ISSN (Online): 2455-7838
SJIF Impact Factor (2016): 4.144

EPRA International Journal of

Research & Development

(IJRD)

Monthly Peer Reviewed & Indexed International Online Journal

Volume:2, Issue:8, August 2017

Published By : EPRA Journals

ISSN: 2455-7838(Online)
EPRA International Journal of Research and Development (IJRD)
SJIF Impact Factor: 4.144 Volume: 2 | Issue: 8 | August | 2017

AN EXTENDED STRUCTURED-LEAST SIGNIFICANT BIT STEGANOGRAPHY TECHNIQUE

Babangida Zachariah[1]
[1]Kaduna State University, Kaduna, Nigeria

Yockson Inuwa Gajere[2]
[2]KSCOE Gidan Waya, Nigeria

Amos Peter[3]
[3]Nuhu Bamali Polytechnic Zaria, Nigeria

ABSTRACT

Security of data communication remains a challenge even though there has been a lot of researches done to handle various security challenges. The application of cryptography and steganography remain some of important approaches used to enhance security. Steganography hides the existence of a communication such that a hacker is unaware of it, thus, does not exploit and bridge such security. However, the use of steganalysis tools have now given the hackers an edge in determining the use of steganography in a message in transmission; and also, able to retrieve the hidden message. Techniques of least significant bit steganography usually performs embedding in a linear sequence of the cover media used which makes the retrieval of the hidden message very easy once the a steganalysis tool detects the presence of steganography in the cover media. Therefore, this paper presents a new technique of steganography that embeds message bits into a cover media in spiral and diagonal fashion; an extended approach of structured-least significant bit steganography proposed in Babangida (2016). The proposed technique was implemented and tested against most relevant existing techniques and was found to perform better in terms of the quality of the cover and Stego-image analysis. However, the results were similar to those of Babangida (2016).

INDEX TERMS: *Steganography, LSB, Spatial Domain, Secret Message, Structure-LSB*

1 INTRODUCTION

Although there has been several researches in the area of data communication security, ensuring absolute security of such communication remains a challenge. This is specifically true with the advancement of technologies that have made such communication possible and easier. That is, the same technologies are being used to exploit the security loopholes inherent in the communication systems (Tawade, et al., 2012). However, the application of cryptography, steganography, and other security approaches have been used to minimize the susceptibility of data communication to security threat.

Cryptography scrambles (encrypts) a data communication such that it is not readable except when unscrambled (decrypted) with the key used in encrypting the message (data communication). The security of data communication provided by the application of cryptography is often a victim of security attacks; and with advancements in technologies and knowledge, cracking a cryptographic key is now becoming an easy task. Therefore, a more sophisticated and complimentary approach is required. Steganography has been used to compliment or augment cryptography (Doshi, et al., 2012).

Steganography is a technique that hides a message which is supposed to be a secret message in a cover media which is not secret and with less suspicion of carrying a hidden message. The secret message is often an important message which should not be accessible to a third party. Therefore, hiding its existence in a cover media which is often either an image, text, audio, or video file is the most suitable approach to the security. In a steganography system, a secret message to be transmitted is converted into bits, a cover media file which is not meant to be secret (that is, does not contain sensitive and "useful" information) is also converted into bits, and the bits of the secret message are then embedded into the bits of the cover media. This should be done such that the distortion is minimal so as not to create suspicion on the part of hackers. Thus, making steganography a sophisticated approach to security and the possibility of attacks on data communication using steganography as a security measure is less likely (Jain & Kumar, 2012; Babangida, 2016).

In steganography, although various method such as distortion method, substitution method, transform domain method, or statistical method are being used, distortion method is often used due to its simplicity and reasonable degree of effectiveness (Gaikwad & Wagh, 2010; Medeni & Souidi, 2010). In distortion method, least significant bit (LSB) algorithm is often used. However, the method of embedding the secret message into a cover media is usually linearly done. For example, when using an image as a cover media in least significant bit steganography, the bits of the secret message are hidden sequentially in the sequence pixels of the image. This makes this approach to security to be easily threatened. A hacker using a steganalysis tool, once a stego image (the image containing the secret message) is uncovered, the hacker only needs to retrieve bits of the 8^{th} bit plane and decode the hidden message (Sarkar, et al., 2007; Sadkhan, et al., 2009; Sun & Lui, 2010; Seyyedi & Ivanove, 2014). The desired security is yet to be achieved. Therefore, while hiding the existence of the secret, it is also desirable to make the retrieval of such a hidden secret message very difficult so that in an event of detection of the stego image by steganalysis technique, the message is not easily gotten. That is, steganography should have added another layer of encryption using the embedding algorithm.

This research work presents an extended approach of embedding secret message bits using the work of Babangida (2016) where structured-least significant bit steganography was proposed. The embedding approach in the work was using the 6^{th}, 7^{th}, and the 8^{th} bit plane of the RGB component of the cover image. The number of secret message bits hidden either in the 6^{th}, 7^{th}, or 8^{th} bit planes of the cover image is based on a random number. For example, three bits of the secret message bits may be hidden in the 6^{th} bit plane of the cover image, then embedding switches to the 7^{th} bit plane and the next three bits of the secret message are hidden in the 7^{th} bit plane, and the next three sequence of bits of the secret message are hidden in the 8^{th} bit plane and then switches back to the 6^{th} bit plane. At retrieval, the same approach is used. The technique added a layer of encryption since when retrieving the bits of the secret message after steganalysis, the hacker often retrieve bits of particular bit plane. In fact, even ones involve in steganography and transmission of such stego images cannot tell of the number of bits embedded on a given plane, or the planes used. This is an additional layer of encryption since it is known that when steganography and encryption are combined together, the security of the system is enhanced (Babangida & Ismail, 2016). Thus, the main aim of this work is to propose a steganography technique that is robust, have high imperceptibility of the stego image, high capacity, and irretrievability of the hidden message by a hacker.

2 RELEVANT EXISTING TECHNIQUES

In Datta et al, a new technique of LSB steganography was proposed. The secret message bits were not hidden directly in the cover image but in the intensity layer of the pixels of the cover image using binary addition. This was done such that the maximum change in the intensity layer remains nominal and does not depend on the number of LSB layer chosen for the

binary addition. To enhance capacity, two secret message bits are hidden in each pixel. Therefore, the technique satisfy the major requirements of steganography, which are robustness, imperceptibility, capacity, and irretrievability by a hacker (Datta, et al., 2016).

In the work of (Apau & Adomako, 2017), ensuring security on mobile platforms such as Blackberry, Android, iPhone, etc. using the technique of encryption and steganography was proposed. RSA encryption algorithm and LSB technique were used. The stego images were analyzed against mean square error (MSE) and peak signal to noise ratio (PSNR) and results showed high security and robustness in the mobile platform.

In Maneseer et al, standard LSB and condition based LSB techniques were proposed. These techniques used a standard reference for the embedding of secret message bits in the cover image. The proposed system was implemented on MATLAB and assessed in terms of number of bits change by the two techniques, mean square error, peak signal to noise ratio. The results showed that the condition based technique change the least number of bits compared to the standard LSB technique, and the security of the technique was enhance since the real data was not hidden but the reference (Manaseer, et al., 2017).

In the work of (Rajendran & Doraipandian, 2017), a symmetric key based LSB steganography technique was proposed. The proposed technique used one-dimensional (1D) logistic map in generating pseudo random keys used in randomly selecting pixels of the cover image where bits of the secret message are to be hidden. The random choice of the pixel locations to embed secret message bits is a security enhancement. The efficiency of the proposed technique was assessed using PSNR and MSE and found to provide efficient security.

Also, in their work, Ziyad et al, proposed an LSB steganography approach based on firefly and particle swarm optimization (PSO) algorithms. The firefly algorithm determine the brightest fireflies (the highest pixel values), calculate the distance between the brightest fireflies and others, calculate the attractiveness between each pixel and the brightest firefly, calculates the movement of pixels to brightest ones, using the three computed parameters, a quick sort algorithm is used to rank them, and then select the various position based on this to hide the secret message in the blue component of the chosen pixel location. For the PSO algorithm, the particle is placed at the center of the cover image, a fitness value is determined, the best position for hiding the secret message is chosen and the bit is hidden at the blue component of the chosen location; and the particle position is updated and then the procedure is repeated. Assessing the two approaches, the authors found that though both firefly and PSO algorithms are good search techniques, the firefly algorithm is better from the steganography point of view (Ziyad, et al., 2017).

3 PROPOSED SYSTEM

The ease of retrieving secret messages linearly embedded in one bit plane of the cover-images once steganalysis tool identifies the Stego-image makes the approach less secured. Our proposed algorithm uses the LSB standard approach and embeds secret messages at the sixth (6^{th}), or seventh (7^{th}), or eight (8^{th}) bit planes of the cover image. That is, the embedding is strictly on one-bit plane that may be either of 6^{th}, or 7^{th}, or 8^{th}-bit plane; and not combined, in which case the embedding switches between planes that are used. This new approach may embed secret message bits in the different bit planes either linearly, spirally/radially or diagonally. The use of the different bit planes and alternative embedding sequence should add the desired security, making retrieval of secret messages difficult. Therefore, the new approach is an extension and a bit modification of (Babangida, 2016).

3.1 The Header

Based on the work of (Babangida, 2016), the proposed header is used here with a single modification. Thus, the Steganography algorithm has a defined structure which is provided as a header information as shown in Table 1. Every other field remain unchanged both in terms of size and meaning.

The introduction of Algorithm field in the header is to determine the particular embedding pattern used. That is, either linear, spirally, or diagonally. The choice of which to use also like the Stego-Type and Switch indicator depends a runtime random number between zero (0) and fifteen (15). For the purpose of this work, if the random number is less than or equal to five (5), the embedding pattern is linear but switching between bit planes as determined by Stego-Type and Switch Indicator; if the random number is greater than five (5) but less than or equal to ten (10), the embedding is done spirally, and otherwise, the embedding is done diagonally. Therefore, at Algorithm field, if the bits are zero-zero (00), it is linear embedding; if the bits are zero-one (01), the embedding is spiral; and if the bits are one-one (11), the embedding is diagonal.

3.2 Bits Embedding Algorithms

The bits of the secret message to be hidden in a cover image are embedded either linearly, spirally, or diagonally. This section presents the algorithms for embedding the bits.

For a given execution of our system with the cover image I, secret message M, bits of the secret message B, stego-type T, switch indicator S, and algorithm A; the system makes a choice of the algorithm as follows

Stego Type 4bits	**Steganography Switch Indicator 3bits**	**Password Indicator 1bit**	**Algorithm 2bits**	**Header Length 6bits**
Data Length 16bits				
Data Start Location 8bits			Password Length 8bits	
Password Start Location 8bits			App Signature 8bits	
Password				
Password				
Data				
Data				

Table 1: Steganography Algorithm Header

```
Begin
        EmbedHeaderInfo()
        B = ConvertToBits(M)
        If A = "00" then
                stegoImage = LinearEmbedding(I, B, T, S)
        ElseIf A = "01" then
                stegoImage = SpiralEmbedding(I, B, T, S)
        ElseIf A = "11" then
                stegoImage = DiagonalEmbedding(I, B, T, S)
        End if
        SaveStegoImage(stegoImage)
End
```

First the header information is embedded linearly starting from the (x_1, y_1) coordinate of the cover image. Converts the secret message to bits equivalent, and then based on the bits stored in A the choice of the algorithm is made. Once the choice of algorithm has been made, the algorithm performs the embedding as described next.

```
LinearEmbedding(I, B, T, S)
        Begin
                imageBits = GetImageBitsLinearlly(I)
                k = 1      //imageBits Pointer
                j = 1      //Secret Message bits B pointer
                while k <= Length(imageBits)
                        bitPlane = SwitchBitPlane(T)
                        for i = 1 To S
                                Embed(imageBits, k, B, j, bitPlane)
                                j += 1
                                k += 8
                        endfor
                endwhile
                stegoImage = WriteImageBitsLinearlly(imageBits)
                return stegoImage
        end
```

```
SpiralEmbedding(I, B, T, S)
        Begin
                imageBits = GetImageBitsSpirally(I)
                k = 1     //imageBits Pointer
                j = 1     //Secret Message bits B pointer
                while k <= Length(imageBits)
                        bitPlane = SwitchBitPlane(T)
                        for i = 1 To S
                                Embed(imageBits, k, B, j, bitPlane)
                                j += 1
                                k += 8
                        endfor
                endwhile
                stegoImage = WriteImageBitsSpirally(imageBits)
                return stegoImage
        end
DiagonalEmbedding(I, B, T, S)
        Begin
                imageBits = GetImageBitsDiagonally(I)
                k = 1     //imageBits Pointer
                j = 1     //Secret Message bits B pointer
                while k <= Length(imageBits)
                        bitPlane = SwitchBitPlane(T)
                        for i = 1 To S
                                Embed(imageBits, k, B, j, bitPlane)
                                j += 1
                                k += 8
                        endfor
                endwhile
                stegoImage = WriteImageBitsDiagonally(imageBits)
                return stegoImage
        end
GetImageBitsLinearlly(image)
Begin
        rowCount = GetHeight(image)- 1
        columnCount = GetWidth(image)
        imageBits = ""
        row = 1
        while row < rowCount
                col = 1
                while col < columnCount
                        //Get bits of RGB at location (row,col)
                        imageBits += GetRGBBits(image, row, col)
                        col++
                endwhile
                row++
        endwhile
        return imageBits
end
GetImageBitsDiagonally(image)
Begin
        rowCount = GetHeight(image)- 1
        columnCount = GetWidth(image)
        imageBits = ""
        r = 1
        while r < rowCount
                row = r
```

```
                col = 1
                while row >= 1 AND col < columnCount
                        //Get bits of RGB at location (row,col)
                        imageBits += GetRGBBits(image, row, col)
                        row--
                        col++
                endwhile
                r++
        endwhile
        c = 1
        while c < columnCount
                row = rowCount-1; col = c
                while row >= 1 && col < columnCount
                        //Get bits of RGB at location (row,col)
                        imageBits += GetRGBBits(image, row, col)
                        row--
                        col++
                endwhile
                c++
        endwhile
        return imageBits
end

GetImageBitsSpirally(image){
Begin
        rowStart = 1; colStart = 1
        rowLength = GetHeight(image) -1
        colLength = GetWidth(image)

          while rowStart <= rowLength AND colStart <= colLength
                for i = rowStart To colLength
                        imageBits += GetRGBBits(image, rowStart, i)
                endfor
                for j = rowStart+1 UpTo rowLength
                        imageBits += GetRGBBits(image, j, colLength)
                endfor
                if rowStart+1 <= rowLength
                        for k = colLength-1 DownTo colStart
                                imageBits += GetRGBBits(image, rowLength, k)
                        endfor
                endif
                if colStart+1 <= colLength
                        for k = rowLength-1 DownTo rowStart
                                imageBits += GetRGBBits(image, k, colStart)
                        endfor
                endif
                Increment(rowStart)
                Decrement(rowLength)
                Increment(colStart)
                Decrement(colLength)
          endwhile
          return imageBits
end
```

The above algorithms describe the high-level implementation of the various major program functions of the extended system. The bits of the cover image are obtained in the order of Red Green Blue sequence either linearly, spirally, or diagonally depending on the choice of the embedding Algorithm. The bits of the secret message are then hidden into the bits of the cover image with reference to the chosen number of bits (switch indicator) hidden at a particular plane; and the chosen planes (either 6^{th}, 7^{th}, 8^{th} bit plane or a combination of these bit planes) into which the secret message is to be hidden. Also, depending on the plane into which the secret bit is hidden, the subsequent bits may be changed so as to reduce the distortion done to the pixel. For example, to hide a bit "0" at the 6^{th} bit location of "01111100" (whose decimal equivalent is 124) would result in "01111000" (whose decimal equivalent is 120 which is less than 124). Thus, the subsequent bits at the 7^{th} and 8^{th} plane are change such that we have "01111011" whose decimal equivalent is 123 which is only 1 less than 124 instead of being 4 less.

3.3 Message Retrieval Process

In retrieving the hidden secret message from the stego image, the system first retrieves the header information to determine the used Stego-Type, Switch Indicator, Algorithm and all other information. Based on this information, the bits of the secret message are retrieved from the stego image and then converted to corresponding text.

4 IMPLEMENTATION AND EXPERIMENTAL RESULTS

The proposed technique was implemented using Visual Basic .NET Framework and experimental results were compared to those of (Babangida, 2016) which this work extends. The steganography was done on PNG images which are lossless and analysis of the stego images was done using Matlab. The Peak-Signal-To-Noise Ratio (PSNR), Mean Square Error (MSE) of the images were computed.

To test the extended system against the standard system proposed in (Babangida, 2016), the system was tested modularly using 53 chars of text as in the previous work. A control execution where embedding was done using linearly, diagonally and spirally embedding on the same images used in the previous work. The results of were similar for all the approaches.

5 SUMMARY, CONCLUSION, AND RECOMMENDATION

This extension proposes more embedding algorithms which takes advantage of the structured header proposed in the previous work. This certainly adds a layer of security as when the message bits are being retrieved from the stego image, the must be done in the order in which they were embedded otherwise, a scrambled text is returned. However, just predicting the order in which the bits were embedded is difficult. This requires one to be able to predict the Stego-Type, Switch Indicator, and Algorithm used.

Also, to add more security and make the accurate prediction of the order of secret message embedding more difficult, more alternate embedding Algorithms could be incorporated in the system. These other algorithms may be mathematically based such as the use of chaotic map theory or other prediction algorithms. Compression and encryption may also be used to reduce the distortion done on the image and to add layer of security to the system.

6 REFERENCES

1 *Apau, R. & Adomako, C., 2017. Design of Image Steganography based on RSA Algorithm and LSB Insertion for Android Smartphones. International Journal of Computer Applications , 164(1), p. 0975 – 8887.*

2 *Babangida, Z., 2016. A Novel Structured-Least Significant Bit Steganography Technique. International Journal of Information Systems and Engineering, pp. 1-9.*

3 *Babangida, Z. & Ismail, M. Z., 2016. A PROPOSED ONLINE SHOPPING AND PAYMENT FRAMEWORK WITH APPLICATION OF ENCRYPTION AND STEGANOGRAPHY FOR ENHANCED SECURITY. International Journal of Information Systems and Engineering, 4(1), pp. 10-29.*

4 *Datta, B., Mukherjeeb, U. & Bandyopadhyay, S. K., 2016. LSB Layer Independent Robust Steganography using Binary Addition. International Conference on Computational Modeling and Security (CMS 2016) : Procedia Computer Science 85, p. 425 – 432.*

5 *Doshi, R., Jain, P. & Gupta, L., 2012. Steganography and Its Applications in Security. International Journal of Modern Engineering Research, 2(6), pp. 4634-4638.*

6 *Gaikwad, D. & Wagh, S., 2010. Colour Image Restoration For An Effective Steganography. Imanager's Journal on Software Engineering, Vol.4 .No.3, pp. 65-71.*

7 *Haynes, K. L., 2011. Using Image Steganography to Establish Covert Communication Channels. International Journal of Computer Science and Information Security, Vol 9, No.9, pp. 1 - 7.*

8 *Hemalatha, S., Acharya, U., Renuka, A. & Kamnath, P. R., 2013. A Secure and High Capacity Image Steganography Technique. Signal & Image Processing – An International Journal Vol.4, No.1, pp. 83 - 89.*

9 *Hemalatha, S. A. U. & Renuka, A., 2013. Comparison of Secure and High Capacity Color Image Steganography Techniques in RGB and YCBCR domains. International Journal of Advanced Information Technology, Vol.3, No.3, pp. 1 - 9.*

10 *Jain, R. & Kumar, N., 2012. Efficient Data Hiding Scheme Using Lossless Data Compression and Image Steganography. International Journal of Engineering Science and Technology (IJEST). Vol. 4 No.08, pp. 3908 - 3915.*

11 *Juneja, M. & Singh, P. S., 2013. A New Approach for Information Security Using an Improved Steganography Technique. Journal of Info.Pro.Systems, Vol 9, No:3, pp. 405 - 424.*

12 *Laskar, S. A. & Hemachandran, K., 2013. Steganography Based on Random Pixel Selection For Efficient Data Hiding. International Journal of Computer Engineering and Technology. Vol.4, Issue 2, pp. 31 - 44.*

13 *Manaseer, S., Aljawawdeh, A. & Alsoudi, D., 2017. A New Image Steganography Depending On Reference & LSB. International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 9, pp. 1950-1955.*

14 *Medeni, M. & Souidi, E. M., 2010. Steganography and Error Correcting Codes. International Journal of Computer Science and Information Security, Vol.8.No.8, pp. 147-149.*

15 *Rajendran, S. & Doraipandian, M., 2017. Chaotic Map Based Random Image Steganography Using LSB Technique. International Journal of Network Security, , 19(4), pp. 593-598.*

16 *Rejani, R., Murugan, D. & Deepu, V. K., 2015. PIXEL PATTERN BASED STEGANOGRAPHY ON IMAGES. ICTACT JOURNAL ON IMAGE AND VIDEO PROCESSING, VOLUME: 05, ISSUE: 03, pp. 991 - 997.*

17 *Sadkhan, S. B., Al-Barky, A. M. & Muhammad, N. N., 2009. "An Agent based Image Steganography using Information Theoretic Parameters. MASAUM Journal of Computing, Vol. 1, No. 2, pp. 258 - 268.*

18 *Sarkar, A. et al., 2007. SECURE STEGANOGRAPHY: STATISTICAL RESTORATION OF THE SECOND ORDER DEPENDENCIES FOR IMPROVED SECURITY. Honolulu, HI, IEEE, pp. II-277 - II-280.*

19 *Seyyedi, A. S. & Ivanove, N., 2014. Statistical Image Classification for Image Steganography Techniques. International Journal of Image, Graphics and Signal Processing, pp. 19 - 24.*

20 *Shanmuga, S. P., Mahesh, K. & Kuppusamy, K., 2012. Efficient Steganography Method To Implement Selected Least Significant Bits in Spatial Domain. International Journal of Engineering Research and Applications, Vol2, Issue 3, pp. 2632 - 2637.*

21 *Sharmila, B. & Shanthakumari, R., 2012. "Efficient Adaptive Steganography For Colour Images Based on LSBMR Algorithm. ICTACT Journal on Image and Video Processing, Vol. 2, Issue:03, pp. 387 - 392.*

22 *Sun, Y. & Lui, F., 2010. Selecting Cover for Image Steganography by Correlation Coeffiecient. s.l., s.n., pp. 159 - 162.*

23 *Swain, G. & Lenka, S. K., 2012. A Technique for Secret Communication Using a New Block Cipher with Dynamic Steganography. International Journal of Security and Its Applications; Vol. 6 No. 4, pp. 13 - 24.*

24 *Tawade, L., Mahajan, R. & Kulthe, C., 2012. Efficient & Secure Data Hiding Using Secret Reference Matrix. International Journal of Network Security & Its Applications (IJNSA), 4(1), pp. 43-50.*

25 *Thiyagarajan, P. et al., 2013. Pattern Based 3D Image Steganography. 3D Research center, Kwangwoon University and Springer 2013, 3DR Express, pp. 1 - 8.*

26 *Ziyad, T. M. A.-T., Jamal, M. A. & Omar, Y. A. A.-H., 2017. Image Steganography between Firefly and PSO Algorithms. International Journal of Computer Science and Information Security (IJCSIS),, 15(1), pp. 9-21.*