Circuits and Systems, 2016, 7, 1722-1733
Published Online June 2016 in SciRes. http://www.scirp.org/journal/cs
http://dx.doi.org/10.4236/cs.2016.78149

Novel Scheme for Compressed Image Authentication Using LSB Watermarking and EMRC6 Encryption

S. J. Jereesha Mary[1], C. Seldev Christopher[2], S. Sebastin Antony Joe[3]

[1]Faculty of EE, Anna University, Chennai, India
[2]Department of CSE, St. Xavier's Catholic College of Engineering, Nagercoil, India
[3]Faculty of I&C, Anna University, Chennai, India
Email: joejerisia@gmail.com, seldev@sxcce.edu.in, ssa_joe@yahoo.com

Received 15 April 2016; accepted 12 May 2016; published 17 June 2016

Abstract

In the current era, transmission and storing of medical data in the digital form is of great concern and thus the requirement for content authentication has aroused. As a solution to these, digital watermarking techniques and encryption schemes have been used to secure medical data like medical images. In this paper a combination of two algorithms to provide image authentication for medical images in the compressed format is proposed. In the proposed method, the watermark image is encrypted using the Enhanced modified RC6 block cipher (EMRC6) algorithm and the encrypted watermark image is watermarked using the simple Least significant Bit (LSB) watermarking technique. The watermarked output image shows no visual imparity and the watermark which has been extracted has no visual difference. The test results show that the watermarked image has high quality and the watermark is very secure. Also the PSNR value of proposed method is 44.966 on an average and 43.0633 for the existing system where LSB technique is integrated with MRC6 for security of watermark. Hence the work is aimed to increase the embedding volume and make the watermark more secure which is the basic requirement of medical image security.

Keywords

Watermark, EMRC6, LSB, Content Authentication, Encryption

1. Introduction

In the recent era where the data are being transformed into digital content, the need for security of medical images

How to cite this paper: Mary, S.J.J., Christopher, C.S. and Joe, S.S.A. (2016) Novel Scheme for Compressed Image Authentication Using LSB Watermarking and EMRC6 Encryption. *Circuits and Systems*, **7**, 1722-1733.
http://dx.doi.org/10.4236/cs.2016.78149

becomes intense. Digital Asset and Right Management Systems (DARMS) is intended to be used with data that are either encrypted or compressed. Encryption schemes are used to provide security to digital data like text, image, audio, video, etc. Encryption is a process in which the input is combined with a key to produce an output that is not in the human readable form. Similarly watermarking methods are used to minimize forgery in the digital data that is being stored in a computer or being transferred over the network.

The method of embedding an image inside the cover image for the purpose of security is called watermarking. Larobina explained various medical image formats that can be used for watermarking [1]. As compressed images are mainly used as input in digital watermarking algorithms, Rabbani described a lossless image compression format for storing images of any type that occupies less storage space in [2]. To overcome the weaknesses of digital signature for content authentication, many researchers proposed watermarking techniques for digital data authentication. In digital signature, content modification or data tampering can be found out. But the location where such alteration has been found could not be identified [3]-[6].

In addition to watermarking techniques, encryption schemes increase the reliability and security of digital content. Depending on the technology used, cryptographic methods are difficult to detect. The usage of an additional key makes it more robust. Watermarking is said to be a form of communication, since the basic task of watermarking is the reliable embedding and detection of digital content. Secure delivery of content is a major task of cryptography in addition to providing reliability. The decrypted form of the content has no protection in cryptography, but it is complimented by watermarking techniques by embedding the watermark inside the digital content. Thus the "analog hole" created by encryption schemes can be sealed by watermarking techniques [7]-[10].

Lot of researchers combined watermarking schemes with encryption algorithms to increase the security of the digital content. RSA, Paillier, Goldwasser-Micali, Elgamelare asymmetric encryption schemes with homomorphic property and they have their downsides. There is loss of compression efficiency in the output cipher text if the message size is small and compression loss is reduced, but the payload capacity decreases if the message size is large in the above said schemes [11]-[14].

The downsides of those are overcome by RC4 symmetric stream cipher scheme with homomorphic property which was proposed by Subramanyam *et al*. [15]. The chances of *data trade-off attacks* which are based on the key scheduling algorithm are caused by low *sampling resistance* in RC4 [16]. The symmetric block cipher RC5 which is more secure and robust than RC4 due to its increased number of rounds to work with the watermarking schemes used in [15] was proposed by Gayathri I.K. [17]. With less than 18 rounds RC5-64 algorithm (64-bit blocks) is prone to differential attack, when the chosen plain text is 2^{44} [17]. RC6 is being used instead of RC5 due to its increased use of registers which was proposed by Kukoo Anna Mathew [18].

Elashry in [19] proposed a method in which LSB is integrated with RC6 to provide security with good image quality. But RC6 undergoes differential linear attack, statistical attack and X^2 attack [20]-[22]. Later to overcome the disadvantages of RC6 an Enhanced version of RC6 (ERC6) is used. It has 8 working registers and it acts on 256-bits input/output blocks.ERC6 encrypts at about 17.3 MB/sec making it about 1.7 times faster than RC6. But it is prone to X^2 attack up to 44 rounds [23].

So in order to overcome the drawbacks of ERC6 a modified version of RC6 (MRC6) is being found with better performance. It uses sixteen working registers instead of four registers in RC6. MRC6 achieves greater security at fewer rounds thus increasing the throughput with minimum encryption/decryption time [24]. But the Enhanced Modified Version of RC6 (EMRC6) has 32 working registers instead of sixteen in MRC6, and integer multiplication is used as an extra primitive operation which intensifies the diffusion obtained per round thus attaining high security and maximum throughput in less number of rounds [25].

LSB technique is used where the embedding process is simple. It is one in which the image pixels of the cover image are replaced by the bits from the secret message. Embedding can be done in any of the eight bits in a bit plane. Bamatraf proposed a LSB technique where the hiding of data is done in the third and fourth LSB of the cover image [26].

Singh *et al*. proposed a watermarking method using replacement of second LSB with inverse of LSB which is a powerful method for image authentication and copyright protection [27]. Puneet analysed various image watermarking using LSB algorithms and the results show that the hiding of the secret data in the first bit is without noticeable distortion on it [28].

The current work is intended to convert the input into JPEG2000 format and encrypt the watermark with LSB watermarking scheme and then watermarking the encrypted watermark with EMRC6 encryption scheme so as to

provide content authentication.

Figure 1 shows the basic model for content authentication, in which the cover image is a color image and the watermark image is the one that has to be embedded in to the cover/input image. The watermark image in encrypted using any encryption scheme and the output encrypted watermark is then watermarked using any spatial or frequency domain watermarking algorithm producing the final watermarked output.

2. Proposed System

The proposed system is aimed at combining LSB watermarking method with Enhanced Modified Version of RC6 (EMRC6) encryption scheme to provide content authentication for compressed images. The cover image "I" may be of any image type and is given as input to the JPEG2000 encoder. The JPEG2000 encoder processes the input image by undergoing five stages by dividing the image into rectangular tile that does not overlap and then it undergoes discrete wavelet transformation (DWT) and is quantized and is further divided into different bit planes.

This is then block coded with optimized truncation which results in stream of compressed byte which is packed into different wavelet packets. This JPEG2000 encoded output is represented as "I_j".

The watermark image represented as "W_m" is encrypted using Enhanced Modified Version of RC6 (EMRC6). The encryption process takes the input watermark and is converted into stream of bits. This is encrypted using the key "K" which is generated from the key of "b" bytes supplied by the user. The encryption process has a round function which has 16 rotations per round. The input stream undergoes 18 such rounds to produce the output ciper "E_w". The Encoded cover image "I_j" and the output cipher "E_w" are given as input to the LSB watermarking scheme

The LSB embedding scheme replaces few characteristics of the last bit of each pixel of the cover image "I_j" with few information from the watermark image "W_m". A pseudo Random Number generator (PRNG) that produces a seed value is used with the LSB embedding in order to overcome the attacks on LSB embedding since LSB is a fragile scheme. Thus the output of the embedding process is the watermarked image "W_e" which is highly secure and has good image quality. The Peak signal to noise ratio (PSNR) value is high and the Mean Square Error (MSE) value is low when compared to the previous method where LSB technique is integrated with MRC6. This method increases embedding volume and make the watermark more secure thus providing content authentication for compressed medical images. **Figure 2** shows the embedding of the watermark in the encrypted image.

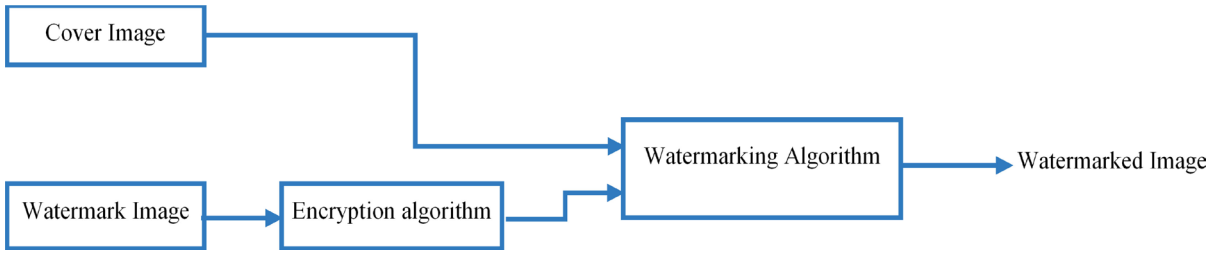


Figure 1. Basic model for content authentication.

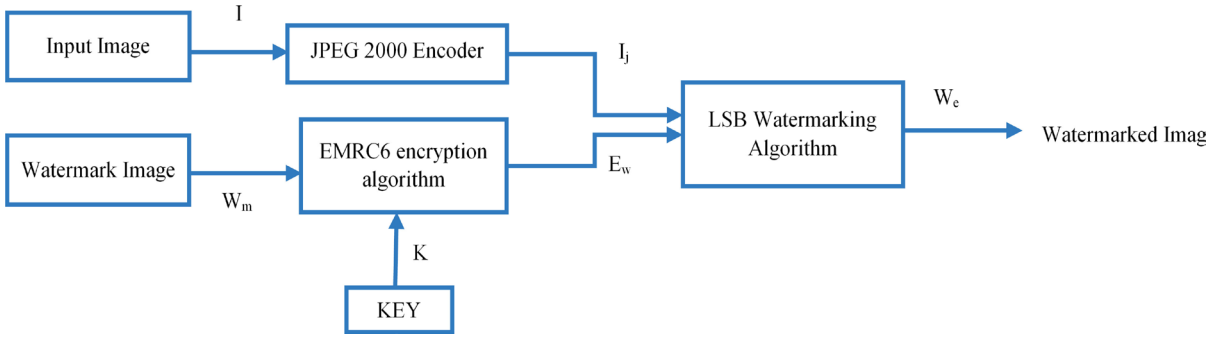


Figure 2. Embedding of watermark.

2.1. JPEG2000 Encoder

In the Proposed algorithm the input is any image and this image is converted into JPEG 2000 compressed code image using JPEG 2000 encoder by undergoing five steps. First the image is split into non overlapping tiles which are unsigned values and is reduced by a constant value. Then Discrete wavelet Transformation (DWT) is done followed by quantization and further the co-efficient are split into different bit-planes using embedded block coding with optimized truncation (EBCOT) coding method. As a final step compressed stream are packed into different wavelet packages [29].

2.2. Enhanced Modified Rivest Cipher 6 (EMRC6)

EMRC6 (32/18/16) has 32 registers each with 'w' bit words, whereas the working registers is less in numbers in the previous version of RC6. EMRC6 has an integer multiplication as an extra basic operation which increases the diffusion attained per round. This provides high security, increase in number of rounds and greater through-put. It can process 1024 bits as a single block per round. The EMRC6 algorithm has three basic modules.

1) EMRC6 key expansion.
2) EMRC6 encryption.
3) EMRC6 decryption.

2.2.1. EMRC6 Key Expansion

The key expansion algorithm of EMRC6 is almost similar to the one in the previous version of RC6. But the main difference in EMRC6 is, more number of words are extracted from the key supplied by the user. The key of length "b" bytes where $0 \le b \le 255$ is supplied by the user. To this user defined key, enough zeroes are added to make the length of the key equal to non-zero integral values and stored in $U[0], U[1], \cdots, U[b-1]$. These user supplied "b" bytes of keys is stored in another array "V" in the little endian format as $V[0], V[1], \cdots, V[c-1]$. The values from "V" table is expanded and stored in a table $T[0], T[1], \cdots, T[16r+31]$, thus producing 16r + 32 sub keys. The left barrel shifter shifts the sub keys by three positions and the resultant is stored again in "T" table. The below algorithm shows the key expansion algorithm of EMRC6.

Input: An array V[0,1,2,…,c-1] that contains 'b' bytes of user supplied key converted to 'c' words and 'r' number of rounds.
Output: An array S[0,1,2,…,16r+31] that has w-bit round keys
Procedure:

```
T[0] = Pw                          // Pw value is defined in equation (1).
For i = 1 to 16r+31 do
T[i] = T[i-1] + Qw                 // Qw value is defined in equation (2).
R1 = R2 = i = j = 0
z= 3 * Max (c, 16r+32)
for y = l to z do
{
R1 = T[i] = (T[i] + R1 + R2) <<< 3
R2 = V[j] = (V[j] + R1+ R2) <<< (Rl+R2)
i = (i + 1) Mod(16r+31)
j = (j + l) Mod c
}
```

The two values P_w and Q_w are called magic constants and is defined as

$$P_w = odd\left((e-2)2^w\right) \text{ and} \tag{1}$$

$$Q_w = odd\left((Ø-1)2^w\right). \tag{2}$$

where

e = 2.7182818 and Ø = 1.618033 is called the golden constant.

2.2.2. EMRC6 Encryption Algorithm

EMRC6 encryption module convert the input into the cipher output using the generated sub key. It uses 32 working registers ($R[0], R[1], \cdots, R[31]$) to store the initial input and the final output. The steps involved in

EMRC6 encryption is

- Addition (+)
- Bitwise EX-OR operation
- Left rotation, a <<< b.
- Integer Multiplication modulo 2^n (*).

The 32 working registers contain the initial input for the encryption process represented as R[ ]. The first byte of the input is stored in the LSB of first register *i.e.*, R[0] and the last byte is stored in the MSB of the last register. For example, the assignment from right to left is parallel as (R1, R2, R3, R4) = (R2, R3, R4, R1).

Input: Array R [0,1,2,…, 31] has the plaintext in it ,'r' number of rounds and 'w' bit round keys from T[0,1,2,…, 16r + 31].
Output: cipher text in register, R[0,1,…,31]
Procedure:

```
R2 = R2 + T[0],      R4 = R4 + T[1],      R6 = R6 + T[2],      R8 = R8 + T[3],
R10 = R10 + T[4],    R12 = R12 + T[5],    R14 = R14 + T[6],    R16 = R16 + T[7],
R18 = R18 + T[8],    R20 = R20 + T[9],    R22 = R22 + T[10],   R24 = R24 + T[11],
R26 = R26 + T[12],   R28 = R28 + T[13],   R30 = R30 + T[14],   R32 = R32 + T[15],
For i = 1 to r do
    {
        k1 = (R2 * (2R2 + 1)) <<< log w
        l1 = (R4 * (2R4 + 1)) <<< log w
        m1 = (R6 * (2R6 + 1)) <<< log w
        n1 = (R8 * (2R8 + 1)) <<< log w
        t1 = (R10 * (2R10 + 1)) <<< log w
        u1 = (R12 * (2R12 + 1)) <<< log w
        v1 = (R14 * (2R14 + 1)) <<< log w
        z1 = (R16 * (2R16 + 1)) <<< log w
        k = (R18 * (2R18 + 1)) <<< log w
        l = (R20 * (2R20 + 1)) <<< log w
        m = (R22 * (2R22 + 1)) <<< log w
        n = (R24 * (2R24 + 1)) <<< log w
        t = (R26 * (2R26 + 1)) <<< log w
        u = (R28 * (2R28 + 1)) <<< log w
        v = (R30 * (2R30 + 1)) <<< log w
        z = (R32* (2R32 + 1)) <<< log w
        R1 = ((R1⊕ k1) <<< l1) + T[16i]
        R3 = ((R3 ⊕ l1 ) <<< k1) + T[16i + 1]
        R5 = ((R5 ⊕ m1) <<< n1) + T[16i + 2]
        R7 = ((R7 ⊕ n1) <<< m1) + T[16i + 3]
        R9 = ((R9 ⊕ t1 ) <<< u1 ) + T[16i + 4]
        R11 = ((R11 ⊕ u1) <<< t1) + T[16i + 5]
        R13 = ((R13 ⊕ v1) <<< z1) + T[16i + 6]
        R15 = ((R15 ⊕ z1) <<< v1) + T[16i + 7]
        R17 = ((R17 ⊕ k) <<< l) + T[16i + 8]
        R19 = ((R19 ⊕ l ) <<< k ) + T[16i + 9]
        R21 = ((R21 ⊕ m) <<< n) + T[16i + 10]
        R23 = ((R23 ⊕ n) <<< m) + T[16i + 11]
        R25 = ((R25 ⊕ t ) <<< u) + T[16i + 12]
        R27 = ((R27 ⊕ u ) <<< t) + T[16i + 13]
        R29 = ((R29 ⊕ v) <<< z) + T[16i + 14]
        R31 = ((R31 ⊕ z) <<< v) + T[16i + 15]
        (R1, R2, ..., R31, R32) = (R2, R3, …, R32, R1)
    }
        R1 = R1 + T[16r + 16],   R3 = R3 + T[16r + 17],   R5 = R1 + T[16r + 18],   R7 = R3 + T[16r + 19],
        R9 = R1 + T[16r + 20],   R11 = R3 + T[16r + 21],  R13 = R1 + T[16r + 22],  R15 = R3 + T[16r + 23],
        R17 = R1 + T[16r + 24], R19 = R3 + T[16r + 25],   R21 = R1 + T[16r + 26],  R23 = R3 + T[16r + 27],
        R25 = R1 + T[16r + 28], R27 = R3 + T[16r + 29],   R29= R1 + T[16r + 30],   R31 = R3 + T[16r + 31],
```

2.2.3. EMRC6 Decryption Algorithm

The EMRC6 decryption process reproduces the original content from the cipher using the sub key. This is the inverse operation of EMRC6 encryption. The following algorithm represents the EMRC6 decryption process. Various steps involved in decryption process are

- Integer subtraction (−)
- Bit wise EX-OR
- Integer multiplication
- Right shift, a >>> b.

Input: Array R[0, 1, 2, …, 31] has the plaintext in it ,'r' number of rounds and 'w' bit round keys from T[0, 1, 2, …, 16r + 31].
Output: plaintext stored in register, R[0, 1, …, 31]
Procedure:

R31 = R31 − T[16r + 31], R29 = R29 − T[16r + 30], R27 = R27 − T[16r + 29], R25 = R25 − T[16r + 28],
R24 = R24 − T[16r + 27], R23 = R23 − T[16r + 26], R19 = R19 − T[16r + 25], R17 = R17 − T[16r + 24],
R15 = R15 − T[16r + 23], R13 = R13 − T[16r + 22], R11 = R11 − T[16r + 21], R9 = R9 − T[16r + 20],
R7 = R7 − T[16r + 19], R5 = R5 − T[16r + 18], R3 = R3 − T[16r + 17], R1 = R1 − T[16r + 16],
{ For i = r down to 1 do
{
(R1, R2, …, R31, R32) = (R32, R1, …., R31)
z = (R32 * (R32 + 1)) <<< log w
v = (R30 * (2R30 + 1)) <<< log w
u = (R28 * (2R28 + 1)) <<< log w
t = (R26 * (2R26 + 1)) <<< log w
n = (R24 * (2R24 + 1)) <<< log w
m = (R22 * (2R22 + 1)) <<< log w
l = (R20 * (2R20 + 1)) <<< log w
k = (R18 * (2R18 + 1)) <<< log w
z1 = (R16 * (2R16 + 1)) <<< log w
v1 = (R14 * (2R14 + 1)) <<< log w
u1 = (R12 * (2R12 + 1)) <<< log w
t1 = (R10 * (2R10 + 1)) <<< log w
n1 = (R8 * (2R8 + 1)) <<< log w
m1 = (R6 * (2R6 + 1)) <<< log w
l1 = (R4 * (2R4 + 1)) <<< log w
k1 = (R2 * (2R2 + 1)) <<< log w
R31 = ((R31) − T[16i + 15] >>> z) ⊕ v
R29 = ((R29) − T[16i + 14] >>> v) ⊕ z
R27 = ((R27) − T[16i + 13] >>> u) ⊕ t
R25 = ((R25) − T[16i + 12] >>>t) ⊕ u
R23 = ((R23) − T[16i + 11] >>>n) ⊕ m
R21 = ((R21) − T[16i + 10] >>>m) ⊕ n
R19 = ((R19) − T[16i + 9] >>>l) ⊕ k
R17 = ((R17) − T[16i + 8] >>> k) ⊕ l
R15 = ((R15) − T[16i + 7] >>> z1) ⊕ v1
R13 = ((R13) − T[16i + 6] >>> v1) ⊕ z1
R11 = ((R11) − T[16i + 5] <<< u1) ⊕ t1
R9 = ((R9) − T[16i + 4] <<< t1) ⊕ u1
R7 = ((R7) − T[16i + 3] <<< n1) ⊕ m1
R5 = ((R5) − T[16i + 2] <<< m1) ⊕ n1
R3 = ((R3) − T[16i + 1] <<< l1) ⊕ k1
R1 = ((R1) − T[16i] <<< k1) ⊕ l1
}
R32 = R32 − T[15], R30 = R30 − T[14], R28 = R28 − T[13], R26 = R26 − T[12],
R24 = R24 − T[11], R22 = R22 − T[10], R20 = R20 − T[9], R18 = R18 − T[8],
R16 = R16 − T[7], R14 = R14 − T[6], R12 = R12 − T[5], R10=R10− T[4]
R8 = R8 − T[3], R6=R6− T[2], R4 = R4 − T[1], R2=R2− T[0]

2.3. LSB Watermarking

Least significant Bit (LSB) technique is based on exchanging few characteristics of each pixel's last bit with some of the information from the input image. The embedding can be done in any of the bit plane, but LSB embedding focuses on embedding in the least significant bit of each pixel. This bit plane is chosen in order to reduce the difference in colors in the watermarked image. Basically LSB embedding scheme is fragile and hence it can be easily broken. But it is being widely used because of its simplicity. This overcomes cropping attack but undergoes a number of attacks if all the LSB is changed to one. For example, Let "01110100" be the bits to be embedded into the least significant bit (LSB) of the input image values. The watermarked output obtained using LSB technique for the given sample input is shown in **Table 1**.

2.3.1. LSB Embedding Algorithm

The steps in LSB embedding algorithm is listed below

- "E_w" is read and is stored as matrix element "E_w[][]".
- "I_j" is normalized and rounded off to the adjacent integer with a bit precision of eight.
- Determine the size of "E_w" and "I_j".
- Expand "E_w[][]" such that the size of watermark is same as that of "I_j". The Components in "E_w[][]" are individual bits that represent pixel values of the "E_w".
- "I_j" is split into pixel and stored in "I_j[][]".
- LSB of each element from "I_j[][]" is replaced by the matching elements from "E_w[][]".
- The resultant matrix is converted into the watermarked image.

2.3.2. LSB Extraction Algorithm

The steps in LSB extraction process is listed below

- "W_e" is read and the pixel values are stored in "W_e[][]".
- Find the size of "W_e".
- An expansion matrix is formed by extracting the LSB of each pixels from "W_e[][]".
- The bits per pixel of "W_e" are determined and the bits are clustered based on it. The expansion matrix consists of repeated pattern of bits at constant intervals.
- Thus multiple watermarks are recovered inside the expansion matrix.

3. Analyzing and Evaluating the Performance

3.1. Analyzing EMRC6 Encryption

EMRC6 is better than any other version of RC6 due to its increased complexity, security and throughput. The EMRC6 encryption is three times faster than RC6 and the throughput is high when compared to its predecessor MRC6. **Table 2** shows the comparison between various parameters of EMRC6 and the parameters of other version of RC6.

Table 1. Watermarked image values.

Input	Watermark	Watermarked Output
00011100	0	00011100
11110010	1	11110011
01011000	1	01011001
10011010	1	10011011
00100111	0	00100110
11101000	1	11101001
11001000	0	11001000
00110101	0	00110100

Table 2. Comparisons of RC6 versions.

Parameters	RC6	ERC6	MRC6	EMRC6
w/r/b	32/20/16	32/16/16	32/16/16	32/18/16
Working Registers	4	8	16	32
Block size (bits)	128	256	512	1024
No. of rounds	20	16	16	18
Rotations per round	2	4	8	16
Encryption time (sec. for 4 MB)	0.281	0.251	0.201	0.171
Throughput (Mb/s)	10.8	17.3	19.5	34.13
Sub-keys	2r + 4	4r + 8	8r + 16	16r + 32
Speed	1 unit	1.7 times faster than RC6	Twice faster than RC6	Thrice faster than RC6
Attack	Linear, differential, chi-squared attack	chi-squared attack	chi-squared attack	No attack

3.2. Security Issues

The diffusion per round in achieved by the use of multiplication which provides high security in lesser number of rounds. Because of sixteen rotations per round the complexity of the algorithm is increased which proportionally increases the security. It is robust against differential linear attack, statistical attack and chi-square attack due to its increased no of rotations per round [25].

Figure 3 shows that the encryption time is increased with increase in the size of the data block. But when compared with the previous version of RC6 it has got better encryption time. **Figure 4** shows the throughput

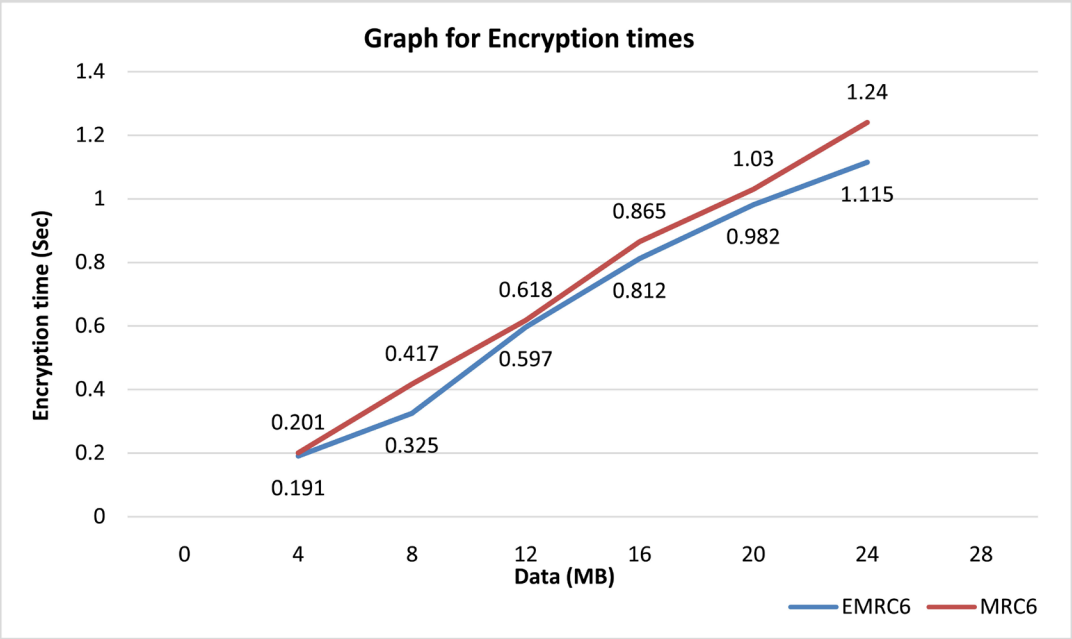


Figure 3. Consequence of encryption time over varied data size.

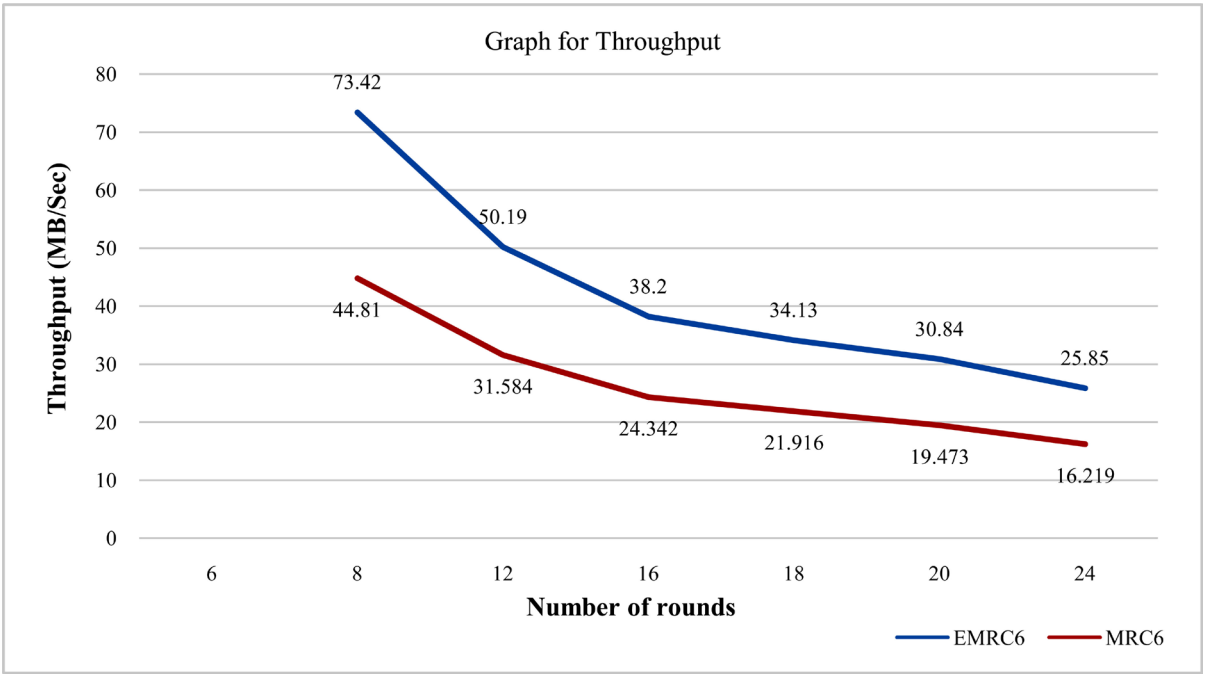


Figure 4. Consequence of number of rounds on throughput.

decreases with increased number of rounds which gives the conclusion that the security is high with more number of rounds since security and throughput are inversely proportional.

3.3. Correlation Coefficient

Correlation Coefficient is the value between pixels in the same place in the input image and watermarked image. Correlation Coefficient is used to measure the image quality between the pixels in the original image and watermarked cipher image at a particular location.

Coefficient Correlation $\gamma(I,C)$ is found using Equation (3).The Expectation and variant value to be substituted in Equation (3) is found using Equation (4) and Equation (5).

$$\gamma_{IC} = \frac{E\left(\left(W_m - E\left(W_m\right)\right)\left(E_w - E\left(E_w\right)\right)\right)}{\sqrt{D\left(W_m\right)D\left(E_w\right)}} \tag{3}$$

$$D\left(W_m\right) = \frac{1}{n}\sum_{i=1}^{N}\left(W_{m_i} - E\left(W_m\right)\right)^2 \tag{4}$$

$$E\left(W_m\right) = \frac{1}{n}\sum_{i=1}^{N}\left(W_{m_i}\right) \tag{5}$$

where,

$E\left(W_m\right)$ & $E\left(E_w\right)$ are Expectation of pixel from W_m and E_w

$D\left(W_m\right)$ & $D\left(E_w\right)$ are Variants of pixel from W_m and E_w.

Table 3 shows the correlation coefficient of three sample images for the existing method and the proposed method which concludes that the correlation coefficient is low for the proposed method is low when compared with other encryption scheme.

3.4. Analysis of LSB Embedding Efficiency

The image quality of the output watermarked cipher image "W_e" is found using the PSNR and MSE calculation. Equation (6) depicts the PSNR calculation and Equation (7) calculates the MSE value.

$$\text{PSNR} = 10\log_{10}\left(\frac{255^2}{\text{MSE}}\right) \tag{6}$$

$$\text{MSE} = \frac{1}{mn}\sum_{x=0}^{m-1}\sum_{y=0}^{n-1}\left(I_j\left(x,y\right) - W_e\left(x,y\right)\right)^2 \tag{7}$$

where

m is the row values of the images.

n is the column values of the images.

x, y is the pixel values of the images.

The PSNR value from **Table 4** predicts that the PSNR value of the sample images is high when compared with the previous method and hence the quality of the watermarked cipher image is of good quality for the proposed method. **Table 5** includes the output of each level and the overall output, for the test images.

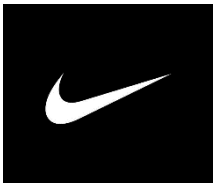
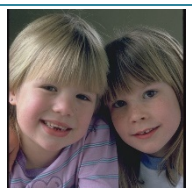
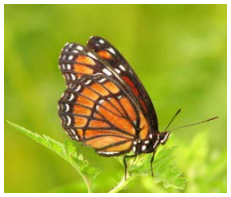
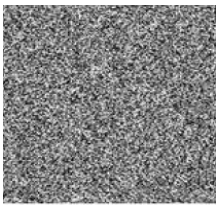
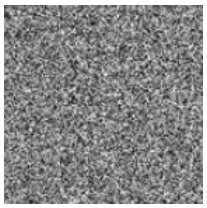
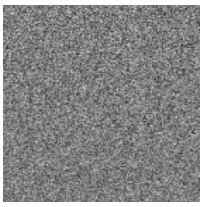
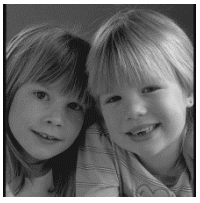
Table 3. Comparison of correlation coefficient.

Image	Nike	Lenna	Girls
Correlation coefficient of Existing Method	0.0001749	0.0033	0.00016
Correlation coefficient of Proposed Method	0.00016	0.0021	0.00002

Table 4. PSNR of the watermarked encrypted Image.

IMAGE	Nike.png	Lenna.bmp	Girls.jpg
PSNR of Existing Method	41.82	42.31	45.06
PSNR of Proposed Method	43.6	44.1	47.2

Table 5. Overall output for test data.

Image			
Watermark			
Encrypted watermark			
Watermarked cipher output			
Size of the Original Image	45 Kb	657 Kb	68 Kb
Size of the Watermark Image	6 Kb	18 Kb	10 Kb
PSNR of Watermarked cipher image (Proposed System)	43.6	44.1	47.2
PSNR of Watermarked cipher image (existing System)	41.82	42.31	45.06
Correlation coefficient of encrypted watermark (Proposed System)	0.00016	0.0021	0.00002
Correlation coefficient of encrypted watermark (existing System)	0.0001749	0.0033	0.00016

4. Conclusions

The combination of EMRC6 (Enhanced modified version of RC6) Scheme with LSB (Lest Significant bit) is a proper method for medical image authentication. The input image is converted into JPEG2000 image to make encryption and embedding scheme simple. EMRC6 provides high security since it withstands almost all attacks which was imposed on previous RC6 version. The security level of watermark is increased by this encryption and the watermark embedding capacity also improved. Even though LSB is a fragile watermark scheme, it is best suited for content authentication. But the watermark is encrypted to make the watermark robust. So that extraction of the watermark is difficult. The encryption speed of EMRC6 is high and the throughput is high when compared with its predecessor. The correlation coefficient is very low proving that the image quality is good. After embedding the PSNR value is high and MSE value is low when compared with other algorithm.

The upcoming work is to provide copyright production using a frequency domain algorithm and EMRC6 encryption scheme.

References

[1] Larobina, M. and Murino, L. (2014) Medical Image File Formats. *Journal of Digital Imaging*, **27**, 200-206.

[2] Rabbani, M. and Joshi, R. (2002) An Overview of the JPEG 2000 Still Image Compression Standard. *Signal Processing*: *Image Communication*, **17**, 3-48. http://dx.doi.org/10.1016/S0923-5965(01)00024-8

[3] Wang, M.S. and Chen, W.C. (2007) A Majority-Voting Based Watermarking Scheme for Colour Image Tamper Detection and Recovery. *Computer Standards & Interfaces*, **29**, 561-570. http://dx.doi.org/10.1016/j.csi.2006.11.009

[4] Eggers, J.J. and Girod, B. (2001) Blind Watermarking Applied to Image Authentication. *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, **3**, 1977-1980. http://dx.doi.org/10.1109/icassp.2001.941335

[5] Lin, E.T., Podilchuk, C.I. and Delp, E.J. (2000) Detection of Image Alterations Using Semi-Fragile Watermarks. In: *Electronic Imaging*, International Society for Optics and Photonics, 152-163. http://dx.doi.org/10.1117/12.384969

[6] Kundur, D. and Hatzinakos, D. (1999) Digital Watermarking for Telltale Tamper Proofing and Authentication. In: *Proceedings of the IEEE*, **87**, 1167-1180. http://dx.doi.org/10.1109/5.771070

[7] Nadeem, A. and Javed, M.Y. (2006) A Performance Comparison of Data Encryption Algorithms. *International Conference on Information and Communication Technologies*, 27-28 August 2005, 84-89.

[8] Cox, I.J., Doërr, G. and Furon, T. (2006) Watermarking Is Not Cryptography. In: Shi, Y. and Jeon, B., Eds., *Digital Watermarking*, Vol. 4283, Springer Berlin/Heidelberg, 1-15. http://dx.doi.org/10.1007/11922841_1

[9] Farah, T., Hermassi, H., Rhoouma, R. and Belghith, S. (2013) Watermarking and Encryption Scheme to Secure Multimedia Information. *World Congress on Computer and Information Technology* (*WCCIT*), Sousse, 22-24 June 2013, 1-5. http://dx.doi.org/10.1109/wccit.2013.6618760

[10] Bouslimi, D., Coatrieux, G. and Roux, Ch. (2011) A Joint Watermarking/Encryption Algorithm for Verifying Medical Image Integrity and Authenticity in Both Encrypted and Spatial Domains. *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Boston, 30 August 2011-3 September 2011, 8066-8069. http://dx.doi.org/10.1109/iembs.2011.6091989

[11] ElGamal, T. (1985) A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. *IEEE Transactions on Information Theory*, **31**, 469-472. http://dx.doi.org/10.1109/TIT.1985.1057074

[12] Rivest, R., Shamir, A. and Adleman, L. (1978) A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Commun. ACM*, **21**, 120-126. http://dx.doi.org/10.1145/359340.359342

[13] Paillier, P. (1999) Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. *Lecture Notes in Computer Science*, **1592**, 223-238. http://dx.doi.org/10.1007/3-540-48910-x_16

[14] Goldwasser, S. and Micali, S. (1984) Probabilistic Encryption. *Journal of Computer and System Sciences*, **28**, 270-299. http://dx.doi.org/10.1016/0022-0000(84)90070-9

[15] Subramanyam, A.V., Emmanuel, S. and Kankanhalli, M.S. (2010) Compressed Encrypted Domain JPEG2000 Image Watermarking. *IEEE International Conference on Multimedia and Expo* (*ICME*), Suntec City, 19-23 July 2010, 1315-1320. http://dx.doi.org/10.1109/icme.2010.5583571

[16] Fluhrer, S., Mantin, I. and Shamir, A. (2001) Weaknesses in the Key Scheduling Algorithm of RC4. In: Vaudenay, S. and Youssef, A.M., Eds., *Selected Areas in Cryptography*, Springer, Berlin, 1-24.

[17] Gayathri, I.K. (2013) Digital Watermarking Using RC5 Encryption on JPEG2000 Images. *International Journal of Engineering Research & Technology*, **2**, 1439-1445.

[18] Mathew, K.A. (2013) Watermarking of JPEG2000 Compressed Images with Improved Encryption. *International Journal of Computer Applications Technology and Research*, **2**, 245-249.

[19] Elashry, F., Allah, O.S.F., Abbas, A.M., El-Rabaie, S. and El-Samie, F.E.A. (2009) Homomorphic Image Encryption. *Journal of Electronic Imaging*, **18**, 033002. http://dx.doi.org/10.1117/1.3167847

[20] Borst, J., Preneel, B., Vandewalle, J. and Leuven, K.U. (1999) Linear Cryptanalysis of RC5 and RC6. In: Knudsen, L., Ed., *Fast Software Encryption, Lecture Notes in Computer Science*, Springer-Verlag, Berlin, 16-30.

[21] Gilbert, H., Handschuh, H., Joux, A. and Vaudenay, S. (2000) A Statistical Attack on RC6. In: Goos, G., Hartmanis, J., van Leeuwen, J. and Schneier, B., Eds., *Fast Software Encryption*, Springer-Verlag, Berlin, 64-74.

[22] Miyaji, A. and Takano, T. (2007) Evaluation of the Security of RC6 against the x2-Attack. *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, E90-A, 22-28.

[23] Ragab, A.H.M., Ismail, N.A. and Faragallah, O.S. (2001) Enhancement and Implementation of RC6 Block Cipher for Data Security. *Electronic Engineering Bulletin*, No. 21, 133-137.

[24] El-Fishawy, N., Danaf, T. and Zaid, O. (2004) A Modification of RC6™ Block Cipher Algorithm for Data Security (MRC6). *International Conference on Electrical, Electronic and Computer Engineering*, Cairo, 5-7 September 2004, 222-226. http://dx.doi.org/10.1109/iceec.2004.1374428

[25] Khanapur, N.H. and Patro, A. (2015) Design and Implementation of Enhanced Version of MRC6 Algorithm for Data Security. *International Journal of Advanced Computer Research* (*IJACR*), **5**, 225-232.

[26] Bamatraf, A., Ibrahim, R. and Salleh, M.N.B.M. (2010) Digital Watermarking Algorithm Using LSB. 2010 *International Conference on Computer Applications and Industrial Electronics* (*ICCAIE*), Kuala Lumpur, 5-8 December 2010, 155-159. http://dx.doi.org/10.1109/iccaie.2010.5735066

[27] Singh, A., Jain, S. and Jain, A. (2013) Digital Watermarking Method using Replacement of Second Least Significant Bit (LSB) with Inverse of LSB. *International Journal of Emerging Technology and Advanced Engineering*, **3**, 121-124.

[28] Kr Sharma, P. and Rajni (2012) Analysis of Image Watermarking Using Least Significant Bit Algorithm. *International Journal of Information Sciences and Techniques* (*IJIST*), **2**, 95-101.

[29] Wu, H. and Ma, D. (2004) Efficient and Secure Encryption Schemes for JPEG 2000. *IEEE International Conference on Acoustics, Speech and Signal Processing*, **5**, 869-872.