



Steganography Scheme Based on Reed-Muller Code with Improving Payload and Ability to Retrieval of Destroyed Data for Digital Images

A. M. Molaei*, M. H. Sedaaghi, H. Ebrahimnezhad

Department of Electrical Engineering, Sahand University of Technology, Tabriz, Iran

ABSTRACT: In this paper, a new steganography scheme with high embedding payload and good visual quality is presented. Before embedding process, secret information is encoded as a block using Reed-Muller error correction code. After data encoding and embedding into the low-order bits of the host image, modulus function is used to increase the visual quality of stego image. Since the proposed method is able to embed secret information into more significant bits of the image, it has improved embedding payload. The steps of extracting data from the host image are independent of the original image. Therefore, the proposed algorithm has a blind detection process which is more suitable for practical and online applications. The simulation results show that the proposed algorithm is also able to retrieve destroyed data by intentional or unintentional attacks such as the addition of noise and filtering due to the use of the error correction code. In addition, the payload is improved in comparison with the same techniques.

Review History:

Received: 20 February 2016
Revised: 26 October 2016
Accepted: 29 October 2016
Available Online: 2 November 2016

Keywords:

Blind Detection
High Payload
Low-Order Bits Embedding
Retrieval of Destroyed Data
Steganography

1- Introduction

Data hiding is the knowledge of embedding original message into a host media and then extracting it from received data [1-4]. What should be considered in this process is that cover media should not be high visually distorted after embedding; i.e. eye is not able to distinguish a change in cover media. Cryptography, digital watermarking and steganography are three branches of data hiding techniques [5-7]. In cryptography, sender converts plain text to cipher text using a secret key and it is decrypted on the side of the receiver to extract plain text. In steganography, the embedded data do not have relation to cover media; in other words, the presence of a message in cover media cannot be detected while in watermarking, the embedded data are related to cover media and cannot be removed or replaced. In this paper, we use grayscale images as cover media for steganography that these images are called host images. The obtained image after embedding into the host image is called *stego image*.

In terms of detection technique, data hiding systems are categorized into two types of blind and non-blind [8-10]. In blind systems, the original data (cover) is not needed in detection stage, i.e. the message can be extracted directly from received data without the original host media. In this technique, the authorized user is able to extract message by a secret key. In non-blind systems, the original data is needed in detection stage. In practical applications, blind systems are more suitable than non-blind systems; but since the message is transmitted to the receiver in the presence of noise, non-blind data hiding systems have fewer problems for detection. For solving this problem in blind data hiding systems, additional information is added to the original message that is called

information encoding. Our proposed detection method in this paper is blind.

In terms of embedding method, there are several common techniques for steganography: Least Significant Bit (LSB) substitution, LSB matching and Pixel-Value Differencing (PVD) [11]. In the first type technique which is the most common and simplest method for steganography, secret bits directly replace the LSBs of the host image. In the second type technique, the LSBs of host image are modified and in the third type technique, the difference between two consecutive pixels is calculated to determine the number of embedded bits.

In 2001, Wang et al. [12] used a genetic algorithm for producing substitution table in order to reduce the visual loss of image after simply embedding important information in low-order bits of the image. Substitution table shows how values of low-order bits of host image should be modified. In 2003, Thien and Lin [13] proposed a simple method for embedding data digit by digit in low-order bits of digital images using modulus function which increased embedding payload. Chan et al. [14] on completion of Thien and Lin [13] method, improved quality of stego image using substitution table. In 2005, Ker [15] proposed a new method called least significant bit matching using Histogram Characteristic Function (HCF) to improve embedding in LSB. After him, Mielikainen [16] proposed a modified method based on LSB matching technique assuming that equal payload, fewer changes were made in the host image. In the studies conducted by Wu and Tsai [17], a method was first proposed as Pixel Value Difference (PVD) which was used to hide secret message in grayscale images and provided more embedding capacity than LSB embedding traditional methods with a low loss of the quality of the image. The important characteristics which they considered in their proposed method was that variations of gray level value in smooth areas can be easily detected by eye in

Corresponding author; Email: a_molaye@sut.ac.ir

any image while these variations are less observed with eyes in edge areas. YANG et al. [18] proposed an adaptive substitution method to embed data into the low-order bits with the aim of preventing sudden changes on the edges of the image and also achieving a better quality of stego image. Their method uses masking of light, edges and texture of host image to estimate the number of LSBs. In their work, more bits are embedded in the pixels belonging to the regions of insensitive to noise in comparison with regions of sensitive to noise. In addition, an optimal pixel adjustment process is used to enhance the visual quality of the stego image using LSB substitution. Zhang et al. [19] investigated a method for improving the capacity and the efficiency of embedding in grayscale images. To this end, they presented two algorithms, called high capacity of information hiding (HCIH) and high-quality information hiding (HQIH). The first algorithm aims to achieve high embedding rate and the second algorithm aims to achieve high embedding efficiency. In the study [20], in addition to the LSB substitution technique, a mixed edge detection mechanism is also employed to increase payload. In this mechanism, the combination of Canny edge detection and Log edge detection techniques is used. A second-order steganographic (SOS) method based on pixel pair matching and modification direction exploiting (MDE) is provided in [21]. Unlike the MDE-based methods [22-26] in which only one secret digit in base B can be concealed into each cover pixel pair, the SOS method is able to perform it for two secret digits.

Crandall [27] for the first time introduced the matrix encoding idea and in 2001 Westfeld [28] implemented this idea in his work called F5-a steganography algorithm. Matrix encoding uses linear codes to increase the visual quality of stego image by preserving high embedding capacity. Zhang et al. [29] proposed methods called Hamming+1 and Golary+2 using error correction binary codes which improved visual quality and embedding capacity. Chang [30] could increase embedding payload in methods [27] and [29] using error correction code of (7, 4) Hamming and LSB embedding. Singh and Siddiqui [31] proposed a robust steganography algorithm based on discrete cosine transform (DCT), Arnold transform and chaotic system. Their algorithm is robust against JPEG compression, the addition of noise, low pass filtering, and cropping attacks.

What was presented in most of the provided methods is a trade-off between payload and visual quality of the stego image. On the other hand, the destruction of stego image with attacks such as the addition of noise or filtering which is done to remove or manipulate important and secret information is one of the important aspects of steganography algorithm design. In the field of data hiding, several papers have been presented to improve payload and increase the visual quality of images. Several algorithms have been also presented to reduce destruction of images after attacks. However, there are limited number of papers which have paid attention to all of the above aspects. Considering the existing sensitivities in military fields, legal fields etc., it is important to provide new methods which concurrently consider all of the three problems mentioned above (payload, visual quality, and robustness against attacks). This study aims to improve the payload and the robustness against destruction parameters (with maintaining the good visual quality).

In this paper, we consider robustness against destructive attacks, such as the addition of noise and filtering using Reed-

Muller codes while introducing a new blind steganography algorithm which provides high embedding capacity with a good visual quality. To enhance the quality of stego image, we will use modulus function in the embedding process. To demonstrate the superiority of our proposed method, we will perform different tests on test images and compared results with similar methods. Results of the tests show the extraordinary performance of our proposed method.

Rest of this paper is organized as follows: in section 2, we review the related literature and works. In section 3, we introduce Reed-Muller codes and their encoding and decoding structures. In section 4, we completely describe the proposed algorithm. Section 5 presents simulations and discusses results. In section 6, conclusions are presented.

2- Related Works

In this section, firstly, the proposed method of Chang [30] will be described. Then, we will introduce the low-order bits substitution method proposed by Thein and Lin [13] to improve the visual quality of stego image.

A. (7, 4) Hamming Method

Chang's method [30] employs (7, 4) Hamming code to hide information. First, it forms 16 classes each, including eight different bit strings of length seven. Then, seven bits of the secret data were read and divided into two parts of three bits and four bits. The 4-bit part is used for selecting one of 16 classes and the 3-bit part is used for selecting one of eight bit strings. Then to embed information, the LSB content of the 7 pixels from host image replaces with the selected bit string. The above process is repeated for the remaining secret data on the next pixels of host image to embed all secret data. The maximum length of data which can be embedded with method [30] into an image with a size of $H \times W$ is equal to $H \times W - 1$.

In extraction stage, a seven-bit sequence from the image pixels is read and the LSB content of each pixel is separated and is juxtaposed as a 7-bit string. This 7-bit string is multiplied by parity check matrix relating to (7, 4) Hamming code to obtain 3-bit syndrome vector. By combining this 3-bit vector and other 4 bits, the embedded 7 bits are extracted. This operation is repeated until all of the embedded bits are extracted.

One of the weaknesses of Chang's method [30] is its limited embedding payload (maximum $H \times W - 1$ bits) because data can be only embedded into the LSB of each pixel. On the other hand, (7, 4) Hamming code is only able to correct an error bit and this can reduce the efficiency of extraction module when encountering hard attacks. Our proposed method does not have any of the two limitations above. In fact, the proposed method while improves the robustness against destruction and maintains the visual quality (using modulus function), will not be faced with the restriction of the payload.

B. Embedding Low-Order Bits Using Modulus Function

The method [13] embeds the secret data into the low-order bits of host image pixels using modulus function. Assuming data are embedded in n low-order bits of each pixel, the data are decomposed into n -bit units. The decimal value of each of these units (x_i) will be in the range of 0 to $2^n - 1$. To embed the i -th unit (x_i) in i -th pixel of host image (y_i), first, the difference value is calculated from Eq. (1):

$$d_i = x_i - (y_i \bmod 2^n) \quad (1)$$

where $Z = X \bmod Y$ is the remainder of the division of X by Y . Then, the minimal difference of d_i is calculated from Eq. (2):

$$d'_i = \begin{cases} d_i & \text{if } -\left\lfloor \frac{2^n - 1}{2} \right\rfloor \leq d_i \leq \left\lfloor \frac{2^n - 1}{2} \right\rfloor \\ d_i + 2^n & \text{if } -2^n + 1 \leq d_i < -\left\lfloor \frac{2^n - 1}{2} \right\rfloor \\ d_i - 2^n & \text{if } \left\lfloor \frac{2^n - 1}{2} \right\rfloor < d_i < 2^n \end{cases} \quad (2)$$

where $\lfloor x \rfloor$ is the largest integer which does not exceed x and $\lceil x \rceil$ is an integer which is obtained from upward rounding of value x . Since value d'_i may be out of the range from 0 to 255 in some cases, the modified value of the i -th pixel of host image after embedding (\hat{y}_i) is calculated from Eq. (3):

$$\hat{y}_i = \begin{cases} y_i + d'_i & \text{if } 0 \leq y_i + d'_i \leq 255 \\ y_i + d'_i + 2^n & \text{if } y_i + d'_i < 0 \\ y_i + d'_i - 2^n & \text{if } y_i + d'_i > 255 \end{cases} \quad (3)$$

To extract the embedded data, it is enough to use the following equation:

$$x_i = \hat{y}_i \bmod 2^n \quad (4)$$

where n is the number of the low-order bits used for embedding, \hat{y}_i is the value of the i -th pixel from stego image and x_i is the value which is hidden into n low-order bits of the pixel \hat{y}_i .

In this paper after encoding, in order to improve the visual quality of stego image, we embed data using low-order bits substitution method which is proposed by Thein and Lin [13].

3- Reed-Muller Codes

In communication sciences, error correction code is an algorithm by which it is possible to detect and correct errors in the received data [32, 33]. A set of error correction codes is divided into two subsets of block codes and convolutional codes. Assuming that k is the length of the message and N is the length of the codeword and $N \geq k$, block codes divide the data into k -bit blocks to form $\mathbf{u}=(u_0, u_1, \dots, u_{k-1})$ and after encoding, a codeword $\mathbf{c}=(c_0, c_1, \dots, c_{N-1})$ is generated. Block codes have $N-k$ parity bit and are decoded as a block to block. Fig. 1 shows the structure of a binary block code.

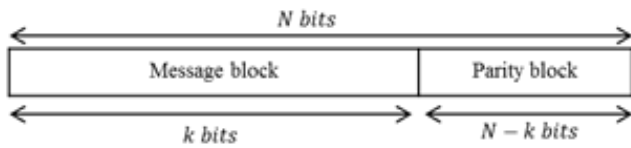


Fig. 1. Structure of a binary block code.

Reed-Muller code is usually shown as $RM(r,m)$ where r and m are positive integers and $0 \leq r \leq m$. In $RM(r,m)$, the length of message and codeword are calculated using equations (5) and (6), respectively:

$$k = \sum_{i=0}^r \binom{m}{i} \quad (5)$$

$$N = 2^m \quad (6)$$

The maximum number of error bits which this code is able to correct is equal to:

$$t = \left\lfloor \frac{2^{m-r} - 1}{2} \right\rfloor \quad (7)$$

In this paper, we will use Reed-Muller (R-M) codes for encoding secret data to be embedded in low-order bits of host image and then decoding low-order bits of stego image to extract hidden information. In the following, we describe R-M encoding and decoding method.

A. Reed-Muller Encoding

For a Reed-Muller code from r order and length of codeword $N=2^m$, generator matrix is defined as [33]:

$$G_{RM(r,m)} = \begin{bmatrix} \mathbf{v}_0 \\ \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_m \\ \mathbf{v}_1\mathbf{v}_2 \\ \mathbf{v}_1\mathbf{v}_3 \\ \vdots \\ \mathbf{v}_{m-1}\mathbf{v}_m \\ \mathbf{v}_1\mathbf{v}_2\mathbf{v}_3 \\ \vdots \\ \mathbf{v}_{m-r+1}\mathbf{v}_{m-r+2} \dots \mathbf{v}_m \end{bmatrix}_{k \times N} \quad (8)$$

where $\mathbf{v}_0 = \underbrace{11\dots 1}_{2^m}$, $\mathbf{v}_i = \underbrace{00\dots 0}_{2^{i-1}} \underbrace{11\dots 1}_{2^{i-1}} \underbrace{00\dots 0}_{2^{i-1}} \underbrace{11\dots 1}_{2^{i-1}} \dots$ for

$1 \leq i \leq m$ which \mathbf{v}_i has a length of equal to 2^m .

Assuming that we want to encode binary message \mathbf{u} , binary codeword \mathbf{c} is generated by binary multiplication operation, that is:

$$\mathbf{c} = \mathbf{u} * G_{RM(r,m)} \quad (9)$$

where the length of \mathbf{u} and \mathbf{c} is k and N , respectively.

B. Reed-Muller Decoding

One of the techniques for decoding Reed-Muller codes is known as multiple error correction decoding algorithm which was presented by Reed [33]. Reed's decoding process uses majority logic to determine the sent bit. Assuming a $RM(r,m)$ code with input message $\mathbf{u}=(u_0, u_1, \dots, u_m, u_{1,2}, \dots, u_{m-1,m}, \dots, u_{1,2,\dots,r}, \dots, u_{m-r+1,m-r+2}, \dots, u_m)$, the corresponding codeword will be as:

$$\begin{aligned} \mathbf{c} &= (c_0, c_1, \dots, c_{N-1}) \\ &= u_0 \mathbf{v}_0 + \sum_{1 \leq i_1 \leq m} u_{i_1} \mathbf{v}_{i_1} + \sum_{1 \leq i_1 < i_2 \leq m} u_{i_1} u_{i_2} \mathbf{v}_{i_1} \mathbf{v}_{i_2} \\ &+ \dots + \sum_{1 \leq i_1 < i_2 < \dots < i_r \leq m} u_{i_1 i_2 \dots i_r} \mathbf{v}_{i_1} \mathbf{v}_{i_2} \dots \mathbf{v}_{i_r} \end{aligned} \quad (10)$$

Assuming received vector $\mathbf{x}=(x_0, x_1, \dots, x_{N-1})$, decoding process includes $r+1$ stage. For $1 \leq i_1 < i_2 < \dots < i_{r-l} \leq m$ where $0 \leq l \leq r$, an index set is formed as:

$$S \triangleq \left\{ a_{i_1-1} 2^{i_1-1} + a_{i_2-1} 2^{i_2-1} + \dots + a_{i_{r-l}-1} 2^{i_{r-l}-1} \right. \\ \left. : a_{i_j-1} \in \{0,1\} \text{ for } 1 \leq j \leq r-l \right\} \quad (11)$$

which is a set of 2^{r-l} nonnegative integer members and values of less than 2^m . Consider E as a set of integer members of $\{0,1,\dots,m-1\}$ which are not member of $\{i_1-1,i_2-1,\dots,i_{r-l}-1\}$, i.e.:

$$E \triangleq \{0,1,\dots,m-1\} \setminus \{i_1-1,i_2-1,\dots,i_{r-l}-1\} \quad (12)$$

$$= \{j_1,j_2,\dots,j_{m-r+l}\}$$

where $0 \leq j_1 < j_2 < \dots < j_{m-r+l} \leq m-1$. Now, form a set of integers as:

$$S^c \triangleq \{d_{j_1} 2^{j_1} + d_{j_2} 2^{j_2} + \dots + d_{j_{m-r+l}} 2^{j_{m-r+l}}\} \quad (13)$$

$$: d_{j_t} \in \{0,1\} \text{ for } 1 \leq t \leq m-r+l$$

which has 2^{m-r+l} nonnegative integer members. Now, form a set of indices as follows for each $q \in S^c$:

$$B \triangleq q + S = \{q + s : s \in S\} \quad (14)$$

Then, decision equations in l -th decoding stage are obtained from Eq. (15):

$$A^{(l)} = \sum_{t \in B} x_t^{(l)} \quad (15)$$

The above relation follows binary addition rule and includes 2^{m-r+l} equations in each stage. If most of the results obtained from these decision equations are zero, input message $u_{i_1 i_2 \dots i_{r-l}}$ will be decoded as $u_{i_1 i_2 \dots i_{r-l}}^* = 0$, and if they are one, $u_{i_1 i_2 \dots i_{r-l}}$ will be decoded as $u_{i_1 i_2 \dots i_{r-l}}^* = 1$.

Assuming completion of the l -th stage of decoding process, form the modified received vector as:

$$\mathbf{x}^{(l)} \triangleq \mathbf{x}^{(l-1)} - \sum_{1 \leq i_1 < \dots < i_{r-l+1} \leq m} u_{i_1 i_2 \dots i_{r-l+1}}^* \mathbf{v}_{i_1} \mathbf{v}_{i_2} \dots \mathbf{v}_{i_{r-l+1}} \quad (16)$$

In the above relation, $\mathbf{x}^{(l-1)}$ is the modified received vector in l -th stage of decoding and $\mathbf{x}^{(0)} = \mathbf{x}$. Now, we go to the next stage and repeat the above process until the end of the $r+1$ stage to decode all received bits.

4- Proposed Method

In this section, we present a steganography scheme with high embedding payload which is robust against the attacks. The proposed method can compete with the similar methods in both cases. Since there is no need for accessibility of the original image to extract secret data, the proposed method is considered as a blind steganography technique. Overall, a block diagram of the proposed method is shown in Fig. 2. In the proposed method, we first divide the sequence of pixels of the host image as a block. Then, secret bit strings are encoded by Reed-Muller block codes. This encoding will give rise to the correction and retrieval of the data which have been destroyed. Secret data can include binary information obtained from an image, a text or any kind of data in binary format. For example, if the desired data for hiding are textual, it can be converted into binary data using ASCII codes. Since the effect of change in values of the first to eight bits is different in a grayscale image with 8-bit pixels (for example, change in value of the first bit creates smaller visual distortion compared with the change in value of the second bit), we will use different R-M codes with different parameters of r and m to encode information for hiding it into

pixel bits of digital image. After bits are encoded, we embed them by method [13] into the low-order bits of the host image. Since in this paper, we use more significant bits in addition to LSB for embedding data, the goal of applying embedding method of Thein and Lin [13] is to reduce Mean Square Error (MSE) between two images before and after embedding and as a result, we achieve higher PSNR. In data extraction stage, we first apply modulus operation on pixels of the received image and then divide and decode a sequence of the image pixels as a block having the secret key.

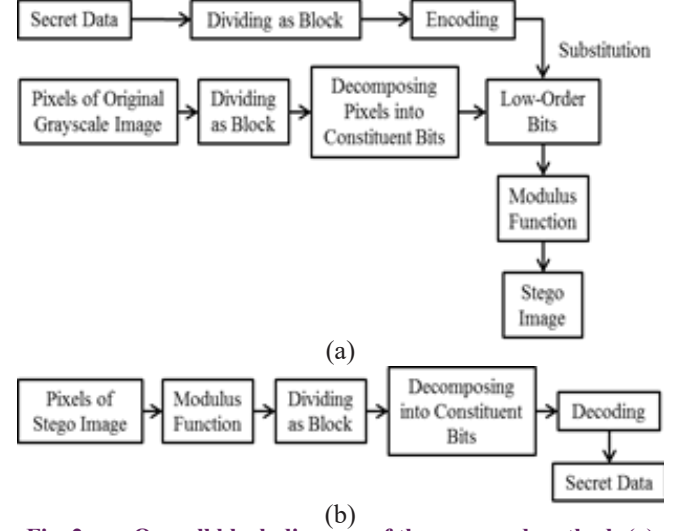


Fig. 2. an Overall block diagram of the proposed method: (a) embedding, (b) extraction.

A. Embedding Process

Assume that we want to hide secret data S into pixels of the grayscale image I with size of $H \times W$. n is the number of the low-order bits used for embedding secret information. The range of n is from 1 to 8 where $n=1$ is LSB. Assuming the use of block code $RM(r_j, m_j)$ for hiding information in j -th low-order bit (j -th bit of each pixel from the right side), maximum size of S is determined by the following relation:

$$\max_{size_S} = \sum_{j=1}^n \left\lfloor \frac{H \times W}{2^{m_j}} \right\rfloor \times k_j \quad (17)$$

$$\text{where } k_j \text{ is obtained from } k_j = \sum_{i=0}^{r_j} \binom{m_j}{i}.$$

First, we divide the sequence of bits S to k_j -bit blocks $u^{(q_j)} = (u_1^{(q_j)}, u_2^{(q_j)}, \dots, u_{k_j}^{(q_j)})$ where q_j is index of each k_j -bit unit of S for hiding into j -th low-order bit and $1 \leq q_j \leq \left\lfloor \frac{H \times W}{2^{m_j}} \right\rfloor$. Now, we encode each block by $RM(r_j, m_j)$ described in section 3-1. By doing so, we will have some encoded blocks with the length of 2^{m_j} as $c^{(q_j)} = (c_1^{(q_j)}, c_2^{(q_j)}, \dots, c_{2^{m_j}}^{(q_j)})$ which should be embedded into j -th low-order bit of pixels of the image I . In this regard, each block will have 2^{m_j} bits to be hidden in the image. To hide these binary blocks into j -th low-order bit, we should divide sequence from pixels of image $I(p_i, 1 \leq i \leq H \times W)$ into n_b the block $b^{(q_j)}$ with length of 2^{m_j} where $n_b = \left\lfloor \frac{H \times W}{2^{m_j}} \right\rfloor$ and $1 \leq q_j \leq n_b$. j -th low-order bit of pixels of each block $b^{(q_j)}$ forms a sequence of bits in the form of $P_{binary}^{(q_j)} = (P_{(q_j-1) \times 2^{m_j} + 1, j}, P_{(q_j-1) \times 2^{m_j} + 2, j}, \dots, P_{q_j \times 2^{m_j}, j})$ which are replaced with bits $c^{(q_j)}$. In fact, until this stage, we encoded j -th



Fig. 3. Encoding process for the first information block and modifying values of the first block ($q_j=1$) of host image: (a) generating the first information block, (b) generating the first host image block, (c) modifying.

low-order bit from pixels of the image by $RM(r_j, m_j)$. For example, Fig. 3 shows an encoding process for the first information block and modifying values of the first block ($q_j=1$) of the host image. After modifying values of all blocks of the host image, we consider modified the decimal form of the low-order bits from pixels of I as an image and we name this image S' . Values of S' will be in the range of 0 to 2^n-1 . Finally, we use the method explained in section (2-2) to embed information. Embedding algorithm is presented step by step as follows.

Data Embedding Process

Input: A grayscale image (I) with the size of $H \times W$ which is used as a host image, secret binary data S , the number of the used low-order bits (n), block code $RM(r_j, m_j)$ for encoding data of S to substitute j -th low-order bit of the host image blocks.

Output: A stego image (I) with the size of $H \times W$.

Step 1: scan pixels of image I ($p_i, 1 \leq i \leq H \times W$) from top to bottom and left to right.

Step 2: For embedding blocks in j -th low-order bit, divide sequence of p_i to n_b block $b^{(q_j)}$ by the length of 2^{m_j} where

$$n_b = \left\lfloor \frac{H \times W}{2^{m_j}} \right\rfloor \text{ and } 1 \leq q_j \leq n_b.$$

Step 3: Consider the initial value of q_j equal to 1.

Step 4: Decompose values of pixels of the block $b^{(q_j)}$ to its constituent bits. Take j -th low-order bit of each pixel and name sequence of these bits $p_{binary}^{(q_j)}$.

Step 5: Reed k_j next bit ($u^{(q_j)} = (u_1^{(q_j)}, u_2^{(q_j)}, \dots, u_{k_j}^{(q_j)})$) where $k_j = \sum_{i=0}^{r_j} \binom{m_j}{i}$.

Step 6: Encode a sequence of the bits obtained from the previous step ($u^{(q_j)}$) using encoder $RM(r_j, m_j)$ described in section 3-2, and name sequence of the encoded bits $c^{(q_j)}$.

Step 7: Modify bits of the block $p_{binary}^{(q_j)}$ to bits of the block $c^{(q_j)}$.

Step 8: Increase q_j to one unit and return to step 4 until $q_j < n_b$.

Step 9: Consider the modified decimal form of the low-order bits from pixels of I as an image S' .

Step 10: Considering a new image S' , compute d_i and d'_i from Eq. (1) and (2), respectively where $1 \leq i \leq H \times W$.

Step 11: Compute the modified value of gray level of i -th pixel of the image I after embedding process (\hat{y}_i) from Eq. (3) to obtain stego image.

B. Extraction Process

Since the designed algorithm is a blind steganography technique, here, we will not need original image. Assuming that receiver is aware of Reed-Muller code parameters (i.e. r_j and m_j), secret data will be extracted by the algorithm which is explained in the following. We name pixels of stego image \hat{y}_i where $i=1, 2, \dots, H \times W$. By applying Eq. (4) on pixels of received stego image, we first obtain the values which have been hidden into n low-order bits of pixel \hat{y}_i (i.e. x_i s). Now, the information is ready for being decoded by Reed decoder. To decode the information hidden in j -th low-order bit, we first divide modified pixels sequence of the image

(\hat{y}_i) to n_b block $b^{(q_j)}$ with length of 2^{m_j} where $n_b = \left\lfloor \frac{H \times W}{2^{m_j}} \right\rfloor$ and $1 \leq q_j \leq n_b$. After decomposing pixels of each block to its 8 constituent bits, we pick up j -th low-order bit from pixels of each block $b^{(q_j)}$ and form a sequence of bits in the form of

$\mathbf{x}^{(q_j)} = \left(x_{1,j}^{(q_j)}, x_{2,j}^{(q_j)}, x_{3,j}^{(q_j)}, \dots, x_{2^{m_j},j}^{(q_j)} \right)$ Each block $\mathbf{x}^{(q_j)}$, after decoding process by the method explained in section 3-3, generates a k_j -bit sequence in the form of $\mathbf{u}^{*(q_j)} = \left(u_1^{*(q_j)}, u_2^{*(q_j)}, \dots, u_{k_j}^{*(q_j)} \right)$ which is obtained from the relation $k_j = \sum_{i=0}^{r_j} \binom{m_j}{i}$. Fig. 4,

for example, shows decoding process of the first modified block and extraction of its first secret block ($q_j=1$). Extraction algorithm is presented step by step as follows.

Data Extraction Process

Input: A stego image (I) with the size of $H \times W$, the number of the used low-order bits (n), block code $RM(r_j, m_j)$ for decoding secret data embedded in blocks of the host image.

Output: Secret data.

Step 1: For pixels of image I ($i=1, 2, \dots, H \times W$), apply Eq. (4) to obtain x_i (the hidden value into n -th low-order bit of pixel y_i).

Step 2: For decoding information available in the j -th low-order bit, divide values sequence of x_i into n_b blocks $b^{(q_j)}$ by a length of 2^{m_j} where $n_b = \left\lfloor \frac{H \times W}{2^{m_j}} \right\rfloor$ and $1 \leq q_j \leq n_b$.

Step 3: Consider the initial value of q_j equal to 1.

Step 4: Decompose values of the block $b^{(q_j)}$ to its constituent bits. Pick up the j -th low-order bit of each block and name sequence of these bits $\mathbf{x}^{(q_j)}$.

Step 5: Decode sequence of the bits obtained from the previous step using Reed decoder which was described in section 3-3 and name k_j -bit decoded sequence $\mathbf{u}^{*(q_j)}$.

Step 6: Increase q_j to one unit. Until information of all blocks is extracted, return to step (4).

5- Simulation Results And Discussion

To evaluate the performance of the proposed method, we considered nine grayscale images with a size of 512×512 as test images and compared our method with other methods by three quantitative measures.

Visual quality is usually evaluated by PSNR criterion which shows a degree of difference between original image and stego image in dB and calculated from the following relation:

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \quad (18)$$

In the above relation, MSE indicates the difference between values of pixels in the original image and stego image and is defined as $MSE = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W (I_{ij} - I'_{ij})^2$ In this relation, H and W are length and height of the image, respectively. I_{ij} are pixels value of the original image and I'_{ij} are pixels value of stego image. The lower the difference between two images, the lower the MSE value and as a result, the higher the PSNR value which is more desirable for goals of steganography.

The second criterion is embedding payload which is the ratio of the number of bits embedded in host image to the number of pixels of host image and is defined as:

$$P = \frac{|S|}{H \times W} \quad (19)$$

where $|S|$ refers to the number of secret bits which are carried by host image. The payload is measured in bits-per-pixel (bpp). The third criterion is error correction which enables the user to retrieve the secret information which has been destroyed in case of intentional or unintentional destruction of stego image. Error correction capacity is defined as:

$$C = \frac{b_c}{b_e} \quad (20)$$

where b_c and b_e are the number of correct bits and the number of error bits after extraction process, respectively.

A change in more significant bits causes more visual distortion and it is necessary that information extraction module in more significant bits have a better error correction performance than that in less significant bits. Therefore, we used $RM(1,3)$ and $RM(2,5)$ for embedding information into the first and second bits of pixels from the right side, respectively which are able to correct 3 and 7 error bits considering Eq. (7). The secret data type used in the simulations has been textual that we convert it into binary data using ASCII codes, initially. Each one of the test images carries 393216 bits of information. Fig. 5 shows some visual results of simulations by the proposed method. In the left column, original images are observed. Middle and right columns show the images embedded by the proposed method and images destroyed by different attacks, respectively. In Table 1, a comparison between the proposed method and the other techniques is performed in terms of

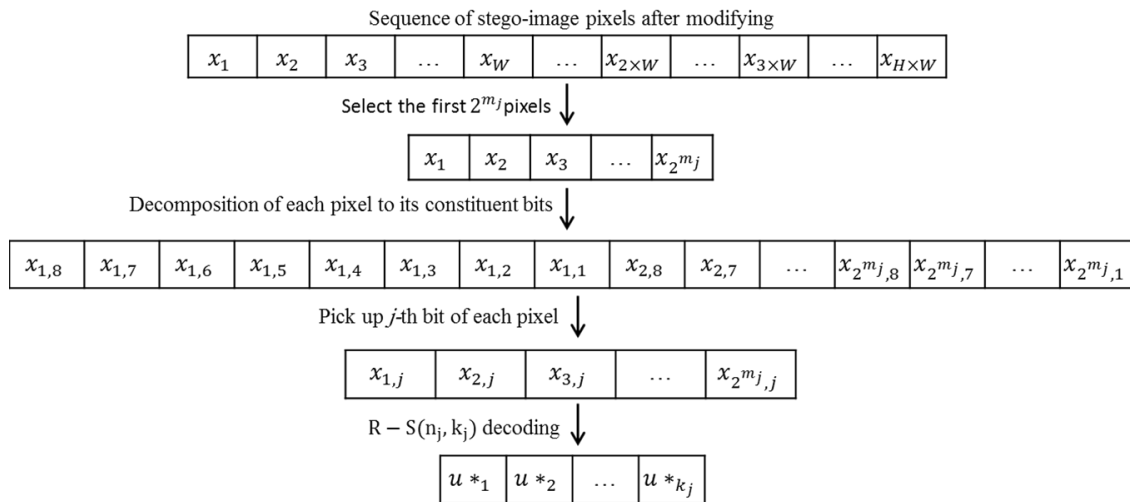


Fig. 4. Decoding process of the first modified block and extraction of its first secret block ($q_j=1$).

Table 1. A comparison between the proposed method and the other techniques in terms of visual quality and embedding payload.













Test Images		The method [30]		The method [18] (r=5)		The method [18] (r=6)		The method HClH [19]		The method HQIH [19]		The method [20]		The method SOS [21]		The proposed method	
Test Image	Image Name	PSNR (dB)	P (bpp)	PSNR (dB)	P (bpp)	PSNR (dB)	P (bpp)	PSNR (dB)	P (bpp)	PSNR (dB)	P (bpp)	PSNR (dB)	P (bpp)	PSNR (dB)	P (bpp)	PSNR (dB)	P (bpp)
	Jet (F16)	50.77	0.99	45.25	1.97	51.15	1.00	46.01	0.75	52.10	0.83	46.87	0.64	49.88	1.58	48.24	1.50
	Lena	50.84	0.99	45.14	1.99	51.14	1.00	47.02	0.75	52.09	0.83	46.86	0.64	49.89	1.58	48.12	1.50
	Pepper	50.12	0.99	45.77	1.80	51.15	1.00	45.12	0.75	52.10	0.83	46.86	0.64	49.88	1.58	48.15	1.50
	Zellda	51.14	0.99	45.13	2.00	51.14	1.00	45.38	0.75	52.10	0.83	46.85	0.64	49.88	1.58	48.13	1.50
Average		50.72	0.99	45.32	1.94	51.15	1.00	45.88	0.75	52.10	0.83	46.86	0.64	49.88	1.58	48.16	1.50

Table 2. Comparison results between the proposed method and the method [1]

Image	Image name	The method [30]				The proposed method						
		PSNR	P	Attack	PSNR (Destroyed)	C	PSNR	P	Attack	PSNR (Destroyed)	C	
	Baboon	50.96	0.99	Gaussian noise: mean 0 & variance 0.01	20.05	1.00	48.13	1.50	Gaussian noise: mean 0 & variance 0.01	20.03	1.20	
	Barbara	50.36	0.99	Gaussian noise: mean 0 & variance 0.1	11.45	1.01	48.14	1.50	Gaussian noise: mean 0 & variance 0.1	11.48	1.21	
	Boats	50.39	0.99	Salt & Pepper noise: density 0.05	18.53	17.49	48.14	1.50	Salt & Pepper noise: density 0.05	18.52	198.10	
	Goldhill	51.09	0.99	Salt & Pepper noise: density 0.5	8.12	1.91	48.14	1.50	Salt & Pepper noise: density 0.5	8.12	2.15	
	Lena	50.84	0.99	Speckle noise: variance 0.04	19.77	0.99	48.12	1.50	Speckle noise: variance 0.04	19.74	1.22	
	Pepper	50.12	0.99	Speckle noise: variance 0.4	10.80	0.99	48.15	1.50	Speckle noise: variance 0.4	10.77	1.19	
	Tiffany	51.15	0.99	Median filter	31.06	1.41	48.14	1.50	Median filter	31.04	1.47	
	Zellda	51.14	0.99	Gaussian low-pass filter: sigma 0.9	37.30	0.99	48.13	1.50	Gaussian low-pass filter: sigma 0.9	37.28	1.21	
Average		50.76	0.99		19.63	3.22	48.14	1.50		19.62	25.97	

visual quality and embedding payload in stego image. The maximum payload in the methods of [30], HClH [19] and HQIH [19] is 1 bit per pixel, while the proposed method is capable of carrying data with a payload greater than 1 bit per pixel, with preserving visual quality. In a compromise between payload and visual quality, the proposed method is competitive with methods [18] and [20]. Although method [21] shows better results, it is not robust against the attacks. Comparison results between the proposed method and the method [30] are given in Table 2. The results show the proposed method guarantees PSNR of above 48.1 dB while the embedding payload is increased by 50% compared with method [30]. Also, the proposed method has increased error correction capacity against all kinds of noises with different intensities and statistical characteristics. We also compared the robustness of the proposed algorithm, algorithm [31]

and algorithm [30] against adding salt and pepper noise with different densities. The obtained results are listed in Table 3. As can be seen, the proposed method shows a great robustness compared with the other two methods.

6- Conclusions

A blind steganography scheme with a high embedding payload was presented. Although our method is able to embed information into more significant bits, it preserves visual quality of image well. The obtained results from the simulations were compared with studies [18], [19], [20], [21], [30] and [31]. While we increased embedding payload, it led to PSNR of more than 48.1 dB. In addition, we increased robustness against noise and filtering attacks using Reed-Muller error correction code and as a result of increasing error correction capacity.

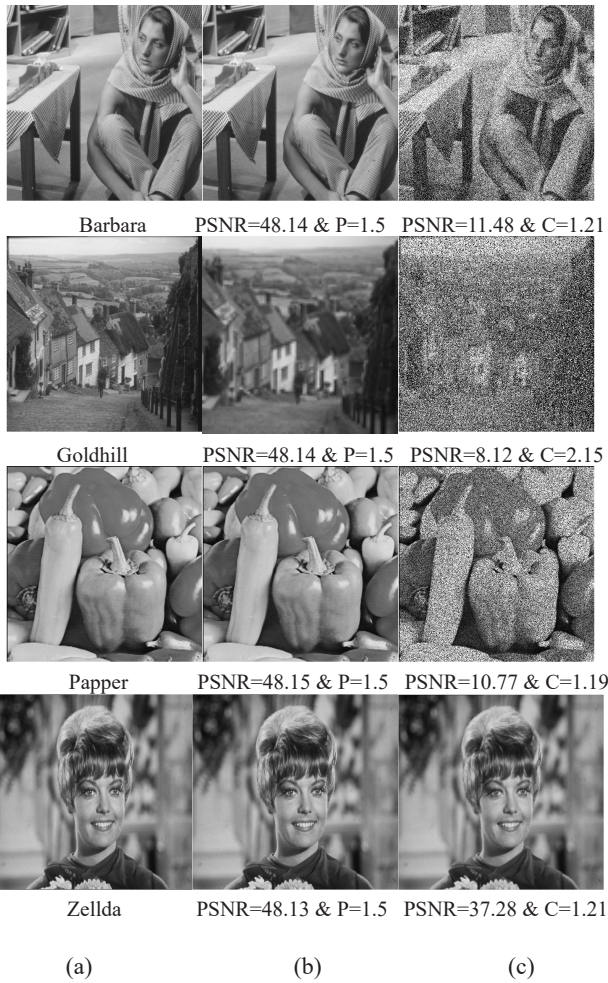
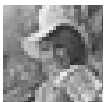



Fig. 5. Some visual results of simulations by the proposed method: (a) host images, (b) stego images with payload 1.5 bpp and (c) stego images destroyed by various attacks (from top to bottom, respectively: Gaussian noise with mean 0 & variance 0.1, Salt & Pepper noise with density 0.5, Speckle noise with variance 0.4 and Gaussian low-pass filter with sigma 0.9).

Table 3. A comparison between the proposed method and the method [31] and [30] in terms of robustness against adding salt and pepper noise with different densities.

Image	Noise Density	The method [31]	The method [30]	The proposed method
		C	C	C
 Girl (Elaine)	0.005	26.59	173.99	12287
	0.007	18.40	125.21	8191
	0.01	13.45	88.62	4467
	0.02	8.23	43.00	1585
 Lena	0.005	19.22	185.97	24576
	0.007	15.09	128.90	9829
	0.01	12.76	85.06	8191
	0.02	6.90	42.98	2047

References

- [1] W. Zeng, Digital watermarking and data hiding: technologies and applications, in: Proc. Int. Conf. Inf. Syst. Anal. Synth, 1998, pp. 223-229.
- [2] M. Wu, B. Liu, Data hiding in image and video. I. Fundamental issues and solutions, Image Processing, IEEE Transactions on, 12(6) (2003) 685-695.
- [3] M. Wu, H. Yu, B. Liu, Data hiding in image and video. II. Designs and applications, Image Processing, IEEE Transactions on, 12(6) (2003) 696-705.
- [4] C.-T. Wang, H.-F. Yu, A Markov-based reversible data hiding method based on histogram shifting, Journal of Visual Communication and Image Representation, 23(5) (2012) 798-811.
- [5] X. Zhang, S. Wang, Vulnerability of pixel-value differencing steganography to histogram analysis and modification for enhanced security, Pattern Recognition Letters, 25(3) (2004) 331-339.
- [6] H.V. Desai, Steganography, Cryptography, Watermarking: A Comparative Study, Journal of Global Research in Computer Science, 3(12) (2013) 33-35.
- [7] J. Dittmann, P. Wohlmacher, K. Nahrstedt, Using cryptographic and watermarking algorithms, Multimedia, IEEE, 8(4) (2001) 54-65.
- [8] A. Abbass, E. Soleit, S. Ghoniemy, Blind video data hiding using integer wavelet transforms, Ubiquitous Computing and Communication Journal, (2007).
- [9] Y. Wang, A. Pearmain, Blind image data hiding based on self reference, Pattern Recognition Letters, 25(15) (2004) 1681-1689.
- [10] X.-Y. Luo, D.-S. Wang, P. Wang, F.-L. Liu, A review on blind detection for image steganography, Signal Processing, 88(9) (2008) 2138-2157.
- [11] H. Yang, X. Sun, G. Sun, A high-capacity image data hiding scheme using adaptive LSB substitution, Radioengineering, 18(4) (2009) 509.
- [12] R.-Z. Wang, C.-F. Lin, J.-C. Lin, Image hiding by optimal LSB substitution and genetic algorithm, Pattern recognition, 34(3) (2001) 671-683.
- [13] C.-C. Thien, J.-C. Lin, A simple and high-hiding capacity method for hiding digit-by-digit data in images based on modulus function, Pattern Recognition, 36(12) (2003) 2875-2881.
- [14] C.-C. Chang, C.-S. Chan, Y.-H. Fan, Image hiding scheme with modulus function and dynamic programming strategy on partitioned pixels, Pattern Recognition, 39(6) (2006) 1155-1167.
- [15] A.D. Ker, Steganalysis of LSB matching in grayscale images, Signal Processing Letters, IEEE, 12(6) (2005) 441-444.
- [16] J. Mielikainen, LSB matching revisited, Signal Processing Letters, IEEE, 13(5) (2006) 285-287.
- [17] D.-C. Wu, W.-H. Tsai, A steganographic method for images by pixel-value differencing, Pattern Recognition Letters, 24(9) (2003) 1613-1626.
- [18] H. Yang, X. Sun, G. Sun, A High-Capacity Image Data Hiding Scheme Using Adaptive LSB Substitution, Radioengineering, 18(4) (2009).

- [19] Y. Zhang, J. Jiang, Y. Zha, H. Zhang, S. Zhao, Research on Embedding Capacity and Efficiency of Information Hiding Based on Digital Images, *International Journal of Intelligence Science*, 3 (2013) 77.
- [20] B. Jena, High payload digital image steganography using mixed edge detection mechanism, 2014.
- [21] Z.-X. Yin, C.-C. Chang, Q. Xu, B. Luo, Second-order steganographic method based on adaptive reference matrix, *IET Image Processing*, 9(4) (2015) 300-305.
- [22] X. Zhang, S. Wang, Efficient steganographic embedding by exploiting modification direction, *IEEE Communications Letters*, 10(11) (2006) 781-783.
- [23] C.-C. Chang, Y.-C. Chou, T.D. Kieu, An information hiding scheme using Sudoku, in: *Innovative Computing Information and Control*, 2008. ICICIC'08. 3rd International Conference on, IEEE, 2008, pp. 17-17.
- [24] W. Hong, T.-S. Chen, C.-W. Shiu, A minimal Euclidean distance searching technique for Sudoku steganography, in: *2008 International Symposium on Information Science and Engineering*, IEEE, 2008, pp. 515-518.
- [25] R.-M. Chao, H.-C. Wu, C.-C. Lee, Y.-P. Chu, A novel image data hiding scheme with diamond encoding, *EURASIP Journal on Information Security*, 2009(1) (2009) 1.
- [26] Z. Yin, B. Luo, MDE-based image steganography with large embedding capacity, *Security and Communication Networks*, (2015).
- [27] R. Crandall, Some notes on steganography, Posted on steganography mailing list, (1998).
- [28] A. Westfeld, F5—a steganographic algorithm, in: *Information hiding*, Springer, 2001, pp. 289-302.
- [29] W. Zhang, S. Wang, X. Zhang, Improving embedding efficiency of covering codes for applications in steganography, *Communications Letters, IEEE*, 11(8) (2007) 680-682.
- [30] C.-C. Chang, T.D. Kieu, Y.-C. Chou, A high payload steganographic scheme based on (7, 4) hamming code for digital images, in: *Electronic Commerce and Security, 2008 International Symposium on*, IEEE, 2008, pp. 16-21.
- [31] S. Singh, T.J. Siddiqui, A Security Enhanced Robust Steganography Algorithm for Data Hiding, *International Journal of Computer Science Issues (IJCSI)*, 9(3) (2012).
- [32] S.B. Wicker, *Error control systems for digital communication and storage*, Prentice hall Englewood Cliffs, 1995.
- [33] L. Shu, S. Lin, D.J. Costello, *Error control coding*, Pearson Education India, 2004.

Please cite this article using:

A. M. Molaei, M. H. Sedaaghi, H. Ebrahimnezhad, "Steganography Scheme Based on Reed-Muller Code with Improving Payload and Ability to Retrieval of Destroyed Data for Digital Images", *AUT J. El ec. Eng.*, 49(1) (2017)53-62.

DOI: 10.22060/ej.2016.814

