Abstract: In this paper a new receiver-side fingerprinting method for color images is proposed. The proposed method belongs to the group of Joint Fingerprinting and Decryption (JFD) methods. Each component of the color image is represented as a point in 3D space, which is formed by 3 color channels. Encryption and fingerprinting can be interpreted as rotation, scaling and translation of these points. Calculations are performed using quaternion algebra. The experimental results show that the proposed method is robust against collusion attacks.

1. INTRODUCTION

Nowadays, the interest in the mass multimedia distribution services is persistently growing. Currently we are using video and images for many purposes, starting from business and ending with entertainment. However, unauthorised redistribution of multimedia - the piracy, is a real threat and it can cause the financial and moral losses to the authors of the content.

The way to trace the pirates is by using a digital fingerprint method [1]. Digital fingerprinting is a data hiding technique in which data is protected by unique sequences ('fingerprints') embedded in the image. Each copy of image consists a hidden fingerprint which is invisible to the human eye and identifies a particular user. If the malicious user redistributes his or her copy, the analysis of the embedded fingerprint should allow to identify the pirate. It should be assumed that pirates will perform attacks intended to remove fingerprints from their copies in order to safely distribute a pirate copy. Thus, embedded fingerprints have to be robust against the removal performed by a single pirate or a group of pirates, i.e. so called collusion attacks.

Fingerprinting methods can be divided into three groups according to the place where the fingerprints can be embedded. Fingerprints can be embedded on the transmitted side and then copies are encrypted and sent to the users via multiple unicast transmissions [2,3]. This is a transmitter-side fingerprinting. Fingerprints can be embedded as the data travels through the network by using special network devices which embed their own watermarks in the transmitted data [4]. This is a in-network fingerprinting. Fingerprints can be embedded on the receiver side after or during the decryption. The transmitter sends only one copy via multicast transmission. This is a receiver-side fingerprinting. The most promising receiver-side fingerprinting methods belong to the group named Joint Fingerprinting and Decryption (JFD) [5,6,7,8,9,10,11,12].

This paper presents a new JFD method for color images. In the proposed method three color channels form a 3D space and each component of the image is a point in this 3D space. The distribution side uses a symmetric cipher to encrypt an image (or rather perceptually essential components of the image) with the encryption key, which is common for all the users, and then sends the encrypted data via multicast transmission to all users. An encryption involves rotation, scaling and translation of points in 3D color space. Calculations are performed using quaternions which are 4-dimensional complex numbers, and they are very convenient for 3D rotations. Each user has a unique decryption key, which is different from the encryption key. The differences between the common encryption key and the individual decryption key cause that the image decrypted by a user contains minor changes which are imperceptible to the human eye and which are unique across all users. Therefore, these changes are the user's fingerprint.

The structures of the paper is as follows. Related work significant to the development of the proposed method is listed in Section 2. Section 3 is a brief introduction to the quaternion algebra. Section 4 describes the proposed JFD method based on quaternion sandwich product. Research results on robustness against collusion attacks are presented in Section 5. The conclusions are discussed in Section 6.

2. RELATED WORK

First time, the concept of fingerprinting combined with decryption process was introduced in the Chameleon [6] by Anderson and Manifavas. The Chameleon is a stream cipher for audio data in which fingerprints are embedded in the least significant bits of protected data. This method is not robust against large collusions of pirates.

Later, Kundur and Karthnik’s method was introduced by the name of Joint Fingerprinting and Decryption [5]. In this method, an encryption is based on scrambling signs of discrete cosine transform (DCT) coefficients of the image. A decryption unscrambles only part of DCT coefficients and the remaining ones remain encrypted and their combination is the user’s fingerprint. Since this method was the inspiration for so many other
scientists, the name is now used to describe whole group of methods which are based on the same idea of having one encryption key and multiple decryption keys which introduce fingerprints into the data.

Another important JFD method is the Fingercasting scheme designed by Adelsbach et al. [7] and later improved by Katzenbeisser et al. [8]. This is a generalization of the Chameleon cipher which embeds spread-spectrum watermarks into audio-visual content. The main difference is that the lookup-table entries are uniformly distributed random elements and that the XOR operation is replaced by a modular addition.

Next JFD method significant for the development of the proposed method is the Hillcast [9,10], which is a simple block cipher based on matrix multiplication, i.e., the generalized Hill cipher. An image is encrypted with a group key and sent to all the users via multicast transmission. A decryption process with the unique decryption key embeds a fingerprint in the image. The Improved Hillcast [11,12] involves fingerprint embedding in the frequency domain by using the DCT coefficients instead of embedding in the spatial domain for greater robustness against signal processing.

Another related work is a fingerprinting feature in a CBC quaternion encryption [13,14,15]. In this method, an encryption of an image is based on quaternion rotation around quaternion key in spatial domain, i.e., calculations are done on pixel values. The fingerprinting feature is a side effect caused by patterns of errors which occurred in the encrypted image. This is a receiver-side fingerprinting although this is not a JFD method. Unfortunately, this fingerprinting method scales very poorly as the required bandwidth grows linearly with the number of users.

In recent years, many JFD methods have been proposed, but to the best of our knowledge, the method proposed in this paper is the first JFD method based on quaternion algebra.

3. QUATERNIONS

Quaternions are an extension of complex numbers [16,17]. They have 4 dimensions: 1 real dimension and 3 imaginary dimensions. A quaternion \( q \) can be represented as follows:

\[
q = a + bi + cj + dk,
\]

where \( a, b, c, \) and \( d \) are scalar numbers (components) and \( i, j, \) and \( k \) are imaginary unit values.

The sum of two quaternions \( q_1, q_2 \) is defined as follows:

\[
q_1 + q_2 = (a_1 + a_2) + i(b_1 + b_2) + j(c_1 + c_2) + k(d_1 + d_2).
\]

The multiplication of two quaternions \( q_1, q_2 \) can be described as follows: we put each quaternion in brackets and multiply out all the terms:

\[
q_1 \cdot q_2 = \begin{array}{l}
(a_1 + b_1i + c_1j + d_1k) \cdot (a_2 + b_2i + c_2j + d_2k) = \\
= (a_1a_2 - b_1b_2 - c_1c_2 - d_1d_2) + \\
+ (a_1b_2 + a_2b_1 + c_1d_2 - c_2d_1)i + \\
+ (a_1c_2 - a_2c_1 + b_1d_2 + b_2d_1)j + \\
+ (a_1d_2 - a_2d_1 + b_1c_2 - b_2c_1)k 
\end{array}
\]

where the rules of multiplication of imaginary units are as follows: \( i \cdot i = j \cdot j = k \cdot k = -1, \quad i \cdot j = k, \quad j \cdot i = -k, \quad j \cdot k = i, \quad k \cdot j = -i, \quad k \cdot i = j, \) and \( i \cdot k = -j \). As follows from (3) the product of quaternions is non-commutative.

The conjugate of a quaternion is a quaternion with the changed signs of the imaginary components:

\[
q^* = a - bi - cj - dk.
\]

The norm of a quaternion is:

\[
\|q\| = \sqrt{a^2 + b^2 + c^2 + d^2}.
\]

To normalise a quaternion we divide each quaternion component \( a, b, c, \) and \( d \) by the norm. The norm of a normalised quaternion is equal to 1.

The inverse of a quaternion is:

\[
q^{-1} = \frac{q^*}{\|q\|^2} = \frac{a - ib -jc -kd}{a^2 + b^2 + c^2 + d^2}.
\]

Quaternions can represent many 3D transformations, such as reflections, scaling, translations and more. However, the main profit from the use of quaternions appears in 3D rotations. Rotations in 3D space using 3 scalar values are nonlinear, they have singularities and they are difficult to combine. To solve these problems we can model 3D rotations in a 4D space using quaternions [16,17].

In formulas (7), (8), and (9), \( q \) is a quaternion representing transformation (in (7) \( q \) has to be normalised), \( P_{\text{in}} \) is a vector (a quaternion with real part equal to 0) representing point before being transformed, and \( P_{\text{out}} \) is a vector (a quaternion with real part equal to 0) representing point after being transformed.

For pure rotation of a point in 3D space we can use the sandwich product:

\[
P_{\text{out}} = q \cdot P_{\text{in}} \cdot q^*.
\]

For rotation with scaling of a point in 3D space we can use the sandwich product:

\[
P_{\text{out}} = q \cdot P_{\text{in}} \cdot q^{-1}.
\]

For translation of a point in 3D space we can use the addition:

\[
P_{\text{out}} = q + P_{\text{in}}.
\]
4. PROPOSED METHOD

In the proposed method, there is one encryption key for all users and one decryption key per user. The image is encrypted once and sent to all the users via multicast transmission. Each user decrypts the received data and at the same time embeds his or her fingerprint into the image. The embedded fingerprint is based on the difference between the encryption key and the user’s decryption key.

The encryption key is constructed as follows:

\[ e = [e_1, e_2, \ldots, e_N]. \]  (10)

where \( e \) is the encryption key, \( N \) is the length of the encryption key, \( e_n \) is a quaternion with random scalar components, e.g. in our implementation components were uniformly distributed numbers from the set \{1; 2; 3; 4\} and \( n = 1,2,\ldots,N \).

The fingerprint for \( h \)-th user is constructed as follows:

\[ f^{(h)} = [f_1^{(h)}, f_2^{(h)}, \ldots, f_N^{(h)}]. \]  (11)

where \( f^{(h)} \) is the \( h \)-th user fingerprint, \( f_n^{(h)} \) is a quaternion with random scalar components of values \{-1; +1\}, e.g. in our implementation fingerprints were taken from bipolar Gold codes \[18\] and \( n = 1,2,\ldots,N \).

The decryption key for \( h \)-th user is constructed as follows:

\[ d^{(h)} = e + \alpha \cdot f^{(h)}, \]  (12)

where \( d^{(h)} \) is the \( h \)-th decryption key, and \( \alpha \) is the embedding strength which determines the balance between the imperceptibility and the robustness of embedded fingerprints. Note that the fingerprint \( f^{(h)} \) will not be the actual fingerprint embedded in the image but in fact it will be its origin.

Before the encryption, the image has to be processed in a way similar to transformations known from the MPEG and JPEG standard. Firstly, a color image in RGB color representation is split into three sub-images, one per each color channel. Secondly, each sub-image is divided into 8 by 8 pixels blocks. Thirdly, the discrete cosine transform (DCT) is performed for each block independently. This operation converts 64 pixel values of each block into 64 DCT coefficients.

The fourth step is to select the most perceptually essential coefficients for an encryption. Encrypting all coefficients will effectively conceal the image. However, it is highly recommended to encrypt only the perceptually essential components of the image because this lowers computation and delay and is enough to prevent image estimation with good quality \[7,19\]. The perceptually essential coefficients are clustered in the upper left corner of each block. However, the DC coefficient (in 1st row and 1st column of each block) should not be selected because it could significantly degrade the image quality. In order to spread a fingerprint over the entire image, the coefficients should be selected from as many blocks as possible. For the purpose of this paper, let assume for simplicity that only one coefficient of each block is selected for an encryption. As a result, three vectors \( r, g, b \) are formed from coefficients of each color channel:

\[ r = [r_1, r_2, \ldots, r_M], \]  (13)
\[ g = [g_1, g_2, \ldots, g_M], \]  (14)
\[ b = [b_1, b_2, \ldots, b_M], \]  (15)

where \( r_m, g_m, b_m \) are selected coefficients from red, green and blue channel respectively, \( m = 1,2,\ldots,M \), and \( M \) is the number of selected coefficients in each color channel.

The fifth step is to form a vector of quaternions which will be encrypted:

\[ x = [x_1, x_2, \ldots, x_M], \]  (16)
\[ x_m = 0 + r_m j + g_m j + b_m k, \]  (17)

where each quaternion \( x_m \) in vector \( x \) represents a point in 3D space of color channels, and \( m = 1,2,\ldots,M \).

Selected coefficients are encrypted by using the encryption key according to the formula:

\[ y_m = e_l \cdot [(x_m + s_1 + s_2) \mod N_2] \cdot e_l^{-1}. \]  (18)
\[ l = [(m-1) \mod N] + 1. \]  (19)
\[ y = [y_1, y_2, \ldots, y_M], \]  (20)

where \( y_m \) is an encrypted quaternion, \( x_m \) is a quaternion formed from selected DCT coefficients, \( e_l \) is a quaternion from the encryption key \( e, m = 1,2,\ldots,M \) is the number of selected coefficients in each color channel, \( N \) is the length of the encryption key, \( y_0 \) is an initialization quaternion which is a fixed quaternion with a real part equal to 0 and non-negative imaginary parts, \( s_1 \) and \( s_2 \) are constants designed to shift values of the all DCT coefficient to positive values and make modular addition possible. For example, in our implementation \( y_0 = 0 + 10i + 10j + 10k, s_1 = 0 + 1000i + 1000j + 1000k, \) and \( s_2 = 2000 \).

Afterwards, the encrypted coefficients have to be extracted from quaternions \( y_n \) and put back in the same places in DCT blocks from which they were previously selected. All blocks are quantized for a fixed level of quality, set in a zigzag order, converted to a binary form and RLE/DPCM/Huffman-coded according to the JPEG standard. The encrypted image is sent to all users via one multicast transmission.

In order to decrypt the data, a receiver has to reverse the above-mentioned coding operations, and then extract the encrypted coefficients from the DCT blocks and recreate the \( y \) vector, as in formulas from (13) to (17). The joint fingerprinting and decryption for the \( h \)-th user is performed according to the formula:
\[
x_m^{(h)} = [(d_l^{(h)})^{-1} \cdot y_m \cdot d_l^{(h)} - y_{m-1}] \mod s_2 \] - s_1, \quad (21)
\]
\[
l = [(m-1) \mod N] + 1, \quad (22)
\]
\[
x^{(h)} = \left[ x_1^{(h)} \ x_2^{(h)} \ldots \ x_N^{(h)} \right], \quad (23)
\]
where \( x^{(h)} \) is the vector of the decrypted quaternions \( x_m^{(h)} \) with the embedded fingerprint of the \( h \)-th user, \( d_l^{(h)} \) is a quaternion from the decryption key \( d^{(h)} \) of the \( h \)-th user, and \( y_0 \) is an initialization quaternion. The difference between \( e \) and \( d^{(h)} \), shown in the formula (12), introduces fingerprint into the decrypted image during the decryption process shown in the formula (21).

As it can be seen, the proposed JFD method operates in the cipher-quaternion chaining mode of encryption, similar to the cipher-block chaining mode. This is visualized in Figure 1 for better understanding.

\[ f_{emb}^{(h)} = (x^{(h)} - x)/\alpha, \quad (24) \]
where \( x \) is the vector of quaternions selected for encryption from the original image, \( x^{(h)} \) is the vector of corresponding quaternions in the fingerprinted image of the \( h \)-th user, \( \alpha \) is the fingerprint embedding strength, and \( f_{emb}^{(h)} \) is the embedded fingerprint in the \( h \)-th user’s image. Note that the \( f_{emb}^{(h)} \) is different from the \( f^{(h)} \), as pointed before.

Figure 2 shows the correlation coefficients between the embedded fingerprint \( f_{emb}^{(10)} \) of the 10-th user and the embedded fingerprints \( f_{emb}^{(h)} \) of all the users. It can be seen that the fingerprint has low correlation with all the other fingerprints and a high correlation with itself.

![Fig. 2. Correlation coefficients between the fingerprint of the 10-th user and the fingerprints of all the users.](image)

Afterwards, the decrypted and fingerprinted coefficients have to be extracted from quaternions \( x_m^{(h)} \) and put back in the same places in DCT blocks from which they were previously selected. In the end, the receiver has to perform the inverse discrete cosine transform (IDCT) for each block independently and recreate an image by combining all of the blocks and all the color channels. As the result, the fingerprint spreads through all of the pixels in each block.

In order to extract the fingerprint from the fingerprinted image, the distribution side has to perform a non-blind detection according to:

The example of implementation of the proposed method is shown on Figure 3, which presents the original images of Lena and a garden, their encrypted and decrypted versions, as well as the fingerprints extracted from the decrypted images. Note that there is only 1 DCT coefficient per block selected for the encryption.

In case of the illegal redistribution of the image, i.e. the piracy, the distribution side has to calculate the embedded fingerprints using formula (24) for the pirate copy and each of user’s copy. Then, the distribution side has to calculate correlation coefficients for the fingerprint extracted from the pirate copy and the fingerprints of all users. If the calculated correlation coefficient for a specific user’s fingerprint is greater than a fixed detection threshold, this specific user is assumed a pirate.
5. RESEARCH RESULTS

The following scenario was simulated. A total of 500 users had ordered color images. Ten different color images of size 512 by 512 pixels were used which presented various natural objects, people, animals, etc. The fingerprints that were used were bipolar Gold codes [18] of length 4096 which were encapsulated in sequences of 1024 quaternions. The embedding strength $\alpha$ was chosen as $\alpha = 0.025$ and the PSNR ranged from 41 to 42 dB, so the embedded fingerprints remained unnoticed to the human eye. A distribution of each of the 10 images was repeated 3 times, each time for a different set of keys. This gave a total of 30 simulated distributions. For each of the 30 simulated distributions, 4 attacks occurred, i.e. collusion attacks by averaging the fingerprinted copies performed by 40, 60, 80 and 100 pirates.

The correlation coefficient values for a fingerprint from the pirate copy and each user’s fingerprint was compared with a detection threshold $t$. If the correlation coefficient value was greater than the threshold $t$, a user was considered a pirate. The true positive (TP) means that a pirate is correctly identified and false positive (FP) means that an innocent user is wrongly accused. Figure 4 contains (a) a plot of $P_{TP}$ against $t$, (b) a plot of $P_{FP}$ against $t$ and (c) the ROC (Receiver Operating Characteristics) curve, which is a plot of $P_{TP}$ against $P_{FP}$. Different line styles on the plots mean different numbers of pirates involved in a collusion attack.

The method’s effectiveness depends on the chosen acceptable $P_{FP}$. E.g., let us assume that the acceptable $P_{FP} = 0.01$, then $P_{TP} = 0.996$ in case of a collusion attack of 60 pirates, 0.963 for 80 pirates, and 0.853 for 100 pirates. Therefore, every pirate can be effectively identified with a very high success rate, even if the number of pirates in the collusion is up to 80. Moreover, we achieved a faultless pirate identification in all of the simulations, i.e. $P_{TP} = 1$ and $P_{FP} = 0$, in case of a collusion of 40 pirates.

This is a very good result in comparison to other known methods. For example, the Chameleon cipher [6] is robust against up to 4 pirates in collusion. The Fingercasting scheme [7,8] is based on Cox’s watermarking scheme [2] which was tested for a collusion of only 5 pirates. Parviainen and Parnes’s fingerprinting method [20] is robust against up to 10 pirates in collusion. The Hillcast method [9,10] can identify up to 10 pirates with a high success rate. The Improved Hillcast [11,12] can identify up to 30 pirates with a high success rate. A fingerprinting feature in CBC quaternion encryption [15] can identify up to 30 pirates with a high success rate, although this is not a JFD method and it scales very poorly. In comparison to these techniques and based on the experimental results we can claim that the proposed method has high robustness against collusion attacks.

6. CONCLUSION

The paper presents effective solution to the problem of piracy in the form of a joint fingerprinting and decryption method based on quaternion sandwich product for color images. Each component of the color image is represented as a point in 3D space, which is formed by 3 color channels. Encryption and fingerprinting can be interpreted as rotation, scaling and translation of these points.

The performed experiments show great robustness against collusion attacks. Even for a collusion attack of up to 80 pirates there is still a high probability of identifying the pirates with a very low probability of
Fig. 4. The method’s resistance to collusion attacks in case of 40, 60, 80, 100 pirates.

40 pirates ——— 80 pirates
———— 60 pirates ——— 100 pirates

accusing innocent users. In case of a collusion attack of 40 pirates we noted the results of a faultless pirate identification. Although the paper only concerns fingerprinting for still images, the method can also be easily extended to the fingerprinting of video data.

Since the simulation results are very promising, the development of the presented JFD method based on quaternion sandwich product will continue. Future research will focus on robustness against collusion attacks combined with a compression, as well as better extraction and pirate identification algorithm.

REFERENCES