

# Design and Analysis of Reed Solomon Codes for Reversible Data Hiding in Encrypted Images

<sup>1</sup>Parminster Kaur, <sup>2</sup>Amandeep Kaur

<sup>1,2</sup>Dept. of Electronics and Communication Engineering, Punjabi University, Patiala, India

## Abstract

In this paper, Reversible data hiding is proposed by using Reed Solomon codes for increasing embedding capacity of message in encrypted image. Reversible Data hiding is a type of data hiding techniques in which the host image can be recovered exactly. Message is encode into codewords by using RS codes and then these codewords can be embedded into encrypted image by using encryption key. After embedding data the received encrypted image is firstly decrypted by using encryption key and then, receiver uses data-hiding key in order to extract the codewords of RS codes containing messages. Simulation results of two images show that the embedding capacity of the proposed method is better than reference method. The PSNR of the proposed method is 44db. The purpose of this paper is to study and analyze the performance and efficiency of Reed-Solomon (RS) Codes in Reversible data hiding.

## Keywords

Data Hiding, Image Encryption, Image Decryption, Reed Solomon (RS) Codes

## I. Introduction

In communication system, during transmission information or data can be corrupted due to noise or errors and may be access by unauthorized recipient. Nowadays security of the information is one of the important factors in data transmission. Reversible data hiding technique is secure way to embedded information into a cover/digital media and recovery of original image without any loss after extraction of information. Many reversible data hiding technique has been applied. The purpose of hiding the data is to provide the secret communication system.

In 1960, Irving Reed and Gus Solomon invented a new class of error-correcting codes that are called as Reed-Solomon (RS) codes [1]. Reed Solomon codes are subclass of non binary BCH codes. RS codes processes group of bits instead of one bit at a time. RS codes operate on the information by dividing the message stream into blocks of data. Then it adds redundancy bits as per block depending only on the current inputs which is together called as codeword. The symbols are elements of a finite field or Galois Field (GF). Galois field is used for encoding and decoding of Reed Solomon codes. After that, encoding is achieved by adding the remainder of a GF polynomial division into the message. Linear feedback Shift Register (LFSR) technique is used for this division method [2]. At the decoder, the syndrome calculation of the received codeword is carried out. Then we find error locations using Chien algorithm and error magnitudes using Forney algorithm. Further Berlekamp-Massey is used to calculate coefficients of error locator polynomial for error locations and coefficients of error evaluator polynomial for error values.

In this paper, increases the capacity of reversible data hiding in encrypted images is proposed by using the Reed Solomon codes. The proposed data hiding system consists of the three parts which are image encryption, data embedding, and data extraction. The messages are not embedded directly into encrypted image,

codewords of RS codes are embedded. With the feature of error correcting capability of RS codes, the extraction performance of message in a receiver becomes better in comparison with the performance of data embedding directly. Performance results in two images and RS codes show that our proposed scheme can be efficiently recovered the embedded data in reversible data hiding systems.

## II. Proposed Scheme

The proposed data hiding scheme is shown in fig. 1. The proposed scheme consists of three phases, which are encryption image, data hiding, and recovery image & data extraction. In encryption part, the original image is encrypted by the encryption key which generates pseudo-random values. Then the codewords are made from messages in encoders of RS codes and then codewords of RS codes are embedded into the encrypted image by using data-hiding key. Then, the encrypted image after embedded the RS codes is firstly decrypted by encryption key and the decrypted image is similar to the original image. Then, receiver uses the data-hiding key in order to extract the codewords of RS codes. To estimate message successfully, BM algorithm among decoding of RS codes is considered.

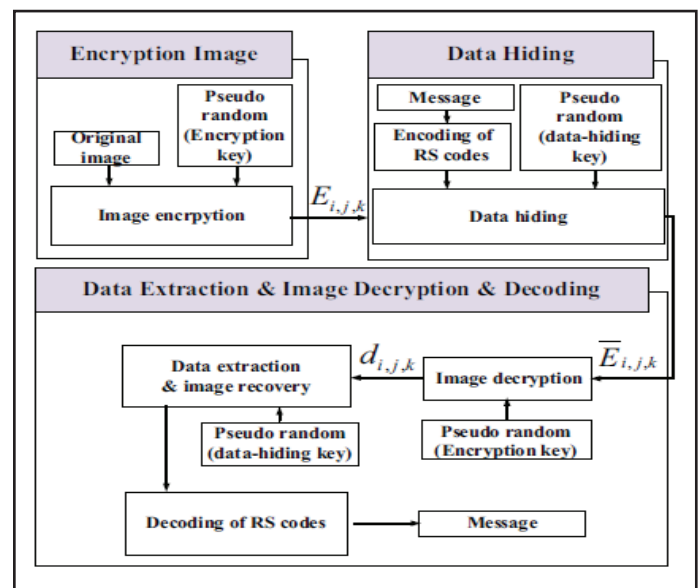


Fig. 1: Block Diagram of the Proposed Data-Hiding System

## A. Image Encryption

It is assumed that the original image consists of only each pixel with gray value that is represented by 8 bits. The bits of a pixel are denoted as  $g_{i,j,k}$  where  $(i, j)$ , and  $k$  mean the pixel Position and  $k$ -th bit in 8 bits, respectively. The gray value in  $(i, j)$  fixel is denoted as  $p_{ij}$ . The relationship between bits in a pixel and gray values are given as,

$$g_{i,j,k} = \left[ \frac{p_{ij}}{2^k} \right] \bmod 2, (0 \leq k \leq 7) \quad (1)$$

And

$$p_{i,j} = \sum_{k=0}^7 g_{i,j,k} \cdot 2^k \quad (2)$$

In an encryption part, encrypted images are made by using encryption key which generates pseudo-random values. The encrypted pixel value are made by exclusive-or operation as

$$E_{i,j,k} = g_{i,j,k} \oplus r_{i,j,k}, \quad (3)$$

Where  $r_{i,j,k}$  are the pseudo-random value made from the encryption key [6].

### B. Data Hiding

Firstly a data hider encodes message by using encoders of RS codes. Generally RS codes [3] are defined in Galois Field, GF ( $2^q$ ), where  $q$  is a positive integer. The RS Codes are represented as  $(n, k)$  RS codes with length  $n$  and dimension  $k$ , are defined as  $n=2^q-1$ ,  $t=(n-k)/2$ , where  $t$  denotes the maximum number of errors which can be correct by using RS decoder.

Since the RS codes are a class of cyclic codes, the message and codeword can be expressed as a message polynomial  $u(X)$  and a codeword polynomial  $c(X)$ , respectively.

$$u(X) = u_0 + u_1X + \dots + u_{k-1}X^{k-1}, \quad (4)$$

$$c(X) = c_0 + c_1X + \dots + c_{n-1}X^{n-1}, \quad (5)$$

where  $u_i \in (2^q)$  for  $i=0,1,\dots,k-1$  and  $c_j \in (2^q)$  for  $j=0,1,\dots,n-1$ . For up to  $t$ -error correction, the generator polynomial is defined as

$$g(X) = (X - \alpha)(X - \alpha^2) \dots (X - \alpha^{2t}) \\ = g_0 + g_1X + \dots + g_{2t}X^{2t} \quad (6)$$

Where  $\alpha$  is a primitive element of GF ( $2^q$ ). To make systematic codewords of RS codes, encoding can be defined as

$$c(X) = p(X) + X^{2t}u(X), \quad (7)$$

Where  $p(X)$  is the parity polynomial with degree  $< 2t$ . The  $p(X)$  can be calculated as a remainder when  $g(X)$  is divided into  $X^{2t}u(X)$ . The division algorithm of systematic encoding process of RS codes is shown in Fig. 2.

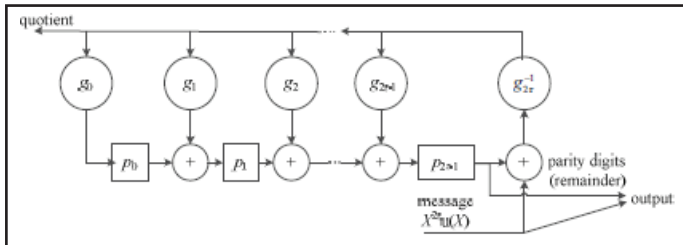


Fig. 2: Division Algorithm for Systematic Encoding of RS Codes

The data-hider can embed code words of RS codes into the encrypted image by using some procedures, but he does not need to know the original image. Firstly, the encryption image is divided into a number of non-overlapping blocks sized by  $s \times s$ , where  $s$  means a number of pixels. In the block, the encryption bits  $E_{i,j,k}$  which satisfy  $(m-1) \cdot s \leq i \leq m \cdot s$ ,  $(n-1) \cdot s \leq j \leq n \cdot s$ , and  $0 \leq k \leq 1$  exist within the same block where  $n, m$  are positive integers. Then, each block will include two message bits.

For each block, using values generated by using data-hiding key, the  $s^2$  pixels into four sets  $S_0, S_1, S_2, S_3$  are divided. The probability that a pixel belongs to four sets is uniform distribution. If the message bit is 00, the three Least Significant Bits (LSB) of each encrypted pixel in  $S_0$  are flipped.

$$\overline{E_{i,j,k}} = \overline{E_{i,j,2}E_{i,j,1}E_{i,j,0}}(i,j) \in S_0 \quad (8)$$

If the message bit is 01, the 3 LSB of each encrypted pixel in  $S_1$  are flipped.

$$\overline{E_{i,j,k}} = \overline{E_{i,j,2}E_{i,j,1}E_{i,j,0}}(i,j) \in S_1 \quad (9)$$

If the message bit is 10, the 3 LSB of each encrypted pixel in  $S_2$  are flipped.

$$\overline{E_{i,j,k}} = \overline{E_{i,j,2}E_{i,j,1}E_{i,j,0}}(i,j) \in S_2 \quad (10)$$

If the message bit is 11, the 3 LSB of each encrypted pixel in  $S_3$  are flipped.

$$\overline{E_{i,j,k}} = \overline{E_{i,j,2}E_{i,j,1}E_{i,j,0}}(i,j) \in S_3 \quad (11)$$

### C. Data Extraction and Image Recovery

To extract messages,  $r_{i,j,k}$  is generated by the encryption key, and the decryption image is made by result of exclusiveor of received data. The decryption bits is denoted as  $d_{i,j,k}$ . The original five Most Significant Bits (MSB) are extracted correctly because the five MSBs are not flipped. For a selected pixel, if embedded bits in block is 00 and the pixel belongs to  $S_1, S_2, S_3$ , or if embedded bits is 01 and the pixel belongs to  $S_0, S_2, S_3$ , if embedded bits in block is 10 and the pixel belongs to  $S_0, S_1, S_3$ , or if embedded bit is 11 and the pixel belongs to  $S_0, S_1, S_2$ , the encrypted bits are same as the original bit. On the other hand, if the embedded bit is 00 and the pixel belongs to  $S_0$ , if the embedded bit is 01 and the pixel belongs to  $S_1$  if the embedded bit is 10 and the pixel belongs to  $S_2$ , if the embedded bit is 11 and the pixel belongs to  $S_3$

$$d_{i,j,k} = r_{i,j,k} \oplus \overline{E_{i,j,k}} \\ = r_{i,j,k} \oplus \overline{g_{i,j,k}} \oplus r_{i,j,k} \\ = \overline{g_{i,j,k}}(k=0,1,2) \quad (12)$$

This Eq. (12) shows that the three decrypted LSB must be different from the original LSB.

After decrypting image, the image is segmented into blocks by using data hiding key. The block is also divided into  $s^2$  pixels which are divided four sets,  $S_0, S_1, S_2, S_3$  in the same way. For each decrypted block, all the three LSB in  $S_0$  is flipped, and all the three LSB in  $S_0$  is flipped to make new four blocks. The new blocks are denoted as  $H_0, H_1, H_2, H_3$ . Either  $H_0, H_1, H_2, H_3$  must be the original block and in another once all three LSB flipped. Extracting the RS code, we define a function which measures the fluctuation in  $H_0, H_1, H_2, H_3$ .

$$f = \sum_{i=2}^{s-1} \sum_{j=2}^{s-1} |p_{i,j} - \left( \frac{p_{i-1,j} + p_{i,j-1} + p_{i+1,j} + p_{i,j+1}}{4} \right)| \quad (13)$$

The result value of the Eq. (13) is consisted of four values which are results when using  $H_0, H_1, H_2, H_3$ . We denote the value using  $H_0, H_1, H_2, H_3$  as  $f_0, f_1, f_2, f_3$  respectively. The result of original block is generally lower than that of all three LSB flipped due to the spatial correlation in natural image. Therefore, by comparing

$f_0, f_1, f_2, f_3$  data is extracted. If  $f_0 < (f_1, f_2, f_3)$ ,  $H_0$  is regarded of the original block and then hidden data is 00. If  $f_1 < (f_0, f_2, f_3)$ ,  $H_1$  is regarded of the original block, and hidden data is 01. If  $f_2 < (f_0, f_1, f_3)$ ,  $H_2$  is regarded of the original block, and then hidden data is 10. If  $f_3 < (f_0, f_1, f_2)$ ,  $H_3$  is regarded of the original block, and hidden data is 11. Therefore, metric  $c_i$  to estimate codeword bit of RS codes is written as

$$c_i = \begin{cases} 00, & \text{if } f_0 < (f_1, f_2, f_3) \\ 01, & \text{if } f_1 < (f_0, f_2, f_3) \\ 10, & \text{if } f_2 < (f_0, f_1, f_3) \\ 11, & \text{if } f_3 < (f_0, f_1, f_2) \end{cases} \quad (14)$$

With the all data bits, the data is divided into the size of RS code. Then, the RS codes are fully formed for the blocks and decode the RS Codes by using BM algorithm [4], which is efficient to find error location polynomial based Linear Feedback Shift Register (LFSR). Then, error location and error value can be calculated easily. The process is shown bellow. The extracted codeword is denoted as

$$c' = c + e \quad (15)$$

To calculate syndromes,

$$\begin{aligned} S_i &= c(\alpha^i) + e(\alpha^i) \\ &= e(\alpha^i) \\ &= e_0 + e_1 X + \dots + e_{n-1} X^{n-1} \end{aligned} \quad (16)$$

since  $c(X) = 0$  for  $X = \alpha^i$   $i = 1, 2, \dots, 2t$ . Let the error location polynomial as

$$\Lambda(X) = \Lambda_0 + \Lambda_1 X + \dots + \Lambda_{n-1} X^{n-1} \quad (17)$$

There existed relation between syndrome coefficients of error location polynomial as:

$$S_i = -\sum_{j=1}^v \Lambda_j S_{i-j}, \quad (18)$$

Where  $v < t$ . BM using linear feedback shift register (LFSR) to find error location polynomial of smallest degree which satisfies (18), then error locations can be found by solving the roots. Forney's algorithm [5] utilizes each error location to find corresponding error value.

### III. Experiment Results

Lena and Jet images in figure 3 are used in this simulation which are of size  $512 \times 512$  since they are available for free usage. (15,7) RS code is used among the RS codes in each image since the recovery capability needs to be investigated in the proposed data hiding system.



Fig. 3: Images of Lena and Jet which are used for Simulation  
In fig. 4 and fig. 5 shows the performance of the proposed data

hiding method is better than of the reference method. In fig. 4 and fig. 5 Blue color bar graph shows the performances of reference and red color bar graph shows the performance of proposed method. We investigate the performance by considered the smallest size of  $s$ , PSNR also become well in two images.

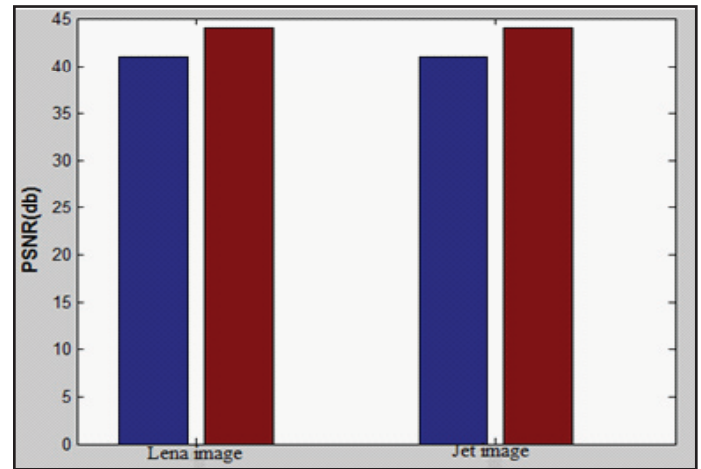


Fig. 4: PSNR Comparison of Two Images According to  $s$  in Data Hiding Systems

The parameter that can be used to measure the quality of the encrypted image is PSNR. The peak signal noise rate (PSNR) [7] is used to estimate the quality between the original image and the recovered image, which denote as  $I$  and  $E$  respectively. To obtain the PSNR of the encrypted image

$$\text{PSNR} = 10 \log_{10} \left( \frac{255^2}{\text{MSE}} \right) \quad (19)$$

Where, Mean square error (MSE)

$$\text{MSE} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - E(i, j)]^2 \quad (20)$$

Where,  $I(i, j)$  is the original image  
 $E(i, j)$  is the recovered image

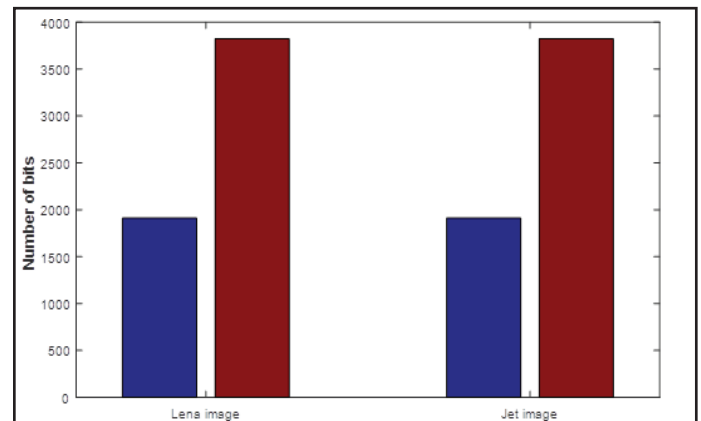


Fig. 5: Shows the Message Embedding Capacity of Two Images in Data Hiding Systems

To evaluate the performance of proposed work is comparing with existed work, by considering the smallest size of  $s$  when there is no bit error occurs. To obtain the size of message in the encrypted image for the  $s$ , 'Size of message', in table is defined as

$$2 * \left[ \left( \left\lceil \frac{512}{s} \right\rceil \right)^2 \cdot \frac{k}{n} \right], \quad (21)$$

Table 1: Simulation results of Lena and Jet images

	Lena		Jet	
	Ref.	(15,7)	Ref.	(15,7)
Code rate	0.47	0.47	0.47	0.47
S	8	8	8	8
Size of message	1911	3822	1911	3822
Gain (%)	*	100%	*	100%
PSNR	41	44	41	44

It can be seen in Table 1, capacity of embedding message in data hiding system by using RS codes is double than the reference system. Also recovers the original image without distortion after extraction of data.

#### IV. Conclusion

In this paper, we increase capacity of embedding message for reversible data hiding in encrypted image without distortion is proposed by using RS codes. The proposed method works in three parts: image encryption, data embedding and data extraction and image recovery. In image encryption part, the original image is encrypted by using encryption key which generates the pseudo-random values. Then data hider embeds the codewords of RS codes into encrypted image. Receivers firstly decrypt the encrypted image by using same encryption key. Then extract the codewords from the decrypted image and decode them in order to get the message. Simulation results show that the performance of the proposed method in reversible data hiding system shows high embedding capacity and good visual quality of recovered image.

#### References

- [1] I. S. Reed, G. Solomon, "Polynomial Codes Over Certain Finite Fields", Journal of the Society of Industrial and Applied Mathematics, pp. 300-304, 1960 printed in U.S.A.
- [2] Aqib Al Azad, Minhazul Huq, Iqbalur. Rahman Rokon, "Efficient Hardware Implementation of Reed Solomon Encoder and Decoder in FPGA using Verilog", International Journal of Advancements in Electronics And Power Engineering (ICAEPE'2011), Bangkok, Dec.2011.
- [3] Aqib Al Azad, Md Imam Shahed, "A Compact and Fast FPGA Based Implementation of Encoding and Decoding Algorithm Using Reed Solomon Codes," International Journal of Future Computer and Communication, Vol. 3, No. 1, February 2014.
- [4] Z. Ni, Y. Q. Shi, N. Ansari, W. Su, "Reversible data hiding", IEEE Transactions on Circuits Systems and Video Technology, Vol. 16, No. 3, pp. 354-362, March 2006.
- [5] Dipalaxmi Chaudhari, Mayura Bhujade, Pranali Dhumal, "VHDL Design and FPGA Implementation of Reed Solomon Encoder and Decoder for RS (7, 3)", International Journal of Science, Engineering and Technology Research (IJSETR), Vol. 3, Issue 3, March 2014.
- [6] Taesoo Kim, Sunghwan Kim, "Efficient Transmission of Reversible Data Hiding in Encryption Images by Using Reed-Solomon Codes," IEEE 3rd International Conference on Future Internet of Things and Cloud, pp. 765-769, Oct. 2015.
- [7] Huang-Chi Chen, Yu-Wen Chang, Rey-Chue Hwang, "The Modulation Method based on Reed-Solomon code for Watermarking," IEEE, pp. 633-637, Sept 2012.