Paper

# Digital Fingerprinting Based on Quaternion Encryption Scheme for Gray-Tone Images

Bartosz Czaplewski[1], Mariusz Dzwonkowski[1,2], and Roman Rykaczewski[1]

[1] *Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology, Gdańsk, Poland*
[2] *Department of Radiological Informatics and Statistics, Medical University of Gdańsk, Gdańsk, Poland*

**Abstract—In this paper a new idea of digital images fingerprinting is proposed. The method is based on quaternion encryption in the Cipher Block Chaining (CBC) mode. Quaternions are hyper-complex numbers of rank 4 and thus often applied to mechanics in three-dimensional space. 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 quaternions in a three-dimensional space around another quaternion (key). On the receiver's side, a small amount of unnoticeable by human eye errors occurs in the decrypted images. These errors are used as a user's digital fingerprint for the purpose of traitor tracing in case of copyright violation. A computer-based simulation was performed to scrutinize the potential presented quaternion encryption scheme for the implementation of digital fingerprinting. The obtained results are shown at the end of this paper.**

***Keywords—cryptography, multimedia, security, watermarking.***

## 1. Introduction

Nowadays, delivering sensitive digital multimedia contents confidentially over vulnerable public networks is a matter of high importance. The problem received much attention, however the number of possible methods which enable the transmitted data interception, is increasing rapidly.

Quaternion encryption as presented in [1], [2] uses the unique properties of quaternions in order to rotate vectors of data in three-dimensional space. Because of their unique structure, quaternions are commonly used in place of matrices to perform rotations [3]. In a proposed method, the authors treat a data vector rotation 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 [3], [4]. Additionally, when encrypting a color image in RGB representation, it is possible to increase the encryption efficiency even further. In that case a single quaternion can successfully store information about pixel all three colors.

The model proposed by authors in [1], [2] uses the Electronic Code Book (ECB) encryption mode. However, it is well known that altering encryption by including dependencies between corresponding blocks of encrypted data will result in achieving a greater security. Development of the Cipher Block Chaining (CBC) implementation resulted in a algorithm discovery that generates a small amount of unnoticeable errors on the receiver's side during the decryption process. These sets of errors is used as digital fingerprints for the purpose of pirate tracing. In this case, digital fingerprinting should be considered as an encryption technique additional feature.

The purpose of this paper is to present the implementation and quaternion experimental results encryption with a digital fingerprinting. The security issue of the quaternion encryption algorithm has been discussed in [5]–[7]. To the best of authors knowledge, this is the first work which shows a digital fingerprinting for images based on quaternion calculus. It is important to note that the proposed algorithm is an ongoing work. Further studies on the method are necessary and are highlighted in the paper.

## 2. 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 $\mathbb{R}^3$. A quaternion $q$ is defined by [8]:

$$q = w + xi + yj + zk, \tag{1}$$

where $w$, $x$, $y$, $z$ are real coefficients of quaternion $q$, and $i$, $j$, $k$ are imaginary units with the following properties [8]:

$$\begin{aligned} i^2 = j^2 = k^2 = ijk = -1, \\ ij = -ji = k, \\ jk = -kj = i, \\ ki = -ik = j, \end{aligned}$$

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

$$q = \begin{bmatrix} w & x & y & z \end{bmatrix}^T \quad \text{or} \quad q = (w, \vec{v}) = \left( w, \begin{bmatrix} x & y & z \end{bmatrix}^2 \right). \tag{2}$$

The sum of two quaternions $q_1$, $q_2$ is defined by adding the its corresponding coefficients, i.e. in the same manner as for complex numbers [5]:

$$q_1 + q_2 = (w_1 + w_2) + (x_1 + x_2)i + (y_1 + y_2)j + (z_1 + z_2)k. \tag{3}$$

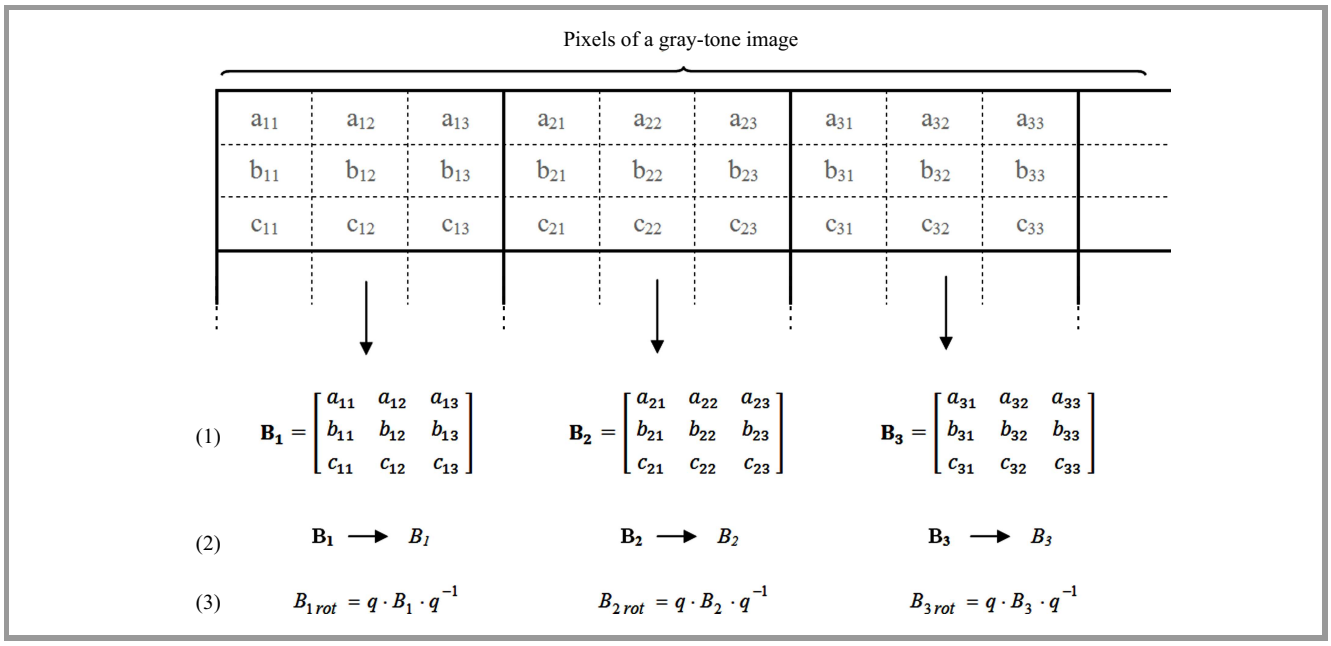



*Fig. 1.* Encryption of a gray-tone image using the quaternion method.

The product of two quaternions is more complex due to its anticommutativity of the imaginary units during the multiplication process. The product of the two quaternions $q_1$, $q_2$ consists scalar and vector products [5]:

$$q_1 \cdot q_2 = (w_1 w_2 - \vec{v}_1 \cdot \vec{v}_2, w_1 \vec{v}_2 + w_2 \vec{v}_1 + \vec{v}_1 \times \vec{v}_2). \quad (4)$$

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

$$q^* = w - xi - yj - zk \quad \|q\| = \sqrt{w^2 + x^2 + y^2 + z^2}, \quad (5)$$

$$q^{-1} = \frac{q^*}{\|q\|^2} = \frac{w - xi - yj - zk}{w^2 + x^2 + y^2 + z^2}. \quad (6)$$

It is important to notice that in the case of a unit quaternion, for which the norm is equal to 1, there is the following relation: $q^{-1} = q^*$.

## 3. Quaternion Encryption

The proposed encryption scheme is based on quaternion rotation by creating a quaternion (key) around will be rotating another quaternion (data). Let's consider two quaternions $q = [w\ x\ y\ z]^T$ 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 rotated data vector $[a\ b\ c]^T$. The quaternion rotation is written as:

$$P_{rot} = q \cdot P \cdot q^{-1}. \quad (7)$$

There are two possible ways of implementing a proposed quaternion encryption. First approach called quaternion method focuses on the Eq. (7) and it is entirely based on quaternion calculus. The alternative matrix method introduces a rotation matrix [1], [2], [5], which enables the implementation of quaternion encryption via a matrix multiplication. The matrix method is easier to implement, however it lacks the fast computing advantages of the quaternion method [3], [4]. In presented algorithm the quaternion method is used.

The quaternion method is entirely based on the quaternion rotation shown in Eq. (7). 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 in Eq. (8) [1], [2].

$$P = \left(0, \begin{bmatrix} a \\ b \\ c \end{bmatrix}\right) \rightarrow B = \left(0, \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix}\right). \quad (8)$$

The encryption and decryption process for the quaternion method with the new extended quaternion B (meant to store data information) is shown in Eqs. (9) and (10) respectively.

$$B_{rot} = q \cdot B \cdot q^{-1}, \quad (9)$$

$$B = q^{-1} \cdot B_{rot} \cdot q, \quad (10)$$

where $B_{rot}$ is the rotated quaternion $B$.

It is also possible to compute a rotation matrix [1], [2], [5]–[7] in order to obtain quaternions 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.

In order to better understand the encryption process via the quaternion method let's consider an example as shown in Fig. 1. Assume that aim is to encrypt a random gray-tone image, represented by pixels which values are in

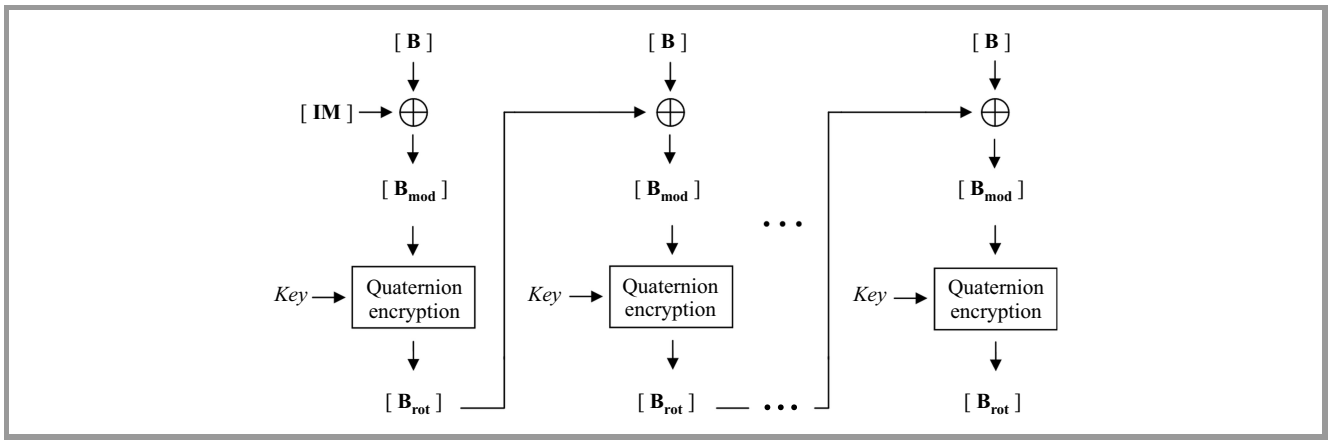



***Fig. 2.*** Example of dependency between matrices in quaternion encryption algorithm for CBC mode.

the 0–255 range. First is to organize those pixels in the appropriate data matrices **B** (step 1), and then convert those matrices to quaternions *B* (step 2) and perform the quaternion rotation according to Eq. (9) (step 3).

## 4. Cipher Block Chaining Mode of Encryption

The model presented in [1], [2] concerns encryption in Electronic Code Book mode, which means that the blocks of data **B** are encrypted and decrypted separately according to Eqs. (9) and (11). 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 previous steps (Fig. 2). 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 rotated according to Eq. (9).

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. This issue is addressed in the next section.

## 5. Error Occurrence

Floating point representation is similar in concept to scientific notation. According to standard IEEE-754, a floating point number consists of a sign bit, exponent bits and significant bits also known as coefficient or mantissa bits. It is important to notice that during the bitwise binary addition of elements **B** and $\mathbf{B_{rot}}$ matrices, there is not possibility to allow the addition of all corresponding bits with modulo 2 arithmetic (XOR operation). This could lead to a situation where a special number type could be obtained [9]. It is important to implement exceptions for all such numbers. Moreover, there is necessity to support the appropriate range of precision for floating point numbers. According to a paper [9], the range for single precision floating point number is $1.2 \cdot 10^{-38} \ldots 3.4 \cdot 10^{38}$.

The main problem occurs during the decryption process. While performing quaternion calculations on the receiver's side it is impossible to achieve exactly the same value of a floating point number as it is on the transmitter's side. This could lead to a situation where errors could be encountered in the decrypted data. Nevertheless, it is possible to minimize the number of errors. For example, in case of gray-tone images, it is possible to keep the errors at an unnoticeable by human eye level of a maximum of $\pm 3$ pixel value even on image regions with uniform color (gray tone).

In order to better understand error behavior on the receiver's side, an example as shown in Fig. 3 is considered. The authors use an initialization matrix IM in the first step of the CBC mode of encryption. Figure 3 presents only one pair of subsequent elements from data matrix B and initialization matrix **IM**. In order to perform the XOR operation on those elements to convert them to their binary floating point representation according to standard IEEE-754 is proceed at first. After the XOR operation a floating point number (a value from matrix $\mathbf{B_{mod}}$) is obtained which is then encrypted with proposed quaternion algorithm and decrypted on the receiver's side. However, due to the quaternion calculations a slightly different floating point number is obtained. If a rounding to the fourth digit after the decimal point is implemented, the ability to control the error size at an acceptable level of a maximum of $\pm 3$ on the receiver's side is achieved (Fig. 3).

Rounding to any further digit after the decimal point could lead to a situation where the obtained floating point number would be wrongly represented in chosen variable sec-

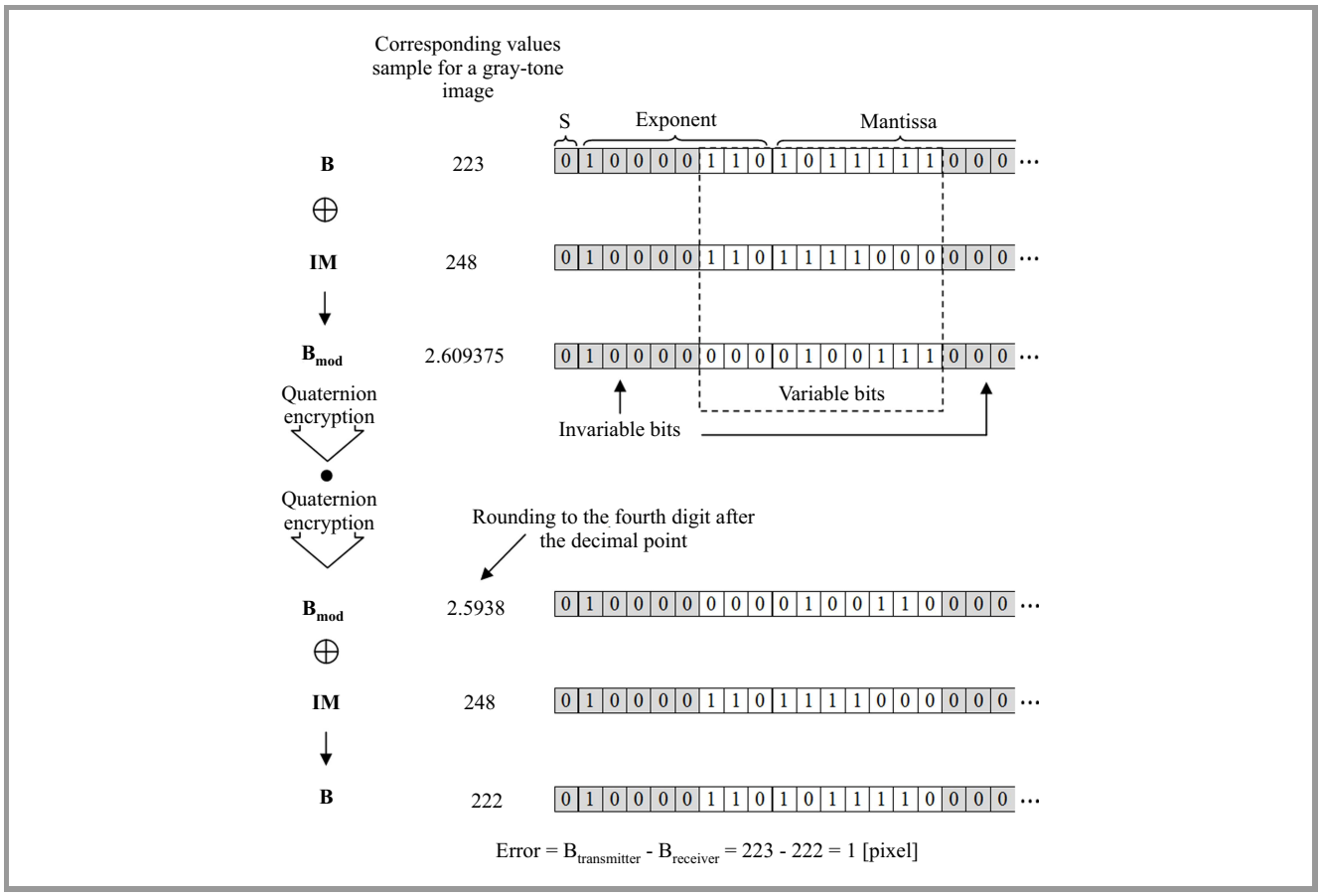



***Fig. 3.*** Error size shown for some corresponding values in terms of the gray-tone image data.

tion of the binary floating point number representation. It would thus generate a significant error. When performing the XOR operation with the element from **IM** on the receiver's side, an element of decrypted data matrix $\mathbf{B_{rot}}$ is obtained which is identical to or different from (maximum $\pm 3$ in terms of the gray-tone image data) the original element of data matrix **B** on the transmitter's side.

Up to this point the reader will probably have started wondering why the binary floating point representation was divided into two sections: constant and variable. According to the beginning of this section, the authors cannot allow the addition of all corresponding bits of matrices **B** and **IM** elements with modulo 2 arithmetic. During the XOR operation a variable part of the binary floating point representation is defined for which such an operation is performed – it contains 3 bits of exponent and 7 bits of mantissa (Fig. 3). The other bits are considered invariable and have constant values. Such an approach (the amount of variable bits) is necessary in order to represent the largest value allowed for a gray-tone image: 255. It will also eliminate the threat of generating special values, and it will solve the precision problem for floating point numbers. Performing the XOR operation on the allowed variable bits will result in producing values in the range of 2–510. The authors will not be able to obtain values less than 2 due to the used constant bit values for each binary floating point representation. Therefore, pixels are treated with a value of 0, and 1 of a gray-tone image, as an equivalent of a pixel value of 2. Thus the produced error is in the allowed range of $\pm 3$.

## 6. Fingerprinting as an Additional Feature

There are two complementary methods for multimedia content and copyright protection. The first method is an encryption, which 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 do 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 [10]–[19], 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. It is important that the fingerprints must be embedded in the multimedia content in such way that they remain invisible to the human eye, or at least do not bother the user. An analysis of the embedded data in a pirate copy aims to identify by whom the copy was illegally re-

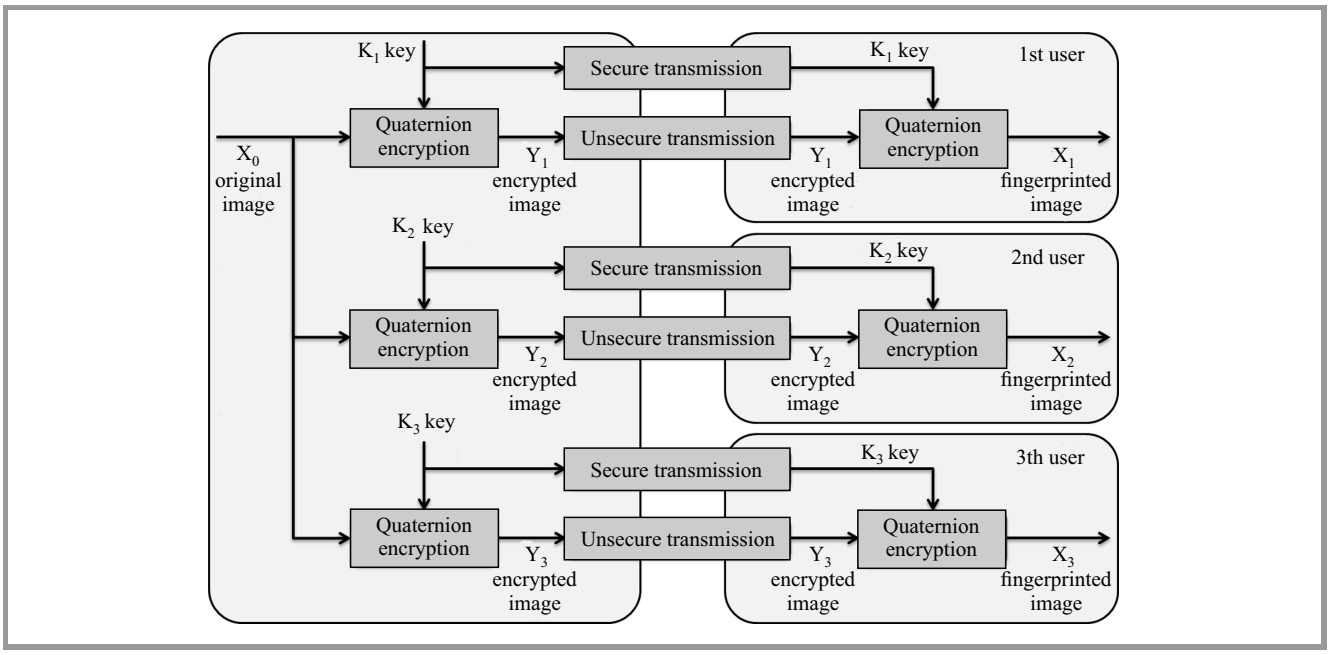



***Fig. 4.*** General scheme of the multimedia distribution system based on a quaternion encryption.

leased. The presence of the fingerprint in the examined illegal copy is the basis for the rogue users identification, which consequently leads to the prosecution and conviction of pirates.

It should be assumed that pirates are aware of the presence of fingerprints in their copies and that they will carry out various attacks intended to remove the fingerprints. The greatest threat to mass multimedia distribution are organized pirates groups who analyze the fingerprinted copies available to them and on this basis produce a pirated copy which is fingerprints free or contains a damaged fingerprint which does not identify the real pirates. Such attacks are called collusion attacks [10], [20]–[22].

The quaternion encryption scheme presented in this paper 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 [10]. 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.

Detection thresholds are not selected analytically, but experimentally. The threshold should be specified in order to meet specific requirements for the application of the fingerprinting system. There are three possible scenarios that impact the choice of threshold [21]. In the "catch one" scenario the main goal is error-free identification of at least one pirate without accusing any innocent user. This criterion is especially important for collection of evidence. In this scenario the detection threshold is usually very high. In the "catch many" scenario, accusing several innocent users is acceptable for the benefit of detecting more pirates. Two groups of suspects are thus obtained: a correctly identified pirates and a wrongly accused users. In this scenario the detection threshold is much lower than in the previous one. In the "catch all" scenario the aim is to detect all of the pirates with an acceptable number of wrongly accused users. This criterion applies in cases where it is necessary to detect all users involved in the crime at all costs. In this scenario the selected detection threshold is the lowest.

## 7. Simulation Results

In each of the performed simulations 100 users ordered the same images, 243 by 243 pixels in size and in an 8-bit grayscale. Six different images were used for the simulations and this process was repeated 5 times for each image.

This gave 30 performed simulations. In order to test the fingerprints' robustness the authors simulated collusion attacks in which 5, 10, 15, 20, 25, 30 pirates were involved. The damaged fingerprint was extracted from a pirated copy, then the correlation coefficients with the fingerprints of all users were calculated and compared to fixed detection thresholds: 0.12, 0.13, 0.14.

A simulated attack was a linear collusion where the fingerprinted copies that were available to pirates were averaged. All fingerprinted copies were averaged with equal weight, so the energy of each pirate's fingerprint was reduced by the same factor, which is the inverse of the number of pirates. This means that the risk of being detected is evenly distributed among the pirates, so it is a fair collusion [20]. More importantly, this attack does not degrade the visual multimedia content quality. Therefore, the linear collusion attack by averaging the fingerprinted copies, is the most probable attack [20].

Figure 5 presents the original image, the encrypted image based on presented CBC quaternion third order encryption algorithm and the decrypted image that is automatically embedded with a random pattern fingerprint. It should be noted that embedded fingerprint invisibility has been reached. The Peak Signal to Noise Ratio (PSNR) for the fingerprinted image is +58.64 dB. In this case the original image is considered as a useful signal and the fingerprint is considered as noise.

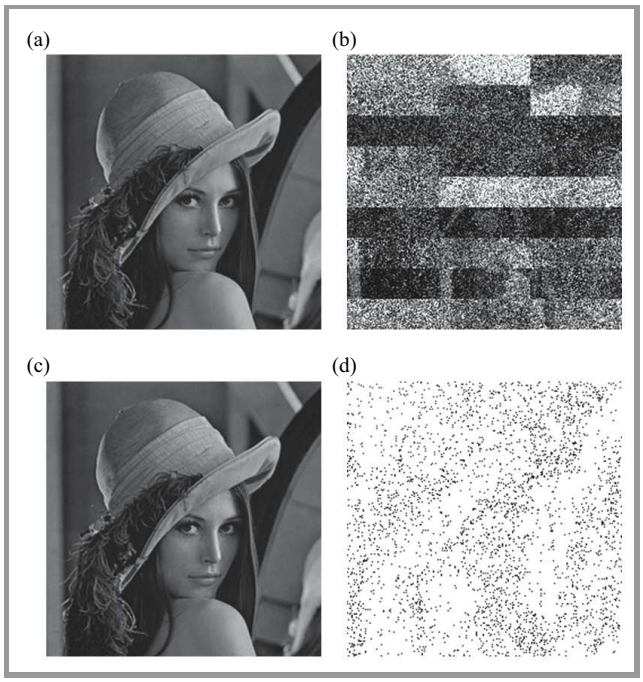



***Fig. 5.*** Example of joint quaternion encryption and fingerprinting: (a) original image, (b) encrypted image, (c) decrypted and fingerprinted image (PSNR = +58.64 dB), (d) fingerprint shown as a pattern of errors.

The correlation analysis of the embedded fingerprints has been performed. The analysis was run for 6 images ordered by 100 users. The auto-correlation coefficient of the fixed user's fingerprint was always equal to 1. The cross-correlation coefficients with the remaining 99 fingerprints was always less than 0.1. Such properties are possible because each fingerprint is very long, as it covers the entire image.

Figure 6 shows the correlation coefficients between the fingerprint from the pirate copy and the fingerprints of all the users in case of 15 pirates involved in the attack. The pirates who performed a collusion attack are users with the lowest ID numbers. The correlation coefficient values for the pirates decrease accordingly to the expanding number of attackers. However, the correlation coefficients of every guilty user are still higher than for any innocent user. It is possible to set an appropriate detection threshold in order to achieve a differentiation between pirates (colluders) and honest users.

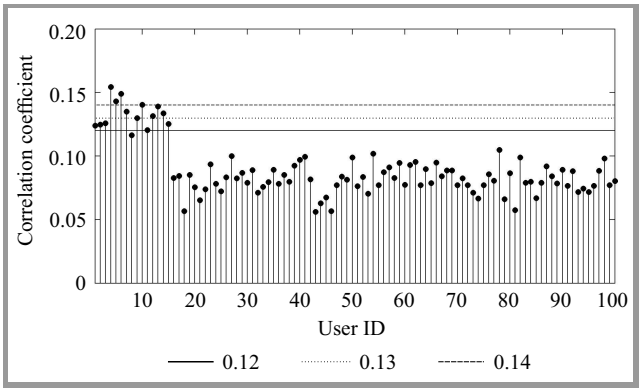



***Fig. 6.*** Correlation coefficients between a fingerprint from a pirate copy and individual users fingerprints after a collusion attack of 15 pirates.

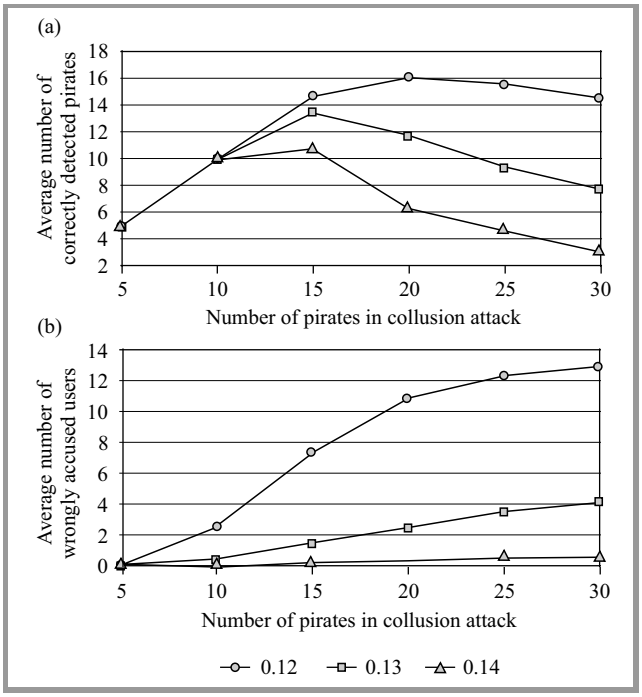



***Fig. 7.*** Results of the robustness testing: (a) the number of correctly detected pirates, (b) the number of wrongly accused users.

Figure 7 show the results of the robustness testing. Each chart shows the average results of 30 simulations. The confidence intervals for the mean values were calculated for a 95% confidence level using a model based on Student's $t$-distribution. The average value $\mu$ and standard deviation $\sigma$ for the whole population is unknown. In order to calculate the confidence intervals for the mean $\mu$ of the population the following formula was used [23]:

$$P\left(m-\frac{t\cdot\sigma}{\sqrt{n-1}}<\mu<m+\frac{t\cdot\sigma}{\sqrt{n-1}}\right)=1-\alpha, \qquad (11)$$

where $m$ is the sample calculated mean value, $\sigma$ is the sample standard deviation, $n$ stands for the sample size, $t$ is the value from the Student-$t$ distribution for $n-1$ degrees of freedom and probability $1-\alpha/2$, and $1-\alpha$ is the confidence factor, which is 0.95. The results are shown in Tables 1 and 2.

Table 1
Confidence intervals for the mean values of correctly detected pirates

| Number of pirates | Threshold 0.12 | | Threshold 0.13 | | Threshold 0.14 | |
|---|---|---|---|---|---|---|
| | $m$ | $\frac{t\cdot\sigma}{\sqrt{n-1}}$ | $m$ | $\frac{t\cdot\sigma}{\sqrt{n-1}}$ | $m$ | $\frac{t\cdot\sigma}{\sqrt{n-1}}$ |
| 5 | 5.00 | 0.00 | 5.00 | 0.00 | 5.00 | 0.00 |
| 10 | 10.00 | 0.00 | 10.00 | 0.00 | 10.00 | 0.13 |
| 15 | 14.67 | 0.23 | 13.43 | 0.86 | 10.77 | 1.47 |
| 20 | 16.03 | 1.45 | 11.73 | 2.11 | 6.40 | 2.02 |
| 25 | 15.50 | 2.71 | 9.37 | 3.04 | 4.63 | 1.88 |
| 30 | 14.53 | 3.90 | 7.80 | 3.02 | 3.17 | 1.51 |

Table 2
Confidence intervals for the mean values of wrongly accused users

| Number of pirates | Threshold 0.12 | | Threshold 0.13 | | Threshold 0.14 | |
|---|---|---|---|---|---|---|
| | $m$ | $\frac{t\cdot\sigma}{\sqrt{n-1}}$ | $m$ | $\frac{t\cdot\sigma}{\sqrt{n-1}}$ | $m$ | $\frac{t\cdot\sigma}{\sqrt{n-1}}$ |
| 5 | 0.10 | 0.12 | 0.00 | 0.00 | 0.00 | 0.00 |
| 10 | 2.50 | 1.68 | 0.37 | 0.40 | 0.00 | 0.00 |
| 15 | 7.37 | 3.74 | 1.57 | 0.99 | 0.17 | 0.22 |
| 20 | 10.87 | 5.45 | 2.53 | 1.34 | 0.40 | 0.32 |
| 25 | 12.37 | 6.12 | 3.60 | 1.74 | 0.60 | 0.40 |
| 30 | 12.84 | 6.18 | 4.10 | 1.90 | 0.60 | 0.35 |

Detection of 100% of the colluders was achieved for every threshold for 10 or fewer pirates (Fig. 7a, Table 1). For a threshold of 0.12 a detection rate about 100% of rogue users was achieved even for 15 colluders (Fig. 7a). Moreover, even for a group of 20 pirates it is still possible to detect c.a. 80% of them. However, there is a very high number of suspected innocent users (Fig. 7b). Such low threshold should be implemented only for high importance data and when of all rogue users detection is absolutely necessary. In terms of discussed three scenarios one can notice that a "catch all" scenario can be achieved by using this or even a lower threshold [21]. For a 0.13 threshold, detection of c.a. 90% of pirates was achieved for 15 pirates in the attack (Fig. 7a). Yet it is still possible to detect even 50% of them for 20 pirates. More importantly, this threshold implies a very small number wrongly suspected fair users (Fig. 7b), which is, considering the confidence intervals, lower than 6 (Table 2). This threshold ensures detection of a pirates substantial part while reducing the number of wrongly accused users, which means that the "catch many" scenario can be achieved. For a 0.14 threshold, pirate detection is significantly degraded (Fig. 7a). However, there is not a single wrongly accused user for 10 or fewer pirates (Fig. 7b). Moreover, even for up to 30 colluded pirates the number of wrongly suspected users is not greater than 1 (Table 2). Such a high threshold should be implemented when the priority is to achieve reliable detection and strong evidence for the prosecution. A "catch one" scenario can be achieved by using this or an even higher threshold.

# 8. Conclusion

The aim of this paper was to demonstrate the potential error patterns application which occurred after the decryption process of presented quaternion encryption algorithm. This application is digital fingerprinting for images. According to the presented simulation results it is possible to achieve satisfactory results in detecting pirates while maintaining a limited number of wrongly accused users. Being able to conduct pirate tracing during a collusion attack while maintaining confidentiality through encryption of the transmitted data would allow to successfully extend the security boundaries that are offered by multimedia distribution systems.

Additionally, it is important to note that when using the quaternion encryption algorithm the number of possible encryption keys is particularly big. This is due to the fact that both the rotation order and the 4 initialization quaternion parameters can be changed [1], [2], [5]–[7]. Thus 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. 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.

In this paper a scheme for digital fingerprinting as an additional feature of the quaternion encryption scheme is presented. Nevertheless, most of the image communications are lossy in nature, so fingerprinting should withstand such signal processing like compression or filtering. A solution to this problem is fingerprint embedding in a discrete cosine transform or wavelet transform domain, and that is the purpose of proposed further studies.

# References


[1] T. Nagase, M. Komata, and T. Araki, "Secure signals transmission based on quaternion encryption scheme", in *Proc. 18th In. Conf. Adv. Inform. Netw. Appl. AINA 2004*, Fukuoka, Japan, 2004, vol. 2, pp. 35–38.

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

[3] R. Goldman, "Understanding quaternions", *Graphical Models*, vol. 73, no. 2, pp. 21–49, 2011.

[4] R. Goldman, *An Integrated Introduction to Computer Graphics and Geometric Modeling.* New York: CRC Press, 2009.

[5] M. Dzwonkowski, "Software implementation and research of quaternion cryptosystem", Master thesis, Faculty of ETI, Gdańsk University of Technology, Gdańsk, 2011, pp. 5–84.

[6] M. Dzwonkowski and R. Rykaczewski, "A New Quaternion Encryption Scheme for Image Transmission", in *Proc. ICT Young 2012 Conference*, Gdańsk, Poland, 2012, pp. 21–27.

[7] M. Dzwonkowski and R. Rykaczewski, "Quaternion encryption method for image and video transmission", *Telecom. Overv. + Telecom. News*, vol. 8–9, pp. 1216–1220, 2013.

[8] F. Zhang, "Quaternion and matrices of quaternions", *Linear Algebra and its Applications*, vol. 251, pp. 21–57, 1997.

[9] W. Kahan, "IEEE Standard 754 for Binary Floating-Point Arithmetic", pp. 1–30, 1997.

[10] K. J. R. Liu, W. Trappe, Z. J. Wang, M. Wu, and H. Zhao, *Multimedia Fingerprinting Forensics For Traitor Tracing.* EURASIP Book Ser. on Signal Process. and Commun., Hindawi Publishing Corporation, vol. 4, 2005.

[11] K. J. R. Liu and H. Zhao, "Bandwidth efficient fingerprint multicast for video streaming", in *Proc. IEEE Int. Conf. Acoust. Speech and Sig. Process. ICASSP '04*, Montreal, Canada, 2004, vol. 5, pp. 849–852.

[12] D. Kundur and K. Karthik, "Video fingerprinting and encryption principles for digital rights management", *Proc. IEEE*, vol. 92, no. 6, pp. 918–932, 2004.

[13] M. Ammar and P. Judge, "WHIM: Watermarking multicast video with a hierarchy of intermediaries", in *Proc. 10th Int. Worksh. Netw. Operat. Sys. Supp. Digi. Audio Video NOSSDAV 2000*, Chapel Hill, USA, 2000.

[14] R. Parviainen and R. Parnes, "Large scale distributed watermarking of multicast media through encryption", in *Proc. IFIP Int. Conf. Commun. Multimed. Secur. Issues of the New Century*, Darmstadt, Germany, 2001, p. 17.

[15] I.J . Cox, J. Kilian, F. T. Leighton, and T. G. Shamoon, "Secure spread spectrum watermarking for multimedia", *IEEE Trans. Image Process.*, vol. 6, no. 12, pp. 1673–1687, 1997.

[16] R. Anderson and C. Manifavas, "Chameleon – A new kind of stream cipher", *Lecture Notes in Computer Science*, Fast Softw. Encryption, E. Biham. Ed, Heidelberg: Springer, 1997, pp. 107–113.

[17] R. Rykaczewski, "Hillcast – A method of joint decryption and fingerprinting for multicast distribution of multimedia data". *Scientific J. WETI PG*, vol. 19, pp. 231–236, 2010

[18] B. Czaplewski and R. Rykaczewski, "Improvement of fingerprinting method based on Hill Cipher by using frequency domain", in *Proc. ICT Young 2012 Conf.*, Gdańsk, Poland, 2012.

[19] B. Czaplewski and K. Czaplewski, "Protection of visual data transmission for vessel traffic systems using joint fingerprinting and decryption method based on modified Hill cipher", *Annual of Navigation*, vol. 19, no. 2, pp. 5–17, 2012.

[20] K. J. R. Liu, Z. J. Wang, M. Wu, and H. Zhao, "Forensic analysis of nonlinear collusion attacks for multimedia fingerprinting", *IEEE Trans. Image Process.*, vol. 14, no. 5, pp. 646–661, 2005.

[21] K. J. R. Liu, W. Trappe, Z. J. Wang, and M. Wu, "Collusion-resistant fingerprinting for multimedia", *IEEE Sig, Process. Mag.*, vol. 21, pp. 15–27, 2004.

[22] S. He and M. Wu, "Joint coding and embedding techniques for multimedia fingerprinting", *IEEE Trans. Inf. Forensics and Secur.*, vol. 1, no. 2, pp. 231–247, 2006.

[23] W. Krysicki, J. Bartos, W. Dyczka, K. Królikowska, and M. Wasilewska, *Probability and Mathematical Statistics in Exercises*, part II, IV ed., Warszawa: PWN, 1999, pp. 49–54.

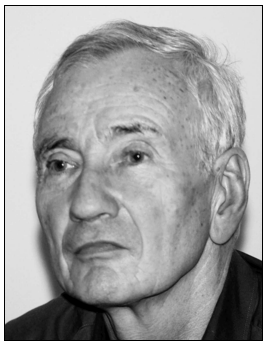


**Bartosz Czaplewski** completed studies in the field of Electronics and Telecommunications majoring in Teleinformation Systems at Gdańsk University of Technology in 2011 received his M.Sc. Eng. Currently, he continues his academic career as a Ph.D. student in the field of Telecommunication and he is working as a lecturer at Gdańsk University of Technology in Dept. of Teleinformation Networks. His current research interests include digital fingerprinting, digital watermarking, cryptography, steganography, network security.
E-mail: czapla@eti.pg.gda.pl
Faculty of Electronics, Telecommunications
and Informatics
Department of Teleinformation Networks
Gdańsk University of Technology
Gabriela Narutowicza st 11/12
80-233 Gdańsk, Poland



**Mariusz Dzwonkowski** received his Master's degree in Telecommunication from Dept. of Teleinformation Networks, Gdańsk University of Technology. He is now working toward his Ph.D. degree in Telecommunication from Gdańsk University of Technology. He is currently a lecturer at Medical University of Gdańsk, Poland in Dept. of Radiological Informatics and Statistics. His research interests include steganography, cryptography with emphasis on quaternion encryption, network security and image processing.
E-mail: mar.dzwonkowski@gmail.com
Faculty of Electronics, Telecommunications
and Informatics
Department of Teleinformation Networks
Gdańsk University of Technology
Gabriela Narutowicza st 11/12
80-233 Gdańsk, Poland

Department of Radiological Informatics and Statistics
Medical University of Gdańsk
Tuwima st 15
80-210 Gdańsk, Poland



**Roman Rykaczewski** received his M.Sc. and Ph.D. degrees in 1968 and 1975, respectively, from Gdańsk University of Technology – Faculty of Electronics, Telecommunications and Informatics. From 1968 to the present he is working as an academic teacher at Gdańsk University of Technology, Faculty of Electronics, Telecommunications and Informatics. His current research mainly focuses on cryptography, watermarking and steganography.

E-mail: romryk@eti.pg.gda.pl
Faculty of Electronics, Telecommunications
and Informatics
Department of Teleinformation Networks
Gdańsk University of Technology
Gabriela Narutowicza st 11/12
80-233 Gdańsk, Poland