matrix **B** of size equal to the size of original image

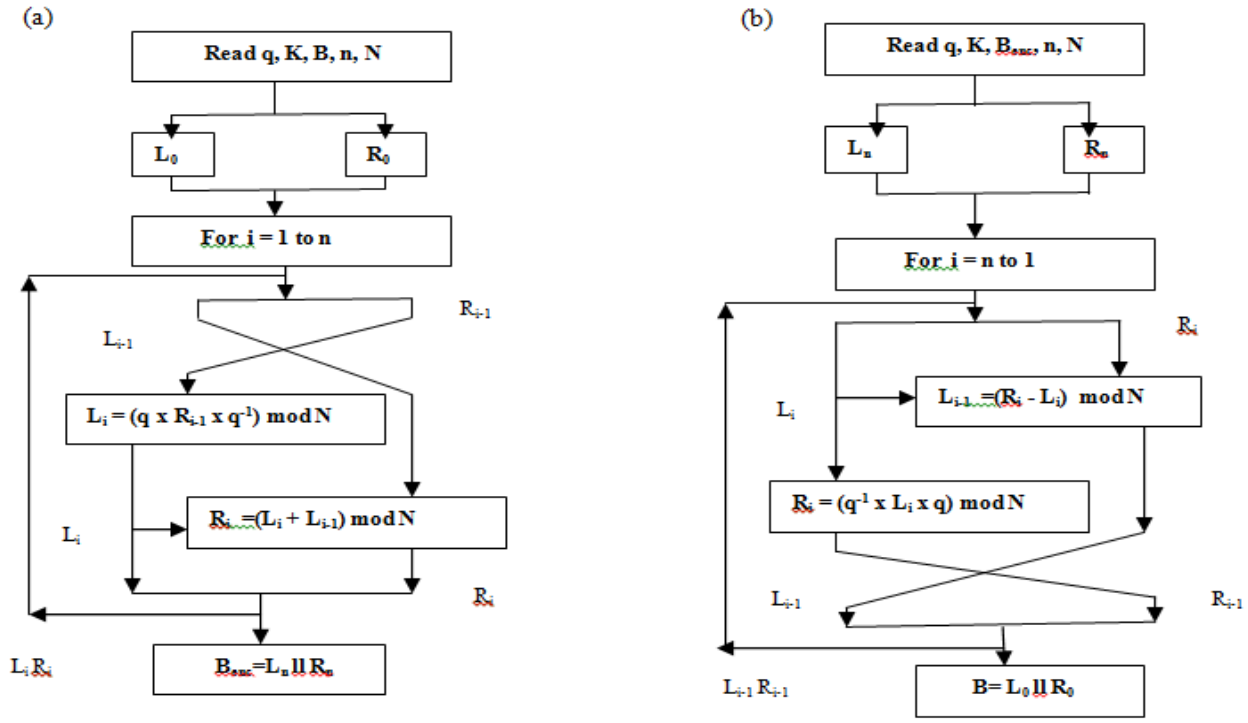


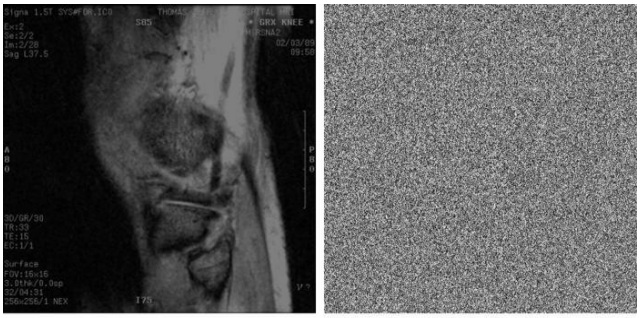

**Fig.5 Process of encryption (a) and decryption (b) for the proposed quaternion scheme.**

### *C. Computation Speed*

One of the main advantages of this algorithm is its fast computation speed. A higher number of rounds will slightly degrade the performance in elapsed time but will guarantee higher security (due to a larger key-space used). Moreover, in a practical scenario encryption with AES could take even more time, especially considering the fact that a more secure scenario will be required, e.g. AES-CBC, AES-CTR, AES-OFB, or AES-OCB

### *D. Avalanche Effect*

In order to study the avalanche effect, let us consider a DICOM image as a plaintext. If we change 1 bit in one of the calculated round keys (quaternion's of higher order), we will obtain a new cipher text (encrypted image) .Such a cipher text will yield a nearly 50% difference in its binary form in reference to the original cipher text. The same situation is achieved when changing 1 bit in the initial encryption key (initialization quaternion). The difference in the binary form of the modified cipher text and the original cipher text is also very close to 50%. Let us now consider an example as shown in Fig 6.a We want to encrypt a DICOM image. We use 9 rounds of the proposed algorithm in the encryption process. Each round has its own unique key (quaternion of higher order). We now assume that the attacker knows 8 unique keys and possesses substantial knowledge about the last key (let us assume the only difference is 1 bit in the binary form in the value *y* of the last quaternion-key). Decryption of the image is performed by the attacker in such a scenario as shown in Fig 6.b Despite the fact that the attacker has knowledge about the encryption keys, he or she will not be able to properly decrypt the image.



(a) (b)

**Fig.6 Unauthorized decryption: (a) – original image, (b) – decrypted image using one key that is different in its binary form (1 bit) from the original key. Image source: Computer Assisted Radiology and Surgery 11th International Symposium and Exhibition, Berlin, June 25-28, 1997.**

It could further be improved, by increasing the computation speed even further by modifying the algorithm structure, by bypassing the decomposition process and introducing parallel operations. Moreover, the main focus now is to

introduce additional operations to the cipher, thus further increasing its robustness capabilities.

| 1 bit change in | F-QFC | S-QFC |
|---|---|---|
| Initial key: $q_0$ | ~50% | ~50% |
| Encryption key: $q_1$ | ~50% | ~50% |
| Key matrix: K | - | ~50% |
| Plaintext: B | ~0% | ~50% |

**Fig.7 Comparison of the Avalanche Effect for the F-QFC and S-QFC Algorithms**

### 2.3 Secure Signals Transmission Based on Quaternion Encryption Scheme [5]

Author has introduces Quaternion Encryption Scheme (QES) as a means to secure signal transmission over wireless networks that are vulnerable to attacks. The sampling data of the signals are detailed in a 3-D vector space and are rotated by another key vector space parameter utilizing quaternion representation.

Quaternion is utilized to generate a four-dimensional encryption key that significantly eliminates the risk of eavesdropping. Furthermore, the quaternion's have capability to provide a meditative encryption system for wireless images or voice transmissions. A PC-based simulation conducted to scrutinize carefully the capability of QES in insuring the highest level of security is reported.

#### *A. Key Generation*

The quaternion representation, henceforth, provides an intrinsic means for robust and resilient encryption system because the quaternion, implies four integrated parameters (keys) independently [11]. These keys might be any function or any random number with a variable size. If any key is implemented incorrectly than the system will fail to disentangle its encrypted signal.

#### B. Encryption model

Suppose that the signal *S* is divided into frames, where $S = \{f_1, f_2, .., f_n\}$. Each frame *f* has a set of 3x3 elements array and suppose that these elements are formed as 3-D vector space, as shown in Fig. We assume that the number of samples per frame is limited to 3x3, and let $f_1 = \{(r_1, s_1, t_1), (r_2, s_2, t_2), (r_3, s_3, t_3)\}$. That is, the signal S can be written as $S= \{(r_1, s_1, t_1), (r_2, s_2, t_2), (r_3, s_3, t_3)\ldots..( r_m, s_m, t_m)\}$ Then the frame *f* is represented by a triad of 3-vectors matrix alignment which is equivalent to the signal data **B**

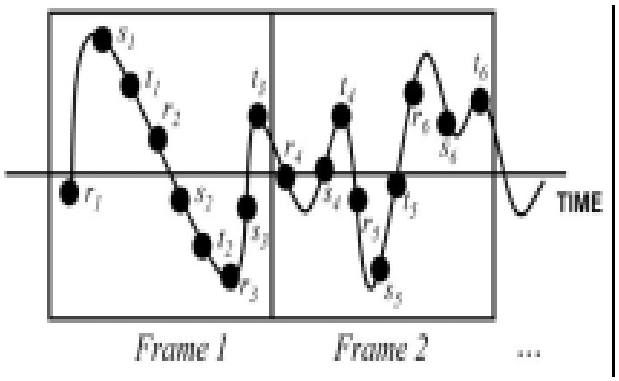



**Fig.8 The signal shows the frame and sampling orders**

In order to perform encryption of the signal data, we implement a different Quaternion Key Order (*QKO*) for every frame in the rotation matrix Γ*(q)*. This method is used to make the frame like a random pattern. We can generate $3^n$ quaternion key orders, where *n* is the order number. The initial order of quaternion key (for $n$=0) is $q_{01}= (w_{01},x_{01},y_{01},z_{01})$ and the rotation matrix of this key is identical. The scalar value *w01* is to be any integer number or sequence, for complexity concern *wnm* is set to zero for order n ≠0. However, it's possible to choose any value to *wnm* for order n ≠0.

$$wnm\begin{cases} \text{=any value} & n=0,\ m=1,2,3 \\ =0 & n=\neq 0,\ m=1,2,3 \end{cases}$$

Now, we need to create 1st *QKO* using Γ*(q)* that has 3 quaternion keys which are $q_{11}$, $q_{12}$ and $q_{13}$, are constructed using first, second and third column of Γ*(q)* ,respectively, and can be expressed as

$$\begin{aligned} q_{11} &= ( w_{11} ,x_{11} , y_{11} ,z_{11} ) \\ &= ( 0 , w^2+x^2-y^2-z^2 , 2(xy+wz), 2(xz-wy) ) \\ q_{12} &= ( w_{12} ,x_{12} , y_{12} ,z_{12} ) \\ &= ( 0 , 2(xy-wz), w^2-x^2+y^2-z^2 , 2(yz+wx) ) \\ q_{13} &= ( w_{13} ,x_{13} , y_{13} ,z_{13} ) \\ &= ( 0 , 2(xz+wy), 2(yz-wx) ,w^2-x^2-y^2+z^2 ) \end{aligned}$$

From $q_{11}$ key, a new rotation matrix $\Gamma(q_{11})$ is constructed by substituting $x_{11}$, $y_{11}$ and $z_{11}$ with every *x*, *y* and *z* value, respectively. This will be the same to construct $\Gamma(q_{12})$ and $\Gamma(q_{13})$. Recursively, the value of the new columns in the rotation matrices of $\Gamma(q_{11})$, $\Gamma(q_{12})$ and $\Gamma(q_{13})$ are used to determine encryption keys or quaternion keys of order 2nd and 3rd and so on. Correspondingly, we can construct sequence of quaternion keys in any order ($q_0$, $q_{11}$, $q_{12}$, …, $q_{n3}^n$). Thus, the rotation matrices, which are associated with quaternion keys, implement encryption operation to the data frames successively.

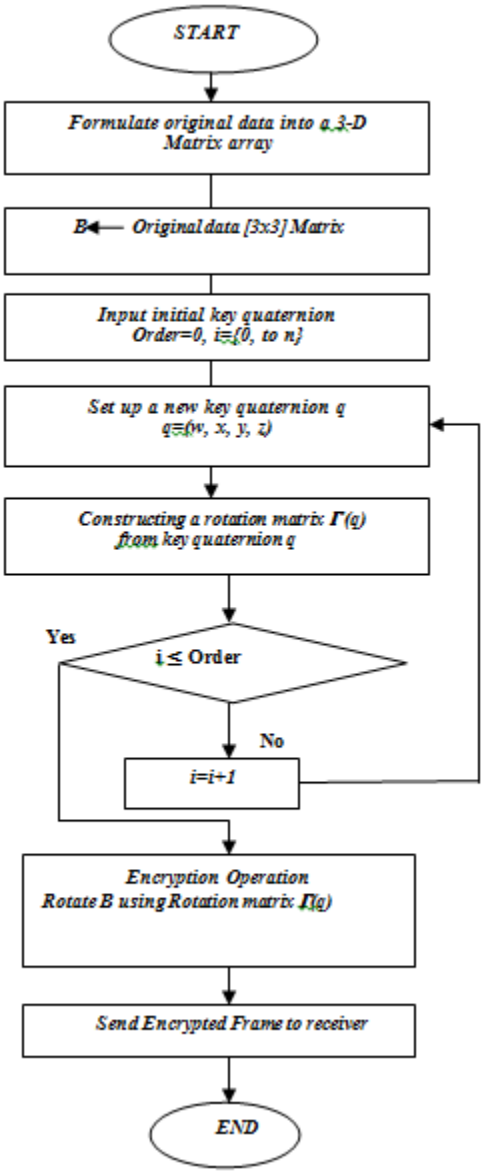



**Fig 9 . Quaternion encryption algorithm**

Furthermore, the quaternion encryption scheme (QES) can be an appropriate model for asymmetric-key encryption for constructing ciphers that are effectively impossible to break.

### 2.4 Digital Fingerprinting Based on Quaternion Encryption Scheme for Gray-Tone Images [6]

A new idea of digital images finger-printing is proposed. This method is based upon quaternion encryption in the Cipher Block Chaining (CBC) mode. Quaternion's are hyper-complex numbers of rank 4 and thus often applied to mechanics in three dimensional spaces. The encryption algorithm described in the paper is designed for gray-tone images but can easily be adopted for color ones.

For the encryption purpose, the algorithm uses the rotation of data vectors presented as quaternion's in a three-dimensional space around another quaternion (key). At the recipient's end, a small amount of unnoticeable by human eye errors occurs in the decrypted images. These errors are then used as a user's digital fingerprint for the purpose of traitor tracing in case of copy-right violation. A PC-based simulation was performed to scrutinize the potential presented quaternion encryption scheme for the implementation of digital fingerprinting

#### *A. Cipher Block Chaining Mode of Encryption*

The model presented in previous papers concerns encryption in Electronic Code Book mode, which means that the blocks of data **B** are encrypted and decrypted separately. However, the security issue of the encryption could be improved significantly if a dependency between subsequent data blocks was introduced. The solution is to adapt the encryption method to use the Cipher Block Chaining encryption mode, i.e., a bitwise binary addition of data matrices **B** and matrices $\mathbf{B}_{rot}$, which have been acquired via quaternion encryption. In the first step a random initialization matrix **IM** is necessary. This matrix, **IM**, must be of the same size as all matrices **B**. As a result a new matrix $\mathbf{B}_{mod}$ is obtained which will be of the same dimension as matrix **B** and the values of its elements will be randomized. At this point a quaternion encryption will begin where the new data matrix $\mathbf{B}_{mod}$ will be converted to a quaternion $\mathbf{B}_{mod}$ and then rotated. However, to make a bitwise binary addition into such an encryption process one have to remember the fact that the values obtained in matrices $\mathbf{B}_{rot}$ are not of a decimal form. This is due to the process of quaternion encryption, which is why the binary addition must support not only decimal numbers but also floating point numbers.

#### *B. Fingerprinting as an Additional Feature*

There are two complementary methods for multimedia content and copyright protection. The first method is an encryption, whose main objective is to provide a protected data confidentiality. This technique ensures that only users with the appropriate decryption keys will be able to decrypt the transmitted multimedia content and use it. Unfortunately, even the best encryption standard does not assure sufficient protection because after the decryption a user with access to the content is able to redistribute it without the author's permission and violate copyrights in the process. The second method is digital fingerprinting, which involves the embedding of an additional hidden data into multimedia content. These data are called fingerprints and each fingerprint identifies one individual user of the system.

The quaternion encryption scheme presented can be used not only for multimedia data encryption, but also as a tool to detect pirates by utilizing error patterns which occur during the decryption. The first stage is multimedia content

encryption ordered by users. Copies intended for each user are encrypted separately with different keys and then sent via multiple unicast transmissions. Keys must also be provided separately for each user in a secure manner.

In the second stage, each user performs decryption of the multimedia content. The artifacts which occur in decrypted images are dependent on the encryption key which makes them different for each user and will be considered as fingerprints. Figure 4 represents this scheme of digital fingerprinting via quaternion encryption. The last stage takes place after capturing an illegally distributed copy. The fingerprint must be extracted from the illegal copy and then the detection of pirates is performed by non-blind detection.

Fingerprint extraction is done by calculating the difference between the original image and the pirate copy. In this case, original unmarked data is used as a reference in the extracting process a fingerprint from an illegal copy. Pirates identification of is based on the correlation analysis of the extracted fingerprint and the fingerprints of all users. If the correlation coefficient of the fingerprint extracted from the pirated copy and the $i^{th}$ user's fingerprint exceeds a detection threshold, the $i^{th}$ user is considered as guilty. Thus, there is a need to define an appropriate detection threshold.

Fingerprint embedding in a discrete cosine transforms or wavelet transform domain is the purpose of proposed further studies.

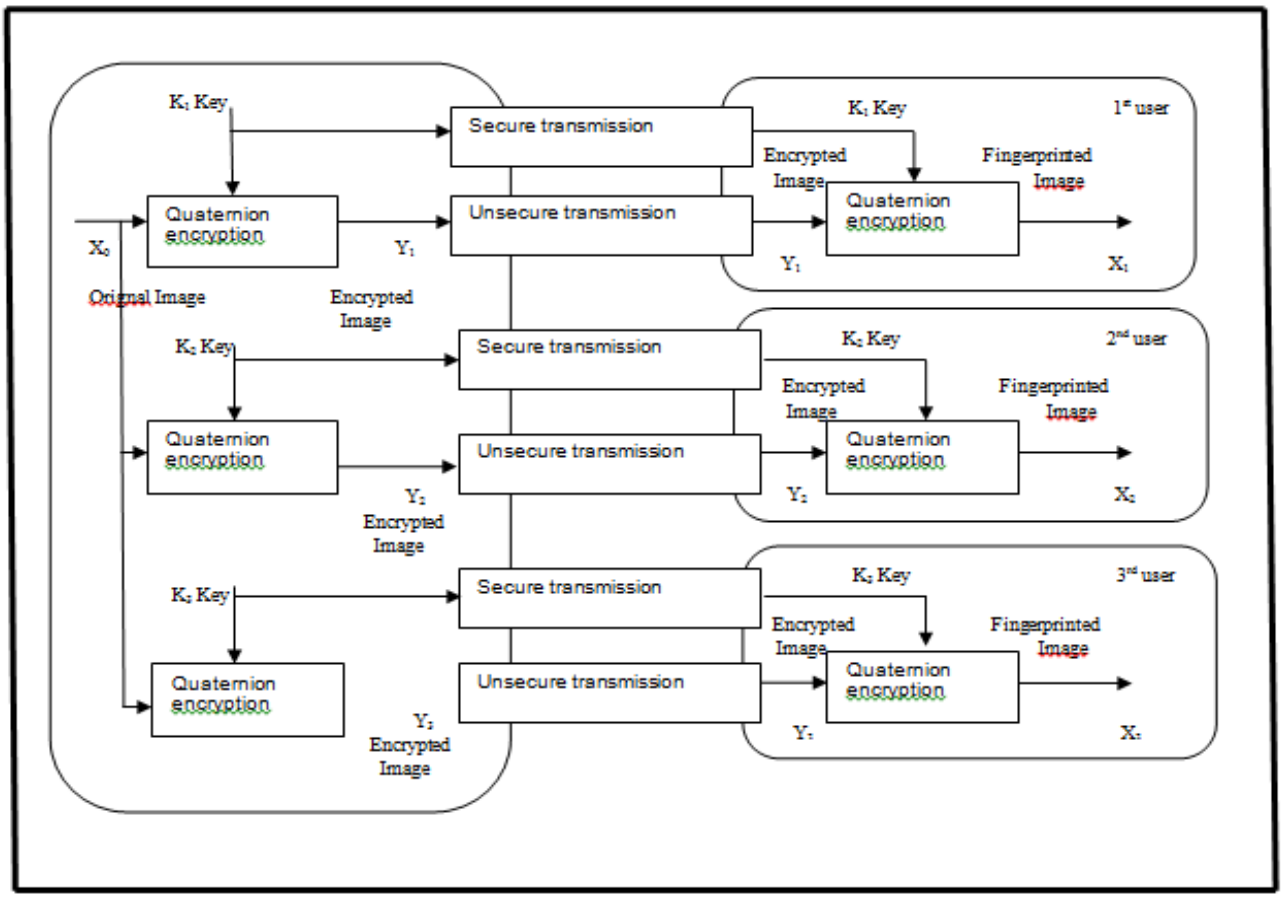



***Fig. 10.*** **General scheme of the multimedia distribution system based on a quaternion encryption.**

### 2.5 Quaternion Feistel Cipher with an Infinite Key Space Based on Quaternion Julia Sets [7]

The Quaternion Feistel Cipher (QFC) having an infinite key space based on quaternion Julia sets is proposed whose basic structure is based on the scheme proposed in 2012 by Sastry and Kumar. The proposed algorithm uses special properties of quaternion's to perform rotations of data sequences in 3D space for each of the cipher rounds. It also uses Julia sets to form an infinite key space. The plaintext is divided into two square matrices of equal size and written using Lipschitz quaternion's. A modular arithmetic was implemented for operations with quaternion's.

The proposed algorithm is designed to encrypt images (both color and gray-tone) but it can also be used to encrypt

textual data. For the purpose of this paper an implementation for RGB color images is presented. Let us now consider a plaintext B, which will be treated as RGB color image data. The plaintext can be written as three matrices B: $B_R$, $B_G$, $B_B$. Each matrix B is of equal size with the image. Each element of all three matrices B is a value in the range of 0–255. For the purpose of the algorithm, all three matrices B should be rewritten as matrices with m rows and 2m columns, where m is calculated according to the rule:

$$\mathbf{m= \sqrt{\ width_B . height_B\ /\ 2}}$$

If the number of elements in such matrices exceeds the original amount of the images' pixels, then the additional elements are filled with random numbers in the range of 0–255. The three obtained matrices B (m × 2m) are split into three square matrices: $\mathbf{L_{0R}}$, $\mathbf{L_{0G}}$, $\mathbf{L_{0B}}$ and three square matrices: $\mathbf{R_{0R}}$, $\mathbf{R_{0G}}$, $\mathbf{R_{0B}}$, each of size m×m. All six square matrices are then written as components of two quaternions, $L_0$ and $R_0$ using the following rule:

$\boldsymbol{L_0 = w_0 + x_0 i + y_0 j + z_0 k,}$ where
$\boldsymbol{w_0 = 0,\ x_0 = [L_{0R}],\ y_0 = [L_{0G}],\ z_0 = [L_{0B}].}$

$\boldsymbol{R_0 = w_0 + x_0 i + y_0 j + z_0 k,}$ where
$\boldsymbol{w_0 = 0,\ x_0 = [R_{0R}],\ y_0 = [R_{0G}],\ z_0 = [R_{0B}].}$

The basic equations governing the encryption and decryption in proposed scheme are very similar in concept to the one presented in previous papers.
In this work, matrix multiplication is substituted by quaternion multiplication:

$L_i = (q \cdot R_{i-1} \cdot q^{-1}) \bmod N,$
$R_i = (L_{i-1} + L_i) \bmod N \text{ for } i = 1 \text{ to } n,$
$R_{i-1} = (q^{-1} \cdot L_i \cdot q) \bmod N,$
$L_{i-1} = (R_i - L_i) \bmod N \text{ for } i = n \text{ to } 1$

### *A. Avalanche Effect*

In order to study the avalanche effect, let us consider a color Lena image as a plaintext. If 1 bit in one of the calculated round keys (quaternion-keys of higher order) is changed, a new cipher text (encrypted image) will be obtained. Such a cipher text will yield a nearly 50% difference in its binary form in reference to the original cipher text. The same situation is achieved when changing 1 bit in the initial encryption key (initialization quaternion). The difference in binary form of the modified cipher text and the original cipher text is also very close to 50%.

### *B. Computation Speed*

One of the main advantages of proposed algorithm, is its fast computation speed. Moreover, in a practical scenario, encryption with AES could take even more time, especially considering the fact that a more secure implementation will be required, e.g., AES-CBC, AES-CTR, AES-OFB, or AES-OCB.

### 2.6 Digital fingerprinting for color images based on the quaternion encryption scheme [8]

In this paper a digital fingerprinting as an additional feature of the quaternion encryption algorithm is proposed. The first goal was to demonstrate the potential application for error patterns which occur during the decryption process. The method is designed to encrypt color images. In this case the use of quaternion calculus is particularly beneficial, as a single quaternion can include information about all three colors of the pixel. This is very effective in terms of the number of calculations required to encrypt an entire color image. Additionally, it is important to note that when using the quaternion encryption algorithm the number of possible encryption keys is particularly large. This is due to the fact that both the rotation order and the 4 initialization quaternion parameters can be changed Thus, attacks, especially on data encrypted with a high rotation order, are possible but rather computationally complex.

### *A. Quaternion encryption*

The proposed encryption scheme is based on quaternion rotation. In order to perform a quaternion rotation we need to create a quaternion (key) around which we will be rotating another quaternion (data). Let us consider two quaternions, $q=[w,x,y,z]^T$ and $P=[0,a,b,c]^T$, where a sub-vector of P: $P = [a\ b\ c]^T$ stores data which we want to rotate around quaternion q (key). The obtained quaternion Prot will be a spatial mapping of the rotated data vector P. There are two possible ways of implementing a proposed quaternion encryption. The first approach is entirely based on quaternion calculus. The alternative method introduces a rotation matrix (Nagase et al., 2004a; Nagase et al., 2004b; Dzwonkowski and Rykaczewski, 2012;Dzwonkowski and Rykaczewski, 2013)which enables the implementation of quaternion encryption via a matrix multiplication. The second approach will be referred to as a matrix method. The matrix method is easier to implement, however, it lacks the fast computing advantages of the quaternion method (Goldman, 2009; Goldman, 2011). The quaternion method is entirely based on the quaternion rotation. It is possible to optimize the rotation process by extending the vector part of the quaternion P in order

$$P=[0,\begin{bmatrix}a\\b\\c\end{bmatrix}] \longrightarrow B=[0,\begin{bmatrix}[a_1\ \ a_2\ \ a_3]\\[b_1\ \ b_2\ \ b_3]\\[c_1\ \ c_2\ \ c_3]\end{bmatrix}]$$

In this way we can encrypt a greater amount of data via a single operation. The encryption and decryption for the quaternion method with the extended quaternion B is shown below

$$\mathbf{B_{rot} = q \cdot B \cdot q^{-1},}$$

$\mathbf{B = q \cdot Brot \cdot q^{-1}}$,

Where, quaternion B stores the data (e.g. pixel values), quaternion q is the key, and quaternion Brot stores the encrypted data. The key generation process is handled independently by treating the component values of the quaternion q as random floating-point numbers, i.e. in the range of (0,1).Every component of the vector part of quaternion B can store information about each color channel, and there is no restriction as to the component's size. In fact, we can put, e.g. matrices of size 3x3 as components for imaginary units i, j and k of quaternion B in order to increase the amount of data encrypted in a single rotation, as shown:

$$B_{rot} = q.\ [0,\ \begin{pmatrix} [\ red\ ]_{3x3} \\ [\ green\ ]_{3x3} \\ [\ blue\ ]_{3x3} \end{pmatrix}\ ]\ .\ q^{-1}.$$

For the matrix method it is necessary to first calculate the rotation matrix. The rotation matrix is always 3x3 in m size, so it is possible to extend the data vector P in order to optimize the matrix multiplication, analogously to the quaternion method. Instead of data vector P, of size 3x1, we can use a matrix B of size 3x3. Encryption and decryption for the matrix method is shown below

$$\mathbf{B_{rot} = \Gamma(q)\ .\ B,}$$
$$\mathbf{B = \Gamma(q)^{-1}\ .\ B_{rot}}$$

Where, the rotation matrix Γ(q) is the key, B is the data matrix and $B_{rot}$ is the encrypted data matrix.

### *B. CBC mode and error occurrence*

The model presented in the paper (Nagase et al., 2004a) concerns encryption in electronic code book (ECB) mode, where quaternion's B are encrypted (rotated) and decrypted separately .However, security could be improved significantly by adapting the encryption method to operate in the cipher block chaining (CBC) mode of encryption. The CBC mode is implemented as follows. Before the rotation of the $k^{th}$ quaternion B there is a bit-wise binary addition of corresponding elements of components of the $k^{th}$ quaternion B and $(k-1)^{th}$ encrypted quaternion $B_{rot}$. In the first step a random initialization quaternion $B_{ini}$ is necessary. The components of this quaternion $B_{ini}$ must be of the same size as the components of quaternion B. The result of this bit-wise binary addition is the $k^{th}$ quaternion $B_{mod}$ and it will be encrypted, thus producing an encrypted $k^{th}$ quaternion $B_{rot}$. However, if we decide to use a bit-wise binary addition, we must take into account that the components of quaternion $B_{rot}$ are not in decimal form. We are able to introduce small and imperceptible patterns of errors on the receiver side during the decryption process. These errors will serve as the user's unique fingerprints.

### *C. Fingerprinting as an additional feature*

The quaternion encryption scheme presented in this paper can also be used as a tool to precisely detect pirates. The overall concept is as follows-

The image is encrypted for each user separately with a different key for each user. The encrypted copies are sent via multiple unicast transmissions. Each user receives a key in a secure manner (e.g. by the key distribution method based on the Quaternion Julia Set by Anand et al. (2009)) that allows to decrypt data. During the decryption, unnoticeable artifacts occur in the decrypted images. These errors are dependent on the encryption key, so they are different for each user and will be considered as unique fingerprints. It is worth noticing that the fingerprint embedding is performed on the receiver side. This highly reduces the computational costs for the distribution side in comparison to transmitter-side embedding schemes (Cox et al.,264 1997).

However, it is not as efficient as JFD methods because the distribution side still has to encrypt data for each user separately. Another advantage is that no special network equipment is required, as opposed to methods which embed fingerprints in the network, e.g. Ammar's method (Ammar and Judge, 2000). Identification of potential pirates is performed by the distribution side, which has access to the original image and its customer's fingerprints, so it is reasonable to use non-blind detection, such as in (Cox et al., 1997; Mishra 269 et al., 2012). Extraction of fingerprints is done by calculating the difference between the original image and the investigated copy.

It was indicated that fingerprints are vulnerable to compression because they are embedded in the original spatial domain (pixel values). A solution to this problem would be fingerprint embedding in the discrete cosine transform domain or the wavelet transform domain.

Another weak point is low scalability due to the fact that the method is not fully a joint fingerprinting and decryption method. Although fingerprints are embedded on the receiver side, the distribution side still has to perform multiple encryptions and to use multiple unicast channels. The last suggestion was to use code-based fingerprints, such as Tardos codes or Boneh-Shaw codes, in order to increase robustness against collusion attacks

## III. COMPARISON BETWEEN VARIOUS ENCRYPTION TECHNIQUES

| Parameters | Encryption Techniques | | | |
|---|---|---|---|---|
| | | 3-DES | AES | QFC |
| | **Computational Speed** | Low | Moderate | High |
| | **Key Length** | 56 bits | 128,192 and 256 bits | $2^{8.(m^2+4n)} \sim 10^{2.4.(m^2+4n)}$ |
| | **Key sensitivity** | weaker key sensitivity | Stronger key sensitivity than 3-DES only | Strong Key Sensitivity |
| | **Known Plaintext attack** | Vunerable to known plaintext attack | Vunerable to known plaintext attack | Robustness to known plaintext attack |

## IV. CONCLUSION

Cryptography is technique for secure communication, in this paper; it has been surveyed about the existing works on the Quaternion-Based Encryption. The quaternion based encryption scheme, significantly improves speed of images encryption in comparison with those originally embedded advanced encryption standard (AES) and triple data encryption standard (3-DES) algorithms. It uses special properties of quaternion's to perform rotations of data sequences in 3D space for each of the cipher rounds. Quaternion-Based Encryption works efficiently for both gray toned and color images .Its fast computation speed and large key length makes it to be used in various applications. Its large key space also makes it secure against the known plaintext attack. When utilizing the quaternion encryption algorithm the number of possible encryption keys is particularly large. Because of the fact that both the rotation order and the 4 initialization quaternion parameters can be changed. Thus possible attacks, especially on data encrypted with a high rotation order, are possible but rather complex. The method can be easily extrapolated to encrypt color images. The features of quaternion based encryption discussed above make it suitable for secure transmission of medical images over a non-secure medium. This paper highlights the application of quaternion based encryption in several fields such as for signal processing and for DICOM Images.

## REFERENCES


[1] T. Nagase, R. Koide, T. Araki, Y. Hasegawa, \A new Quadripartite Public-Key Cryptosystem", in Proc. Int. Symp. Commun. lnfom. Technol. ISCIT 2004, Sapporo, Japan, 2004, pp. 74-79.

[2] Jongchan Baek, Hayeong Jeon, Gwangjin Kim, And Soohee Han, "Visualizing Quaternion Multiplication," IEEE Access, Volume 5, 2017

[3] M. Dzwonkowski, and R. Rykaczewski, "Secure Quaternion Feistel Cipher for DICOM Images," IEEE Trans. Image Process., vol. 28, no.1, Jan. 2019.

[4] M. Dzwonkowski, M. Papaj, and R. Rykaczewski, "A new quaternion based encryption method for DICOM images," IEEE Trans. Image Process., vol. 24, no. 11, pp. 4614–4622, Nov. 2015.

[5] T. Nagase, M. Komata, and T. Araki, "Secure signals transmission based on quaternion encryption scheme," in Proc. IEEE Adv. Inf. Netw. Appl. (AINA), vol. 2, Mar. 2004, pp. 35–38.

[6] B. Czaplewski, M. Dzwonkowski, and R. Rykaczewski, "Digital fingerprinting based on quaternion encryption scheme for gray-tone images," J. Telecommun. Inf. Technol., vol. 2, pp. 3–11, Jul. 2014.

[7] M. Dzwonkowski and R. Rykaczewski, "Quaternion feistel cipher with an infinite key space based on quaternion Julia sets," J. Telecommun. Inf. Technol., vol. 4, pp. 15–21, Dec. 2015.

[8] B. Czaplewski, M. Dzwonkowski, and R. Rykaczewski, "Digital fingerprinting for color images based on the quaternion encryption scheme," Pattern Recognit. Lett., vol. 46, pp. 11–19, Sep. 2014.

[9] B. Czaplewski, "Joint fingerprinting and decryption method for color images based on quaternion rotation with cipher quaternion chaining," J. Vis. Commun. Image Represent., vol. 40, pp. 1–13, Oct. 2016.

[10] H. Liu, Z. Zhu, H. Jiang, and B. Wang, "A novel image encryption algorithm based on improved 3D chaotic cat map," in *Proc. Young Comput. Sci. (ICYCS)*, Nov. 2008, pp. 3016–3021.

[11] M. Dzwonkowski and R. Rykaczewski , "Quaternion encryption methods for multimedia transmission, a survey of existing approaches," Telecommun. Rev. Telecommun. News, vol. 7, pp. 668–671, Jul. 2016.

[12] F. Zhang, "Quaternions and matrices of quaternions," in Linear Algebra and Its Applications. New York, NY, USA: Elsevier, 1997, pp. 21–57.