Comput Syst Sci & Eng (2019) 1: 23–32

International Journal of
Computer Systems
Science & Engineering

A novel approach to steganography: Enhanced least significant bit substitution algorithm integrated with self-determining encryption feature

Resul Das[1]*, **Muhammet Baykara**[1] **and Gurkan Tuna**[2]

[1] *Department of Software Engineering, Technology Faculty, Firat University, 23119, Elazig, Turkey*
Tel: +90-424-2370000-4292, Fax: +90- 424-2367064, E-mail: mbaykara@firat.edu.tr
[2] *Department of Computer Programming, Trakya University, 22020, Edirne, Turkey*
Tel: +90-284-2240283, Fax: +90-284-2240288, E-mail: gurkantuna@trakya.edu.tr

One of the most well-known and used algorithms for Steganography is Least Significant Bit (LSB) substitution. Although LSB has several advantages such as simplicity, efficiency, and easy-to-do implementation, it has some distinct disadvantages such as it openness to miscellaneous attacks. In this study, we aim to improve the traditional LSB algorithm by eliminating its main disadvantage, being easy to detect, and this way propose an enhanced LSB algorithm called E-LSB. We mainly aim to minimize differences which are due to encryption and image hiding steps in LSB algorithm and make it more difficult to notice that some text has been hidden in the original cover image.
As most of the researchers and practitioners in security field argue, stenographic techniques alone are not sufficient for protecting sensitive information and thus must be used together with encryption algorithms. Therefore, the proposed approach integrates E-LSB with an encryption algorithm. E-LSB does not modify the file size and allows the flexibility of choosing one of well-known encryption algorithms including RSA, AES and CAESAR, but others can be implemented in it. With a set of experiments, the proposed approach is compared with the traditional LSB based embedding approach, and its efficiency and usability is analyzed. A set of performance evaluations realized with the developed software tool based on E-LSB algorithm show that E-LSB is better than the traditional LSB algorithm from security point of view.

Keywords: Steganography; Least Significant Bit Substitution; Enhanced LSB; Dynamic Embedding Range Determination; Information Security

1. INTRODUCTION

Steganography is a widely-used technique for information hiding within images [1-4]. It processes a message to be hidden and hides it in a cover image by means of an embedding algorithm. It has many practical uses; therefore, in recent years, steganographic techniques have received notable attention from both the research community and developers. Steganographic techniques process and modify cover images in such a way that some text can be embedded in them. In this way, only intended recipients can process them and unhide the embedded text. Since they are invisible, detecting them is not easy. Most of the existing steganographic techniques rely on substituting redundant parts of a signal with a secret message [5-7]. On the other hand, using steganalysis techniques, unintended users can detect that some text is hidden in the images and uncover it [4, 8, 9].

*Tel: +90-424-2370000-4305, Fax: +90-424-2367064, E-mail: rdas@firat.edu.tr

As it is well-known, Least Significant Bit (LSB) substitution is one of the simplest steganographic techniques used to conceal information within different types of objects such as audio, image and video files [10, 11]. LSB hides information to be protected in the LSB of each pixel in a cover image [10, 12, 13]. Although, it is possible to change only the LSBs of cover images without the modifications being visually detectable, LSB also modifies the file size of the cover images significantly in some cases, one of the most important disadvantages of LSB [14]. LSB substitution is simple, easy to implement and practical, but is not secure and vulnerable to steganalysis attacks which have the ability to uncover information in suspicious files by inspecting changes at the pixel level [1, 15].

In this study, to address the disadvantage of LSB, a complementary approach is followed, and a novel LSB substitution algorithm called Enhanced LSB (E-LSB) is proposed. The proposed substitution algorithm has two advantages over LSB. It keeps the number of modified bits at minimum and does not change the file size. Therefore, without using steganalysis tools, unintended users cannot easily understand that some text is hidden in an image. However, as it is well-known, steganograptic techniques are not sufficient to protect sensitive information due to the threat of steganalysis tools; therefore, they must be used together with encryption techniques [15]. For this aim, in the proposed approach, we integrated well-known encryption algorithms with E-LSB substitution algorithm.

The rest of the paper is organized as follows. Section 2 presents literature review with a focus on LSB-related approaches. In Section 3, LSB algorithm is reviewed and the details of E-LSB algorithm are presented. Section 4 presents the application developed for the integration and implementation of E-LSB substitution algorithm with an encryption algorithm. The results of performance evaluation study are also presented in this section. Finally, the paper is concluded in Section 5.

2. RELATED WORK

Due to its weaknesses and security-related concerns, the security of LSB has been heavily investigated in recent years. Statistical and visual attack methods which can be used when the classical LSB algorithm has been preferred are presented in [16]. When some changes are realized on the last bits by means of LSB substitution, there would be undesired deterioration effects on the original cover image and these can be used to detect the existence of hidden information. For instance, chi-square test which is based on frequency with which pixel values appear in an image is used for this purpose. To address the weakness of the classical LSB substitution algorithm against chi-square attacks, Pseudo Random Number Generators can be used for the selection of pixels used for embedding [17]. However, this method requires preliminary communication and for security-related goals, message length should be kept small.

To address the vulnerability of the classical LSB substitution against statistical analysis based steganalysis attacks, different improvements and enhancements were proposed. Sharp proposed a novel approach called Least Significant Bit Modulation (LSBM) based on a different principle than the classical LSB [18]. Instead of directly changing the last bits of a cover image, it performs embedding by increasing or decreasing the bit value by 1. Mielikainen improved LSBM to make it more robust against attacks and achieved to embed same amount of message by changing fewer pixels than LSBM. In the proposed approach, pixels are used as pairs and one bit of information is embedded into the last bit of the first pixel and the second one bit information is obtained from a function of the pixel pair [19]. Luo et al. claimed that plain and smooth areas in cover images are visually weak and provide low security for information embedding. To address this issue, they proposed a technique which can choose embedding regions and perform embedding based on the size of the message and difference between successive two pixels. When the message to be hidden is short, the proposed technique leaves the plain regions of the cover image as they are and performs embedding on the regions where the sharpness are dense and, therefore, increase the security [20]. Although the proposed technique performs well, it does not use plain regions for embedding and its embedding ratio is low. Mathkour et al. proposed a different approach to steganography by splitting cover images into equal blocks and applying LSB spirally [21]. They proved that their approach is considerably robust against the chi-square attack.

Yalman *et al.* handled information embedding problem with a different approach in which embedding is performed by weighing RGB (Red, Green, Blue) values [22]. They proved that the proposed approach achieves low visual degradation. Wang and Chen proposed a two-way block matching technique which is based on block matching of the cover image and hidden image [23]. By splitting the two images into m $\times$ n blocks, the highest similarity block to ith block in the cover image is found and matched, and this process is repeated. When the process has been completed, index information of well-matched blocks and unmatched blocks is compressed and stored in LSBs to be distributed to the cover image. The authors proved that the proposed technique provides high capacity and low noise [23]. In [24], Chen and Wang presented block matching technique to expand the block search field and achieved to lower the amount of information to be embedded in the block. The proposed technique achieves better image quality than the technique proposed in [23]. In the literature, in addition to LSB substitution, there are other steganographic techniques but, due to the focus of this study, only bit the substitution based techniques were included in the literature review.

Similar to the above mentioned approaches and techniques, in this study we focus on addressing the vulnerability of the classical LSB algorithm since LSB substitution is one of the most commonly used steganographic techniques due to its simplicity, speed and efficiency for most application scenarios. However, instead of completely eliminating the principle of the classical LSB algorithm, we propose an enhancement to it and in this way develop an enhanced form of it, called E-LSB. While keeping the main principle of LSB substitution algorithm, the proposed algorithm addresses the issues causing security-related concerns when the classical LSB is preferred. E-LSB keeps the number of modified bits at minimum and does not change the file size, and this way prevents severe image degradation to prevent the image from being visually detected that it includes an embedded message.

3. PROPOSED EMBEDDING ALGORITHM

The following subsections explain the theory and substitution technique used by the proposed approach and the details of the proposed approach along with the application developed in this study.

3.1 Least Significant Bit Substitution

Although LSB substitution minimizes degradation in image quality, message embedding using a high bit plane instead of the LSB plane of the images has the potential of degrading the image quality severely. Hence, instead of 8-bit gray scale cover images, colour cover images represented by three bytes one for each colour (Red, Green, Blue) are preferred for embedding long messages. For example, the letter 'B' has an ASCII code of 66, 1000010 in binary. To store a 'B' in a 24-bit image, three consecutive pixels are needed:

If the three consecutive pixels before the insertion are:

11110000.10110100.01000101, 00010101.00010011.01110111, 11000111.00110011.01010011

After the insertion, their final values will be:

*1111000**1**.1011010**0**.0100010**0**, 0001010**0**.0001001**0**.0111011**0**, 1100011**0**.0011001**1**.0101001**0***

Using the LSB substitution, a high payload can be embedded in a cover image by carrying one bit of the message to be hidden per byte of pixel data and this way embedding can be realized based on image size [13]. For higher payloads, to prevent unintended users from noticing that an embedding process has been held, larger cover images are preferred. Although, specific statistical techniques can be used to determine whether the LSB substitution has been performed on an image or not, the LSB substitution is mostly seemingly undetectable [13, 14]. On the other hand, the LSB substitution is extremely sensitive to filtering and manipulation operations. Some image manipulation operations including rotation, cropping, scaling, lossy compression and addition of noise can severely destroy the original message [5, 7].

3.2 Details of Enhanced LSB (E-LSB)

Basically, E-LSB is an enhanced form of LSB algorithm. Like LSB, E-LSB embeds a given text in the last bits as explained in the previous subsection. However, E-LSB does not embed the text to be hidden beginning with the first pixel of the cover image. A comparison of E-LSB and traditional LSB substitution algorithms are given in Table 1. E-LSB algorithm consists of three main steps: Encryption, Analysis, and Embedding.

- ☐ Encryption Step
- ☐ Analysis and Encryption Step
- ☐ Embedding Step

Encryption Step: It encrypts text or a text file given as input using an encryption algorithm. Currently, it allows the use of well-known encryption algorithms, i.e. RSA [25], AES [26] and CAESAR cipher [27]. However, other encryption algorithms can also be used in this step by being implemented in the proposed steganography approach. In this way, even if an unintended user detects that there is a hidden message in the cover image by means of a steganalysis tool, he/she will not be able to uncover the hidden message without the key.

Analysis and Encryption Step: It is the key step of the proposed approach since it analysis relative compression performance of the integrated encryption algorithms for a given message and decides which one is the best in terms of compression rate for the given message. In this way, embedding process can be performed with the least modification, i.e. the least number of bit changes, on the cover image.

Embedding Step: It hides the encrypted text in a cover image using the enhanced bit substitution algorithm explained in subsection 3.2.1. To be able to evaluate the relative performance of E-LSB and LSB substitution algorithms, we also integrated classical LSB algorithm into the developed software tool. Hence, the techniques that can be employed by this step are:

- ☐ Traditional LSB algorithm
- ☐ E-LSB algorithm

3.2.1 Enhanced Bit Substitution Algorithm

Before modifying the cover image, it determines the best starting point for embedding, i.e. the most suitable pixel to be used as the first pixel for embedding. Here, the goal is to keep the number of modified bits as low as possible. Therefore, the most suitable pixel ranges in the image which has the largest number of same bits with the encrypted message is used as explained in Figure 1. To be able to perform the reverse process in order to obtain the original message, the algorithm needs to know the starting pixel which has been used for embedding the encrypted original message. Before embedding the encrypted message, E-LSB algorithm places 1 byte zero bits' value in front of the starting pixel. After embedding the encrypted message, it places 1 byte zero bits' value at the end of the message, too.

Pseudocode of enhanced bit substitution algorithm

```
i=0, j=0, counter=0, k=0, textarray[] = int[text.length];
i to image.width
j to image.height

if (k<text.length&& k %3 ==0 &&textarray[k]==
bmp.GetPixel(i, j).R % 2)
{ bmp.SetPixel(i, j, Color.FromArgb((bmp.GetPixel(i, j).R -
bmp.GetPixel(i, j).R % 2) + textarray[k], bmp.GetPixel(i, j).G,
bmp.GetPixel(i, j).B)); )
counter++; k++;}// CONDITION I
if (k<text.length&& k %3 ==1 &&textarray[k]==
bmp.GetPixel(i, j).G % 2)
{ bmp.SetPixel(i, j, Color.FromArgb((bmp.GetPixel(i, j).R ,
bmp.GetPixel(i, j).G - bmp.GetPixel(i, j).G % 2) + textarray[k],
bmp.GetPixel(i, j).B));)
```

Table 1 A comparison of the traditional LSB substitution and E-LSB algorithm.

	Traditional LSB Algorithm	**E-LSB Algorithm**
Integration with an Encryption Algorithm	Traditional LSB substitution algorithm does not check which encryption algorithm is more suitable for a given message if integrated with encryption feature.	E-LSB algorithm analysis a given message to check which encryption algorithm is more suitable in terms of compression rate and employs it.
Starting Pixel for Embedding	LSB starts the embedding process using the first pixel of a cover image.	E-LSB does not embed the text to be hidden starting with the first pixel of the cover image. Instead it uses the best pixel for starting to embedding process and in this way image degradation is minimized.
Number of Modified Bits	Number of modified bits during an embedding process with traditional LSB substitution algorithm depends on message size.	E-LSB algorithm modifies fewer bits than traditional LSB substitution algorithm for message embedding.
Degradation of Image Quality	Traditional LSB substitution algorithm can severely degrade a cover image to embed a long message in some cases.	E-LSB algorithm degrades a cover image less than traditional LSB substitution algorithm.
File Size Change	Traditional LSB substitution algorithm can change the file size of cover images.	E-LSB algorithm does not change the file size of cover images. Therefore, the file sizes of original cover images and altered ones are identical.
Memory Usage	Compared to the proposed algorithm, traditional LSB substitution algorithm uses less memory.	E-LSB algorithm uses a bit more memory than traditional LSB substitution algorithm.
Processing Time	Traditional LSB substitution algorithm embeds a given message more quickly than E-LSB.	Due to its analysis step, E-LSB is slower than traditional LSB substitution algorithm.
Suitability for Devices with Limited Resources	Suitable	Suitable

```
counter++; k++;}// CONDITION II
if (k<text.length&& k %3 ==2 &&textarray[k]==
bmp.GetPixel(i, j).B % 2) { bmp.SetPixel(i, j,
Color.FromArgb((bmp.GetPixel(i, j).R , bmp.GetPixel(i, j).G,
bmp.GetPixel(i, j).B-bmp.GetPixel(i, j).B%2)+ textarray[k]);)
counter++; k++;}// CONDITION III
```

Pseudocode of enhanced bit substitution algorithm – Reverse operation

```
Procedure SolveElsbImage(Bitmap Embedded Image)
i=0; j=0;binaryText=" "; byteText=" ";
for i=0 to image.width
for j=0 to image.height
binaryText+=Convert.ToString(EmbeddedImage.
GetPixel(i,j).R%2);
if(binaryText.Length==8) //Null Control
if(binaryText.Equals("00000000"))
go to FINAL
else
byteText+=binaryText;
binaryText=" ";
end if
end if
binaryText+=Convert.ToString(EmbeddedImage.GetPixel(i,j).G%2)
if(binaryText.Length==8)//Null Control
if(binaryText.Equals("00000000"))
go to FINAL
else
byteText+=binaryText;
binaryText=" ";
end if
end if
binaryText+=Convert.ToString(EmbeddedImage.GetPixel(i,j).B%2)
```

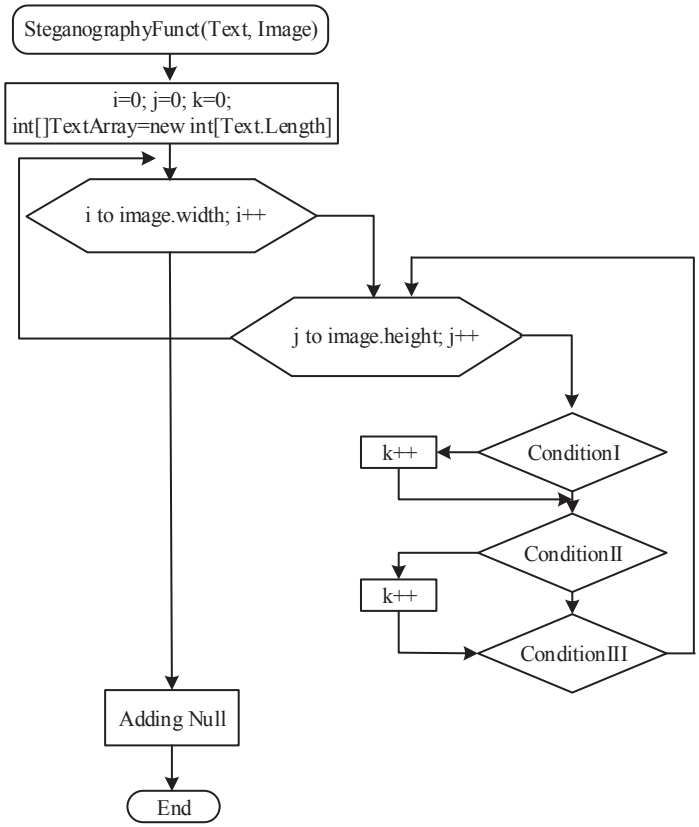

Figure 1 Flow of the enhanced bit substitution algorithm.

```
if(binaryText.Length==8)//Null Control
    if(binaryText.Equals("00000000"))
            go to FINAL
                else
        byteText+=binaryText;
          binaryText="";
              end if
          end if
      end of for
    end of for
    FINAL:
  return byteText;
    END
```

3.3 Discussion on Image Quality Degradation

Although, there are many methods for measuring the level of image quality degradation, Mean Squared Error (MSE) and Peak Signal to Noise Ratio (PSNR) are the most commonly used measures [28]. For message embedding algorithms, MSE enables the comparison of the pixel values of the original image with the altered one, i.e. the one in which a message has been embedded, and represents the average of the squares of the errors between the two images. The error is the amount by which the values of the original cover image differ from the altered one. After embedding step, higher PSNR values indicate that the altered image has been reconstructed to match the original cover image. The objective is to minimize the MSE between images with respect the maximum signal value of the image.

$$\text{MSE} = \frac{\sum_{M,N} I_1(m,n) - I_2(m,n)^2}{M * N} \tag{1}$$

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

where I_1 is the matrix data of the cover image, I_2 is the matrix data of the altered cover image, M is the number of rows of pixels of the images, N is the number of columns of pixels of the images, m is the index of that row, n is the index of that column, and finally R is the maximum signal value existing in the cover image.

4. PERFORMANCE EVALUATION

To evaluate the relative performance of E-LSB algorithm, we developed a software tool in C# to implement and combine it with a set of encryption algorithms. Figure 3 illustrates the workflow of the proposed approach and its main form is shown in Figure 4. As it is shown, it allows uploading of multiple cover images. A message entered by the user is first converted to binary system. Then, the message is encrypted with one of the embedded encryption algorithms. Finally, E-LSB substitution is performed and the software tool presents the results. By analyzing the obtained results, the user (if the tool is used in manual mode, it can also dynamically select the best compression algorithm without a user input) can select the best encryption algorithm depending on his/her needs. As shown in Figure 6, to analyze the relative performance of the embedded encryption algorithms and

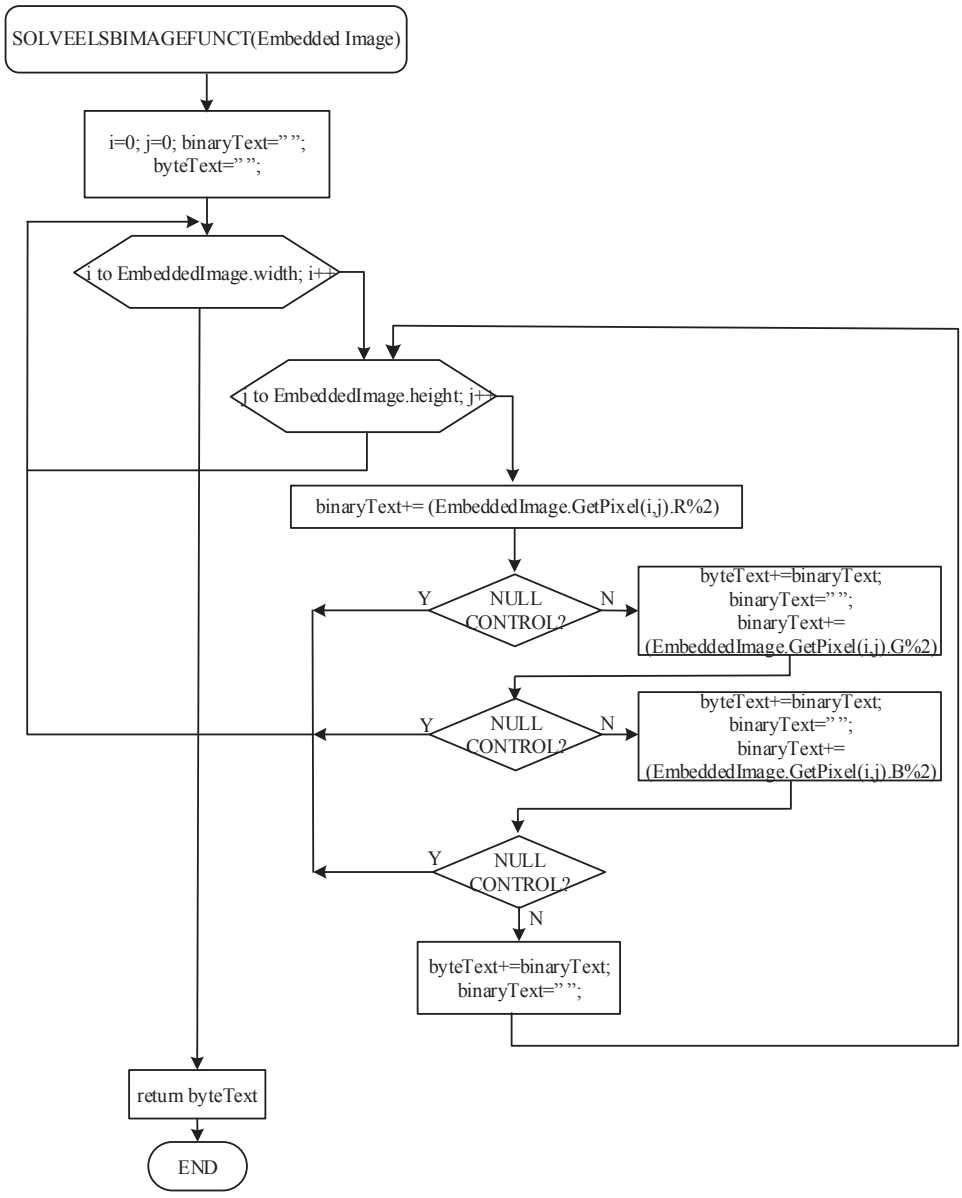

Figure 2 Flow of the enhanced bit substitution algorithm – Reverse operation.

the employed substitution algorithms, we calculated MSE and PSNR values for the case study shown in Figure 5. Based on the obtained results including MSE and PSNR values given in Tables 2–5, we can conclude that the proposed approach does not severely degrade the quality of cover images and performs better than the traditional LSB substitution algorithm.

5. CONCLUSION

Steganography is used to ensure confidentiality of sensitive information by means of information hiding. One of the common techniques of steganography is LSB. However, LSB has some weaknesses and its usability for sensitive information is seen as questionable by some people. In this study, we aim to address LSB's main disadvantage, being easy to detect, and make LSB stronger against steganalysis algorithms by enhancing it. The proposed algorithm called E-LSB is more difficult to be detected by steganalysis algorithms and can be integrated in various applications for encryption and information hiding purposes. E-LSB is superior to the traditional LSB algorithm. Different from LSB, it keeps the number of modified bits at minimum and does not change the file size. On the other hand, it is a bit slower since it performs several bit checks.

Although bit substitution algorithms are used to embed messages into cover images, they alone only provide basic security for information transfer since they do not encrypt the original messages. If they are used with encryption techniques, more secure information transfer can be achieved. Hence, in this study, we integrated E-LSB with well-known encryption algorithms to provide more secure information transfer. Using the software application developed in C# we showed the implementation and usability of the proposed approach. After final improvements, the software toolâĽ™s English version will be made freely available to the research community.

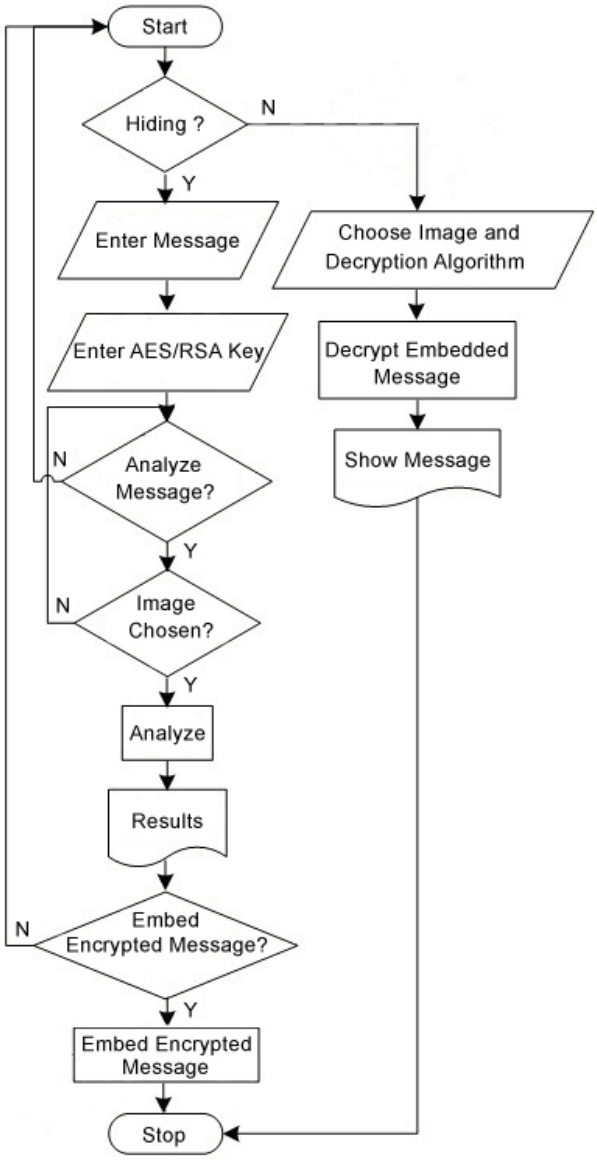

Figure 3 Illustration of the proposed approach.

Table 2 Relative performance of the substitution algorithms in terms of number of modified bits when combined with the encryption algorithms.

	Image 1		**Image 2**		Image 3	
Embedding/Encryption	***LSB***	***E-LSB***	***LSB***	***E-LSB***	***LSB***	***E-LSB***
AES	270	245	279	247	305	246
RSA	552	444	541	452	500	453
CAESAR	118	43	137	43	120	44

Table 3 Relative performance of the substitution algorithms in terms of percentage of modified bits when combined with the encryption algorithms.

	Image 1		**Image 2**		Image 3	
Embedding/Encryption	***LSB***	***E-LSB***	***LSB***	***E-LSB***	***LSB***	***E-LSB***
AES	0.043945313	0.039876302	0.045410156	0.040201823	0.049641927	0.040039063
RSA	0.08984375	0.072265625	0.088053385	0.073567708	0.081380208	0.073730469
CAESAR	0.019205729	0.006998698	0.022298177	0.006998698	0.01953125	0.007161458

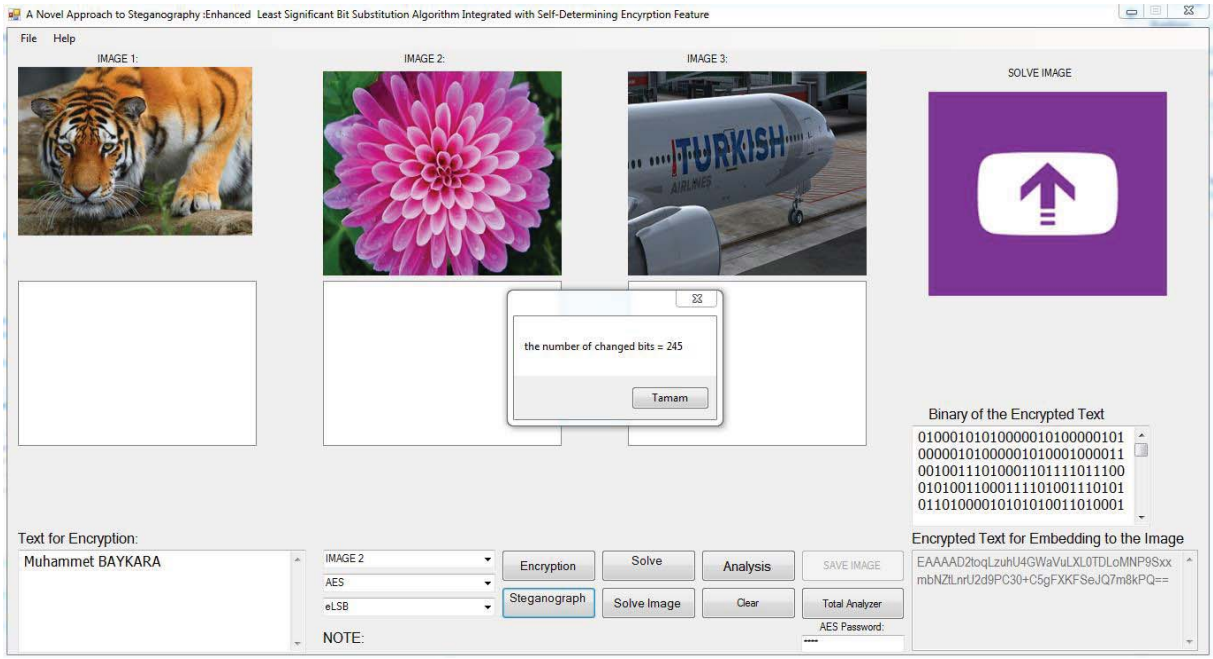

Figure 4 Software tool developed for the implementation of E-LSB algorithm.

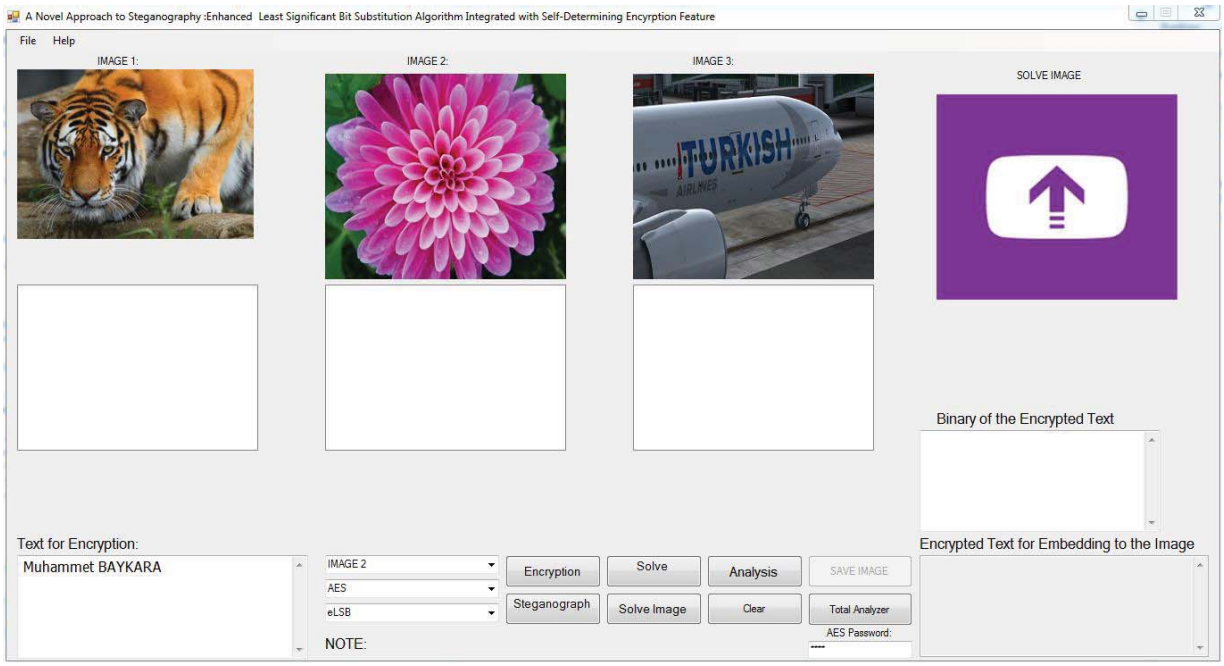

Figure 5 Case study for the relative performance evaluation of LSB and E-LSB substitution algorithms.

Table 4 Relative performance of the substitution algorithms in terms of MSE when combined with the encryption algorithms.

	Image 1		**Image 2**		Image 3	
Embedding/Encryption	***LSB***	***E-LSB***	***LSB***	***E-LSB***	***LSB***	***E-LSB***
AES	0.0742	0.0592	0.2450	0.0608	0.3375	0.0667
RSA	0.1508	0.1283	0.1400	0.1292	0.1450	0.1375
CAESAR	0.0150	0.0133	0.0192	0.0100	0.0192	0.0142

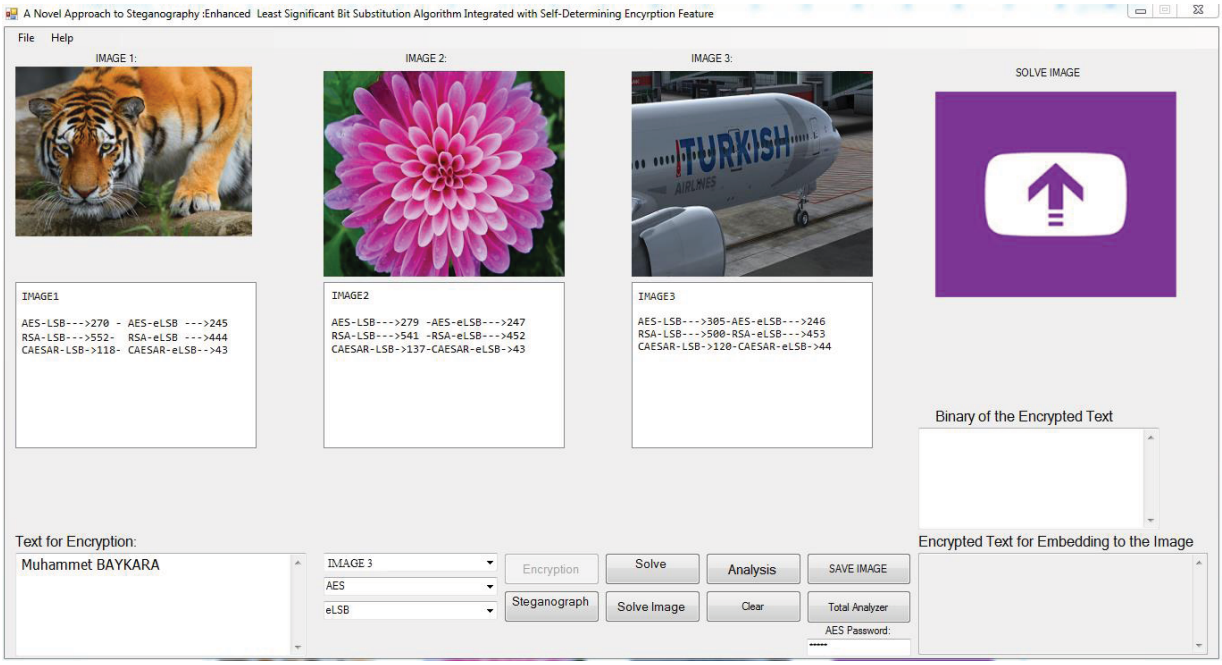

Figure 6 Results calculated for the uploaded cover images.

Table 5 Relative performance of the substitution algorithms in terms of PSNR when combined with the encryption algorithms.

	Image 1		**Image 2**		Image 3	
Embedding/Encryption	***LSB***	***E-LSB***	***LSB***	***E-LSB***	***LSB***	***E-LSB***
AES	77.4921	77.8980	72.2040	77.8627	71.0856	77.8803
RSA	74.5551	75.3158	74.5880	75.2382	74.6547	75.2286
CAESAR	85.2575	85.4549	83.4645	85.4549	83.7278	85.3551

REFERENCES

1. *Information Hiding Techniques for Steganography and Digital Watermarking* (1st ed.). S. Katzenbeisser and F. A. Petitcolas (Eds.). Artech House, Inc., Norwood, MA, USA, 2000.
2. I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker. 2007. *Digital Watermarking and Steganography* (2nd ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
3. N. F. Johnson, Z. Duric, and S. Jajodia. *Information Hidding: Steganography and Watermarking - Attacks and Countermeasures*, Boston, 2001.
4. Z. Duric, M. Jacobs, and S. Jajodia, Information Hiding: Steganography and Steganalysis, in Handbook of Statistics, vol. 24, pp. 171-187, 2005.
5. A. Cheddad, J. Condell, K. Curran, and P. Mc Kevitt, "Digital Image Steganography: Survey and Analysis of Current Methods," *Signal Processing*, vol. 90, no. 3, pp. 727-752, 2010.
6. R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley, New York, 1992.
7. E.T. Lin and E.J. Delp, "A review of data hiding in digital images," in *Proceedings of the Image Processing, Image Quality, Image Capture Systems Conference*, 1999, pp. 274-278.
8. İ. Karadogan and R. Das, "An Examination on Information Hiding Tools for Steganography," *International Journal of Information Security Science (IJISS)*, vol. 3, no. 3, pp. 200-208, 2014.
9. M. Baykara and R. Das, "A Steganography Application for Secure Data Communication," in *Proceedings of 10th International Conference on Electronics, Computer and Computation*, Turgut Ozal University, 7-9 November 2013, Ankara, pp. 309-313.
10. P. Bedi, R. Bansal, and P. Sehgal, "Using PSO in Image Hiding Scheme Based on LSB Substitution," *Communications in Computer and Information Science*, vol. 192, pp. 259-268, 2011.
11. P. Bedi, R. Bansal, and P. Sehgal, "Using PSO in a spatial domain based image hiding scheme with distortion tolerance," *Compuuters & Electrical Engineering*, vol. 39, no. 2, pp. 640-654, 2013.
12. S. Atawneh, "A New Algorithm For Hiding Gray Images Using Blocks, " in *Proceedings of the 2nd Information and Communication Technologies (ICTTA '06)*, 24-28 April 2006, Damascus, SYRIA, pp. 1484-1488.
13. R. Chandramouli and N. Memon, "Analysis of LSB based image steganography techniques," in *Proceedings of 2001 International Conference on Image Processing*, vol. 3, pp. 1019-1022, 2001.
14. K. Thangadurai and G. Sudha Devi, "An Analysis Of LSB Based Image Steganography Techniques," in *Proceedings of IEEE Computer Communication and Informatics*, 3-5 January 2014, Coimbatore, INDIA, pp.1-4.
15. P. Palanisamy and M. K. Rajagopal,"A Steganography Framework For Easy Secret Sharing Through Images," in *Proceedings of Second IEEE International Image Information Processing*, 9-11 December 2011, Shimla, INDIA, 309-312.
16. A. Westfeld and A. Pfitzmann, "Attacks on Steganographic Systems," in *Proceedings of the Third International Workshop on Information Hiding*, 1999, pp. 61-76.
17. C. Guyeux, Q. Wang, and J. M. Bahi, "A Pseudo Random Numbers Generators Based on Chaotic Iterations: Application to Watermarking," *Lecture Notes in Computer Science*, vol. 6318, pp. 202-211, 2010.
18. T. Sharp, "An Implementation of Key-Based Digital Signal Steganography," *Lecture Notes in Computer Science*, vol. 2137, pp. 13-26, 2001.

19. J. Mielikainen, "LSB Matching Revisited," *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 285-287, 2006.
20. W. Luo, F. Huang, and J. Huang, "Edge Adaptive Image Steganography Based on LSB Matching Revisited," *Information Forensics and Security*, pp. 201-214, 2010.
21. H. Mathkour, G. M. R. Assassa, A. Al Muharib, and I. Kiady, "A Novel Approach for Hiding Messages in Images," in *Proceedings of International Conference on Signal Acquisition and Processing*, 2009, pp. 89-93.
22. Y. Yalman, F. Akar, and I. Erturk, "An Image Interpolation Based Reversible Data Hiding Method Using R-Weighted Coding," in *Proceedings of IEEE 13th Int. Conference on Computational 8 Science and Engineering*, 2010, pp. 346-350.
23. R. Wang R and Y. Chen, "High-Payload Image Steganography Using Two-Way Block Matching," *IEEE Signal Processing Letters*, vol. 13, no. 3, pp. 161-164, 2006.
24. S. -K. Chen and R. -Z. Wang, "High-payload image hiding scheme using k-way block matching," in *Proceedings of Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2010, pp. 70-73.
25. R. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120-126, 1978.
26. J. Daemen and V. Rijmen, Vincent, "AES Proposal: Rijndael," National Institute of Standards and Technology. p. 1., March 9, 2003. Available at: http://csrc.nist.gov/archive/aes/rijndael/Rijndael-ammended.pdf [Accessed 21 February 2015].
27. R. Wobst. *Cryptology Unlocked*. Wiley, UK, 2001.
28. *Spatial Synthesis: Centrality and Hierarchy*. Volume I, Book 1. S. L. Arlinghaus and W. C. Arlinghaus. Institute of Mathematical Geography, Ann Arbor, 2005.