ISSN 2321-6905

International Journal of Science Engineering and Advance Technology

Steganography with BSS-RSA-LSB technique: A new approach to Steganography.

Rakesh Nayak
Associate Professor,Department of CSE,
Sri Vasaviengineering College, Tadepalligudem,Andhrapradesh

Abstract— Security of information plays a vital role in data transmission. For this purpose steganography and cryptography is plays a major role. Steganography embed a text into cover image and cryptography convert plain text into the cipher text vice versa. There are many algorithms that has been proposed for steganography and cryptography for safe transmission of the data. But intruder sometimes identify the secret message from the stegano image. In the proposed system we apply BSS technique on secret message after to convert the cipher text using RSA algorithm. The cipher text is embedded into cover image using LSB technique, it generates stegano image.

Keywords—BSS, LSB, RSA, Encryption, decryption

I. INTRODUCTION

The word "Steganography" comes from the Greek and it means "covered or secret writing". As defined today, it is the technique of embedding information into something else for the sole purpose of hiding that information from the casual observer. In stego the information is hidden in the image whereas watermarking actually adds something to the image (like the word "Confidential"), and therefore becomes part of the image. Some people might consider stego to be related to encryption, but they aren't the same thing. We use encryption the technology to translate something readable to something unreadable to protect sensitive or confidential data. In stego the information is not necessarily encrypted, just hidden from plain view.

Steganography to hide it. This two-step process adds additional security. If someone manages to figure out the Steganography system used, he wouldn't be able to read the data, he extracted because it's encrypted.

The word "Steganography" comes from the Greek and it means "covered or secret writing". As defined today, it is the technique of embedding information into something else for the sole purpose of hiding that information from the casual observer. In stego the information is hidden in the image whereas watermarking actually adds something to the image (like the word "Confidential"), and therefore becomes part of the image. Some people might consider stego to be related to encryption, but they aren't the same thing. We use encryption the technology to translate something readable to something unreadable to protect sensitive or confidential data. In stego the information is not necessarily encrypted, just hidden from plain view.

1.2 LSB Technique

Least significant bit (LSB)[6] insertion is a common and simple approach to embed information in an image file. In this method the LSB of a byte is replaced with an M˚s bit. This technique works well for image steganography. To the human eye the stego image will look identical to the carrier image. For hiding information inside the images, the LSB (Least Significant byte) method is usually used. To a computer an image file is simply a file that shows different colors and intensities of light on different areas of an image.

The best type of image file to hide information inside is a 24 Bit BMP (Bitmap) image. When an image is of high quality and resolution it is a easier to hide information inside image. Although 24 Bit images are best for hiding information due to their size. Some people may choose 8 Bit BMP˚s or possibly another image format such as GIF. The reason being is that posting of large images on the internet may arouse suspicion. The least significant bit i.e. the eighth bit is used to change to a bit of the secret message. When using a 24 bit image, one can store 3 bits in each pixel by changing a bit of each of the red, green and blue color components. Suppose that we have three adjacent pixels (9 bytes) with the RGB encoding

10010101 00001101 11001001
10010110 00001111 11001011
10011111 00010000 11001011 11

When the number 300, can be which binary representation is 100101100 embedded into the least significant bits of this part of the image. If we overlay these 9 bits over the LSB of the 9 bytes above we get the following (where bits in bold have been changed)

ISSN 2321-6905

10010101 00001100 11001000
10010111 00001110 11001011
10011111 00010000 11001010

Here the number 300 was embedded into the grid, only the 5 bits needed to be changed according to the embedded message. On average, only half of the bits in an image will need to be modified to hide a secret message using the maximum cover size.

The Fibonacci LSB Data Hiding Technique proposed by Battisti et al. investigates decomposition into a different set of bit-planes, based on the Fibonacci– p-sequences, given by,

$$F_p(0) = F_p(1) = 1$$

$$F_p(n) = F_p(n-1) + F_p(n-p-1), \forall n \geq 2, n \in N$$

and embed a secret message-bit into a pixel if it passes the Zeckendorf condition, then during extraction, follow the reverse procedure.

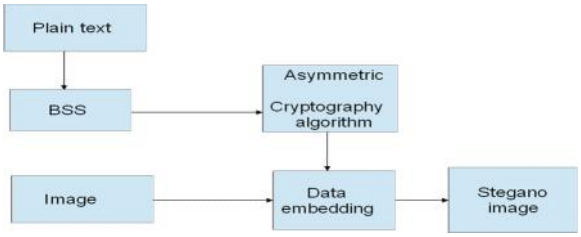
Data hiding technique using prime decomposition as an improvement over Fibonacci. Virtual bit-planes are generated using Prime Decomposition. The weight function of the Prime Number System is defined as:

$$P(0) - 1, P(i) = p_i, \forall i \in Z^+, p_{i=i^{th}}\ Prime, p_0 = 1, p_0 = 2. p_0 = 3, p_0 = 5 \ldots\ldots$$

and embed a secret message-bit into a pixel if after embedding it still remains as a valid representation. It has been shown that this technique not only increases the options for embedding by increasing number of bit-planes but also gives less distortion than classical binary and Fibonacci Decomposition, while embedding message in higher bit-planes.

I. PROPOSED SYSTEM

A new system is proposed, wherein two popular techniques RSA and BSS are combined in steganography. which provide a better security. In this proposed system we apply BSS technique on secret message and then convertit to cipher text using RSA algorithm. The cipher text is embedded into cover image using LSB technique, it generates stegano image.



2.1 BSS

2.3.1 Bit shifting

Bit shifting[2] is an operation done on all the bits of a binary value in which they are moved by a determined number of places to either the left or right. Bit shifting is used when the operand is being used as a series of bits rather than as a whole. In other words, the operand is treated as individual bits that stand for something and not as a value. Bit shifting is often used in programming and has at least one variation in each programming language. Bit shifting may also be known as a bitwise operation.

2.3.2 Bit stuffing

Bit stuffing is the process of inserting non information bits into data to break up bit patterns to affect the synchronous transmission of information. It is widely used in network and communication protocols, in which bit stuffing is a required part of the transmission process. Bit stuffing is commonly used to bring bit streams up to a common transmission rate or to fill frames. Bit stuffing is also used for run-length limited coding. The system deals with security of data by using BSS encryption and decryption.

2.3.3 Sender Process

In this process every eight bytes of plain text becomes seven bytes of cipher text. So another advantage of this method is when it encrypts it reduces the size of the data.In this process let us consider I1 , I2 , I3 , I4 , I5 , I6 , I7 and I8 represents 8 printable characters of plain text and the values in the boxes represents the byte equivalent values of each character. i.e. a1 a2 a3 a4 a5 a6a7 represents 7bits of character I1 and their value may be either 0 or 1. Similarly remaining character bits are represented in boxes as shown in figure 1. In this process the last character I8 bits h1 , h2 , h3 , h4 , h5 , h6 , h7 are shifted and stuffed in to the characters I7 , I6, I5 , I4 , I3 , I2 , I1 respectively as shown in figure.

I1:	0	a1	a2	a3	a4	a5	a6	a7
I2:	0	b1	b2	b3	b4	b5	b6	b7
I3:	0	c1	c2	c3	c4	c5	c6	c7
I4:	0	d1	d2	d3	d4	d5	d6	d7
I5:	0	e1	e2	e3	e4	e5	e6	e7
I6:	0	f1	f2	f3	f4	f5	f6	f7
I7:	0	g1	g2	g3	g4	g5	g6	g7
I8:	0	h1	h2	h3	h4	h5	h6	h7

Before Encryption.

I1:	h7	a1	a2	a3	a4	a5	a6	a7
I2:	h6	b1	b2	b3	b4	b5	b6	b7
I3:	h5	c1	c2	c3	c4	c5	c6	c7
I4:	h4	d1	d2	d3	d4	d5	d6	d7
I5:	h3	e1	e2	e3	e4	e5	c6	c7
I6:	h2	f1	f2	f3	f4	f5	f6	f7
I7:	h1	g1	g2	g3	g4	g5	g6	g7

After Encryption.

For example the eight characters are a, b, c, d, e, f, g and h. i.e. I1 = A, I2 = B, I3 = C, I4 = D, I5 = E, I6 = F, I7 = G, and I8=H. The equivalent byte values of these characters before encryption are as follows:

I1 ASCII value 65 and its bits: 01000001, i.e. a1= 1, a2= 0, a3= 0, a4 =0, a5= 0, a6= 0, a7= 1.

I2 ASCII value 66 and its bits: 01000010, i.e. b1= 1, b2= 0, b3= 0, b4= 0, b5= 0, b6=1, b7= 0.

I3 ASCII value 67 and its bits: 01000011, i.e. c1= 1, c2= 0, c3= 0, c4= 0, c5= 0, c6= 1, c7= 1.

I4 ASCII value 68 and its bits: 01000100, i.e. d1= 1, d2= 0, d3= 0, d4= 0,d5= 1,d6 = 0,d7= 0.

I5 ASCII value 69 and its bits: 01000101, i.e. e1= 1, e2= 0, e3= 0, e4= 0, e5= 1, e6= 0, e7= 1.

I6 ASCII value 70 and its bits: 01000110, i.e. f1= 1, f2= 0, f3= 0, f4= 0, f5 = 1, f6 = 1, f7 = 0.

I7 ASCII value 71 and its bits: 01000111, i.e.g1= 1, g2= 0, g3= 0, g4= 0, g5= 1, g6=1, g7= 1.

ISSN 2321-6905

I8 ASCII value 72 and its bits: 01001000, i.e. h1=1, h2= 0, h3= 0, h4= 1, h5=0, h6= 0, h7= 0.

After encryption by using BSS method the equivalent byte values of these characters are as follows.

I1 bits: 01000001, i.e. h7 = 0, a1 = 1, a2 = 0, a3 = 0, a4 = 0, a5 = 0, a6 = 0, a7 = 1.
I2 bits: 01000010, i.e. h6 = 0, b1 = 1, b2 = 0, b3 = 0, b4 = 0, b5 = 0, b6 = 1, b7 = 0.
I3 bits: 01000011, i.e. h5 = 0, c1 = 1, c2 = 0, c3 = 0, c4 = 0, c5 = 0, c6 = 1, c7 = 1.
I4 bits: 11000100, i.e. h4 = 1, d1 = 1, d2 = 0, d3 = 0, d4 = 0, d5 = 1, d6 = 0, d7 = 0.
I5 bits: 01000101, i.e. h3 = 0, e1 = 1, e2 = 0, e3 = 0, e4 = 0, e5 = 1, e6 = 0, e7 = 1.
I6 bits: 01000110, i.e. h2 = 0, f1 = 1, f2 = 0, f3 = 0, f4 = 0, f5 = 1, f6 = 1, f7 = 0.
I7 bits: 11000111, i.e. h1 = 1, g1 = 1, g2 = 0, g3 = 0, g4 = 0, g5 = 1, g6 = 1, g7 = 1.

2.3.4. Receiver Process

In decryption process every seven bytes of cipher text produces eight characters of plain text. So after decryption process the decrypted data will automatically get its original size. The fallowing figure 3 shows data before decryption, and figure 4 shows the data after decryption.

I1:	h7	a1	a2	a3	a4	a5	a6	a7
I2:	h6	b1	b2	b3	b4	b5	b6	b7
I3:	h5	c1	c2	c3	c4	c5	c6	c7
I4:	h4	d1	d2	d3	d4	d5	d6	d7
I5:	h3	e1	e2	e3	e4	e5	e6	e7
I6:	h2	f1	f2	f3	f4	f5	f6	f7
I7:	h1	g1	g2	g3	g4	g5	g6	g7

Apply BSS.

I1:	0	a1	a2	a3	a4	a5	a6	a7
I2:	0	b1	b2	b3	b4	b5	b6	b7
I3:	0	c1	c2	c3	c4	c5	c6	c7
I4:	0	d1	d2	d3	d4	d5	d6	d7
I5:	0	e1	e2	e3	e4	e5	e6	e7
I6:	0	f1	f2	f3	f4	f5	f6	f7
I7:	0	g1	g2	g3	g4	g5	g6	g7
I8:	0	h1	h2	h3	h4	h5	h6	h7

Original data.

A. Maintaining the Integrity of the Specifications

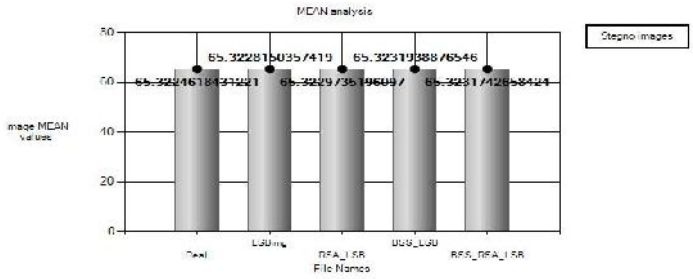
The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

II. EXPERIMENTAL RESULTS

In thhis paper, an attempt is made to compare the original image with the image only by applying LSB stegnography, RSA-LSB stegnography, BSS-LSB stegnography and BSS-RSA-LSB stegnography by using various statistical methods.

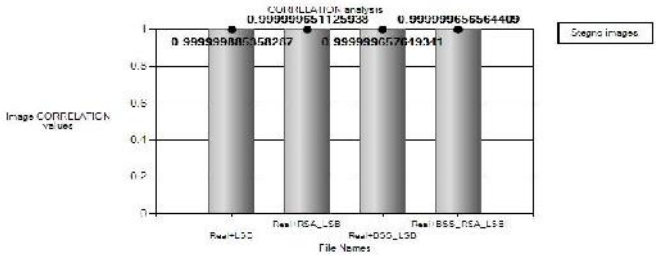
Mean:

Method and Image	Mean
Original Image	65.3224618431221
LSB Steganoraphy	65.3228150357419
RSA-LSB Steganoraphy	65.3229735196097
BSS-LSB Steganoraphy	65.3231938876546
BSS-RSA-LSB Steganoraphy	65.3231742658424



Correlation analysis

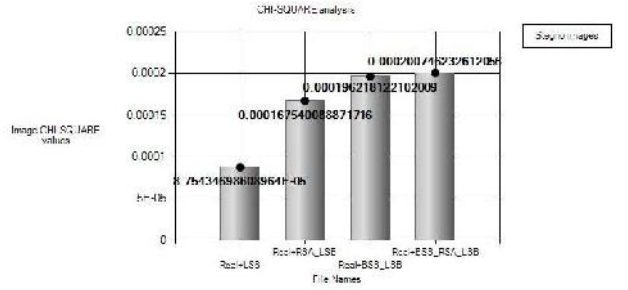
Compare with original image	Correlation Coefficient
LSB Steganography	0.999999885358287
RSA LSB Steganography	0.999999651125938
BSS LSB Steganography	0.999999657649341
BSS-RSA-LSB Steganography	0.999999656564409

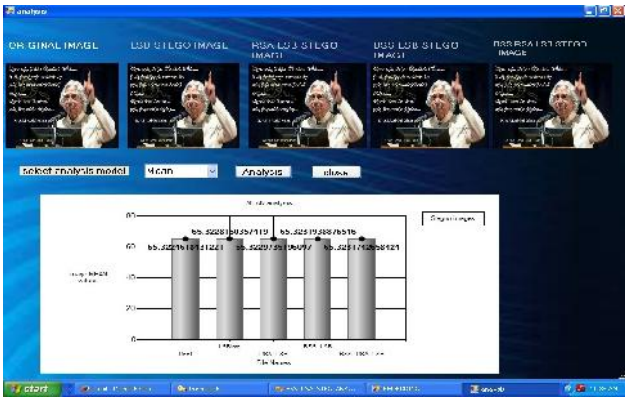


Chi-Square analysis

Compare with original image	Chi-Square
LSB Steganography	8.75434698608964E-05
RSA LSB Steganography	0.000167540088871716
BSS LSB Steganography	0.000196218122102009
BSS-RSA-LSB Steganography	0.000200745232612056

ISSN 2321-6905





IV.CONCLUSION

The proposed system used in this project encrypts the secret message before we apply the BSS technique after embedding into the image. From the analysis we compared RSA with BSS and without BSS based on mean, correlation coefficient and Chi-square methods and conclude that RSA with BSS is suitable for secret message hiding in Steganography.

REFERENCES

[1]. G.Suman, P.Anuradha,"RSA Algorithm and LSB Steganography", International Journal of Engineering Research & Technology (IJERT), Vol.2 Issue 10, October – 2013, ISSN: 2278-0181.

[2]. B.Ravi Kumar, Dr.P.R.K.Murti,"Data Encryption and Decryption process Using Bit Shifting and Stuffing (BSS) Methodology", International Journal on Computer Science and Engineering (IJCSE), Vol. 3 No. 7 July 2011, ISSN: 0975-3397.

[3]. B.Ravi Kumar, P.R.K.Murti, B.Hemanth Kumar,"An Authenticated Bit Shifting and Stuffing (BSS) Methodology for Data Security", Computer Engineering and Intelligent Systems www.iiste.org, Vol 2, No.3, ISSN 2222-1719, ISSN: 2222-2863.

[4]. Mehdi Hussain and Mureed Hussain," A Survey of Image Steganography Techniques", International Journal of Advanced Science and Technology, Vol. 54, May 2013.

[5]. Kshetrimayum Jenita Devi,"A Secure Image Steganography Using LSB Technique and Pseudo Random Encoding Technique", May 2013.

[6]. Mamta Juneja and Parvinder S.Sandhu,"An Improved LSB Based Steganography Technique for RGB Color Images", International Journal of Computer and Communication Engineering, Vol. 2, No. 4, July 2013.

[7]. Shailender Gupta, Ankur Goyal, Bharat Bhushan," Information Hiding Using Least Significant Bit Steganography and Cryptography", I.J.Modern Education and Computer Science, 2012.

[8]. Jatinder Kaur, Ira Gabba,"Steganography Using RSA Algorithm", International Journal of Innovative Technology and Exploring Engineering (IJITEE) Volume-3, Issue-3, August 2013, ISSN: 2278-3075.

[9]. Anil Kumar, Rohini Sharma,"A Secure Image Steganography Based on RSA Algorithm and Hash-LSB Technique", International Journal of Advanced Research in Computer Science and Software Engineering Volume 3, Issue 7, July 2013 ISSN: 2277 128X.