Enhanced Security: Implementation of Hybrid Image Steganography Technique using Low-Contrast LSB and AES-CBC Cryptography

Edwar Jacinto G, Holman Montiel A, Fredy H. Martínez S
Facultad Tecnológica, Universidad Distrital Francisco José de Caldas, Bogotá, Colombia

***Abstract*—Now-a-days, sensitive and confidential information needs to be exchanged over open, public, and not secure networks such as the Internet. For this purpose, some information security techniques combine cryptographic and steganographic algorithms and image processing techniques to exchange information securely. Therefore, this research presents the implementation of an algorithm that combines the AES-CBC cryptographic technique with the LSB steganographic technique, which is statistically enhanced by image processing by looking for low-contrast areas where the encrypted information will be stored. This hybrid algorithm was developed to send a plaintext file hidden in an image in BMP format, so the changes in the image are invisible to the human eye and undetectable in possible steganographic analysis. The implementation was performed using Python and its libraries PyCryptodome for encryption and CV2 for image processing. As a result, it was found that the hybrid algorithm implemented has three layers of security over a plaintext encrypted and hidden in a digital image, which makes it difficult to break the secrecy of the information exchanged in a stego-image file. Additionally, the execution times of the hybrid algorithm were evaluated for different sizes of plaintext and digital image files.**

Keywords*—Steganography; cryptography; LSB; low contrast areas; AES-CBC algorithm

I. Introduction

When sensitive or confidential information needs to be sent securely between two parties communicating over media with a high probability of attack, e.g., public, open, or unsecured networks such as the Internet, it is necessary to employ information security techniques to perform this exchange. Some techniques can be used for information hiding and others for information encryption [1]. In the case of information concealment, steganographic techniques allow hiding a secret message in a cover message in such a way that its existence is not detectable to others but only to the receiver of the information. In the case of information encryption, cryptographic techniques allow exchanging secret information between sender and receiver through the encryption and decryption of coded messages.

Additionally, when the information traffic on the Internet today is analyzed, it is evident that the conventional type of communication is based mainly on sending images and video, which is how images have come to be selected as a means to communicate secret information securely. Nowadays, there is literature on different image steganographic techniques for information concealment [2] [3], and different cryptographic techniques for information encryption [4]. Such studies classify the existing algorithms, indicate the performance parameters, and show the advantages, possible applications, and attacks or security problems they may present [5] [6].

Thus, some of these studies have concluded that one way to improve information security, increasing the reliability, robustness, and solidity in the exchange of information is to combine steganographic techniques with cryptographic techniques [1] [7]. One way to do this is to take the sensitive or confidential information to be transmitted to perform an encryption process by implementing some cryptographic technique, and then take the encrypted message to perform a mixing process with a cover image using some steganographic technique [8].

Some examples of this are: [9] where the message to be transmitted is encrypted in two stages, the first by Caesar cipher and the second by chaos theory; the encrypted message is embedded in the cover image using the Least Significant Bit (LSB) substitution steganographic algorithm. [10] where the message to be transmitted is encrypted using Advance Encryption Standard (AES) encryption; at the same time, the cover image is preprocessed to resize it and identify the areas where the LSB substitution process was performed using inverse Wavelet Transform and Artificial Neural Networks (ANN). The author in [11] where the message to be transmitted is encrypted using AES encryption, including a hash process; this hash encrypted text is embedded in a cover image through Dynamic Octa Pixel Value Differencing (DOPVD) embedding algorithm that includes LSB + PVD approach. The author in [12] where the image to be transmitted is encrypted using a large secret key through XOR operation; the encrypted image is embedded in a cover image by LSB obtained a stego-image; finally, the stego-image is watermarked in time domain and frequency.

For this reason, this research aims to implement an algorithm that combines cryptographic and statistically enhanced steganography techniques for sending plain text files over a digital image in BMP format. That algorithm develops using Python and the OpenCV libraries as the base implementation language, considering the size restrictions of such information to be hidden as well as the resolution of the cover image [13].

In the case of cryptographic technique, it was decided to use Advanced Encryption Standard (AES) as the encryption

method. Because it is the standard cipher [14] [15] [16], given its security level, information encryption speed (capacity), and current availability in the internal architecture of processors as a dedicated hardware block [17] [18] [19], making it native in any application [20]. The only configurable parameter on AES is the cipher operation mode, which is associated with the order in which the keys and the initialization vector are combined with the information to be encrypted. Therefore, in this case, the Cipher Block Chaining (CBC) operation mode was chosen.

In the case of the statistically enhanced steganographic technique, it was decided to use the Least Significant Bit (LSB) substitution as a base method [21] [22] [23], enhanced in terms of selecting the information hiding areas. Such enhancement is achieved by using image processing techniques to choose a low contrast area [24], where the image entropy is less affected [25], offering a robust solution in terms of a possible stego-analysis. The image processing technique uses applied statistics concepts as mathematical criteria for locating the hidden and encrypted information. For this purpose, it is based on the characteristics of the analyzed images as a random variable, where the histogram's high dispersion can measure an image's high contrast. That is, the higher the contrast of the stego-image, the higher the security level given by this extra layer based on the processing and analysis of digital images (PAID).

Therefore, this paper presents the implementation and validation of a hybrid crypto-steganographic system. Section II describes the structure proposed to implement the hybrid system highlighting the three main elements. Section III explains the development of the software application step by step, showing: how the user key is entered, how the information encryption process, how the area where the encrypted message will be hidden is chosen, and how the execution of the LSB algorithm to reach the output stego-file. Section IV presents the validation of the implemented hybrid system and performs a performance analysis of the complete application's processing time. Finally, in Section V the conclusions according to these results are shown.

II. Methodology

The proposed structure is a hybrid technique that combines the AES-CBC cryptographic technique with enhanced LSB steganography to hide the information in the lower contrast area of the image, as shown in Fig. 1. The upper part of the graph shows the plaintext encryption process corresponding to the sensitive or confidential information to be transmitted. The lower part shows the processing of the cover image to determine the low contrast area. The right side of the figure shows the embedded process of the cipher text using a classical LSB technique to conceal it in the low contrast area, obtaining a stego-file containing the encrypted information immersed in the image (called Stego-Image).

This proposed structure complies with the philosophy of Feistel networks, which is none other than having the same architecture of the solution for the encryption and decryption of the information. It is a reversible structure where it is only necessary to reverse the order of the blocks to carry out the decryption process.

For the implementation, an application was made in Python 3.X, using the OpenCV libraries for image processing, in addition to using Numpy to work with vectors and matrices and MatplotLib to visualize the partial and final results. In this case, the information to be encrypted and hidden is a plain text encoded in UTF-8, to which a data type change process must be performed to be encrypted with AES in a CBC operation mode, always working with a pure binary string or in 64 bits format. Finally, the Cryptodome library was used, which has every one of the cryptographic functions necessary to encrypt the information.

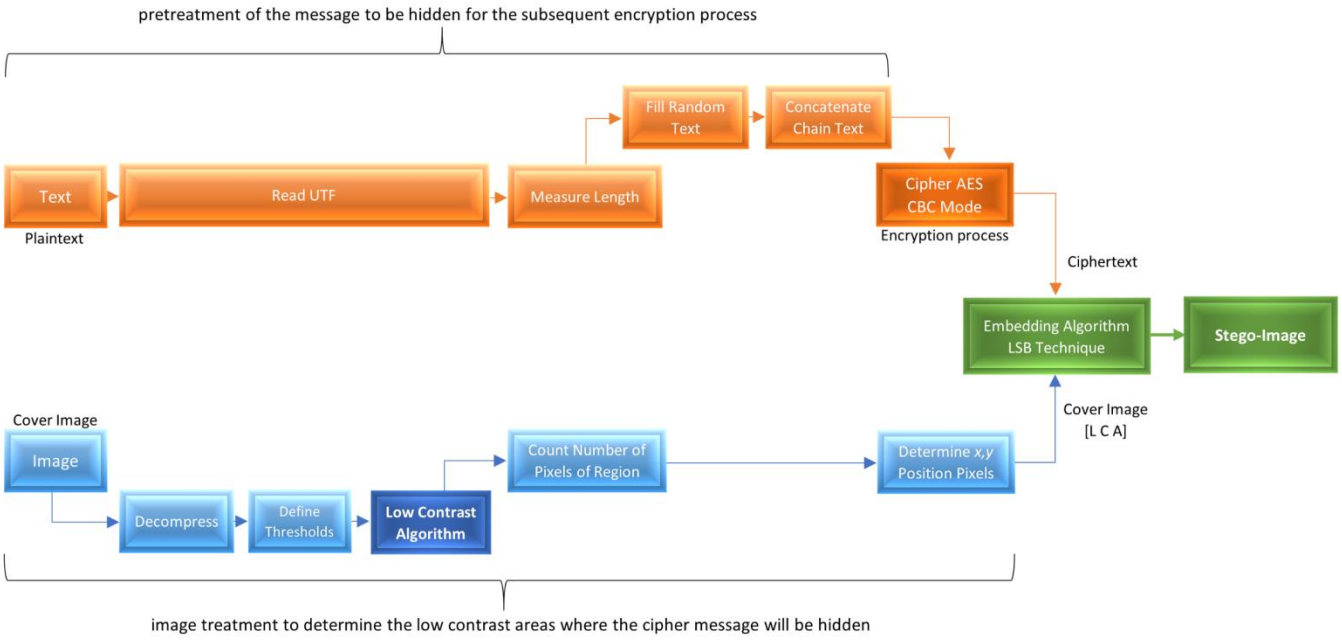


Fig. 1. General Block Diagram Implementation.

A. User Key Expansion

The aim is to encrypt the plain text using a standard block cipher such as AES-CBC; this algorithm requires a key and an initialization vector. For that, a key expansion process must be performed, and then the encryption of this key with an asymmetric algorithm [26].

Since block ciphers must have a key size equal to the block's size to be encrypted, a minimum procedure is required to ensure that the size of the session key is entered by the person or entity that will perform the steganography process complies with this characteristic. For this purpose, a key expansion procedure is performed, using the functions of the Cryptodome library developed for Python in its version 3.0 or higher, employing the PKCS #5 key expansion process, shown in the pseudo-code of Fig. 2.

This block cipher algorithm generates a series of subkeys to increase the entropy of the encrypted information; this process of generation and mixing of the subkeys with the information to be encrypted corresponds to the CBC mode of operation. This mode of operation generates high entropy since, in each of the cipher rounds, the K-th subkey is mixed with the information, obtaining at the end of the encryption process a new key that is mixed with the initialization vector. This process is automated and standardized, achieving greater security in the encryption of the information.

B. Encryption Process of the Plaintext Message

Once the operation mode of the encryptor, its working mode, the key of the indicated size, and the implicit generation of the initialization vector are clear, the information encryption process must be clearly understood to achieve its compatibility with the image file. In this case, the size of the plaintext file must be clear to be encrypted, what type of data is necessary to achieve the encryption process, and generate empty vectors where the encrypted information will be returned. For this process, it must have some of the tools offered by the NumPy library.

Next, the process of reading the text size to be encrypted is performed and compared with the number of pixels chosen to blend the ciphertext with the image. It must be guaranteed that the number of characters of the input text will never be the same as the number of pixels; therefore, a random filler text must be generated so that no empty spaces are created in the output stego-file. This process is done to avoid the simple detection of hidden information. Fig. 3 shows in a simple pseudo-code this programming scheme.

Once the character string is ready to perform the information encryption process, taking into account the dimensions and type of data required, an algorithm is applied that selects the low contrast areas where it is more difficult to detect the hidden information.

C. Statistical Method: Choice of the Low Contrast Area

The aim is the cover image file processing using applied statistics concepts to identify the areas of lower contrast in the image, areas where the encrypted information will be placed. In other words, a process of selecting in which part of the image to place the hidden information is made. This task requires using image processing functions available in the OpenCV (CV2) libraries.

According to [24], an algorithm is used to detect some pixels with low contrast, complying with the following criteria to detect the areas where the human eye does not detect any change:

The criterion for detecting dark areas is in the image. This criterion is described in (1). It requires determining a working window, the number of pixels of the convolution matrix. For establishing a series of local medians $m_{s_{xy}}$, which will be compared with the global median of the entire image m_G using a weighting constant k_0, where the value of this constant depends on the gray level that will be given as "dark."

$$m_{s_{xy}} \leq k_0 \cdot m_G \quad (1)$$

```
1.  Input = KeySession //the user enters the key
2.  Salt = str(rnd(N_std)) //generate the random number
3.  for I in 0 to N_times
4.      AES_KEY = HASH (Input + Salt) // calculate the digest N times to the
5.  End
```

Fig. 2. Pseudo-code for user Key Expansion.

```
1.  PlainText = open (AnyFile.txt.encode(UTF-8)) // Open File
2.  SizeText = len(PlainText*8) //8 bits per character  // calculate length per char to pixel
3.  ImgPixels = CountPixels (rows*columns)  //measure length of the image
4.  Bits_Array = zeros(ImgPixels) // create an empty array
5.  Len_Random_str = round (ImgPixels - SizeText / 8) //calculate length of the fill text
6.  For I in range (0, len_random_str):
7.          Random_str[i] = random('a','z') //generate random char
8.  End
9.  Plain_text_fill = plain_text +random_chain //create the final char array
10. Plain_Text_Bytes = str.encode.bytes(plain_text_fill) // To Cypher plaint text in bytes only
```

Fig. 3. Pseudo-code for Encryption Process of the Plaintext Message.

The possible selection criterion is for finding low contrast areas. This criterion is described in (2). It compares the local standard deviation in a certain pixel window $\sigma_{s_{xy}}$ with the global standard deviation σ_G of the whole image by taking as a weighting factor or comparison criterion a factor k_2. This factor is determined by the experience of how scattered are the grayscale values in the low contrast regions to be detected.

$$\sigma_{s_{xy}} \leq k_2 \cdot \sigma_G \quad (2)$$

On the other hand, a possible error is generated in the selection criterion that [24] describes as enhancing a constant area, where the standard deviation would be zero evidently. Such a problem must be applied depending on the characteristics of the chosen image and is described by (3).

$$K_1 \cdot \sigma_G \leq \sigma_{s_{xy}} \quad (3)$$

This equation describes the way to compare a minimum local standard deviation in a certain pixel window $\sigma_{s_{xy}}$ with the global standard deviation σ_G with a factor K_1, avoiding enhancing or selecting constant zones. In other words, it becomes undesirable to select a pixel from a zone with the same gray level as a candidate for the LSB algorithm.

The algorithm applying the above mathematical criteria in practice was implemented through a pseudo-code, shown in Fig. 4. As a result, a binary matrix is obtained, which clearly identifies the low contrast zones where the LSB information mixing algorithm will be used to obtain the stego-image with the concealed information in these specific zones.

D. LSB Method (Least Significant Bit)

It is a method that seeks to place a binary string with the information to be hidden in the stego-file. In this case, having encrypted information, the binary string will be in a pure binary format or base64; these types of data result from the encryption process. Then, mixing or embedding this information in the cover image is performed through a simple binary mask. Fig. 5 shows the pseudo-code that mixes or embeds the encrypted information in the least significant bit of the cover image in the areas chosen by the statistical algorithm (low contrast areas).

It is necessary to ensure the correct functioning of the LSB algorithm that an image with large low-contrast areas compared to the total image size should be chosen. On the other hand, it is recommended to work with images of a size larger than the possible size of the plaintext to be encrypted and hidden in the digital image. In other words, the size of the stego-image of the input image should be much larger than the stego-message, which should be cipher using an encryption algorithm.

For this case, it was chosen only to hide plaintext files since it requires less processing than applying such processing to multimedia files. However, if the application requires it, the same technique can be applied to other types of files or combinations.

```
1.  Img = cv2.read(ImFile.Imext) //read the image
2.  ImgGray = cv2.cvtColor(Img, cv2.COLOR_BRG2GRAY) //Convert to only one matrix
3.  GlobalMedian = np.mean(ImgGray)  // calculate global median
4.  GlobalDevStd = np.std(ImgGray) // calculate global standard deviation
5.  Pix = NWin
6.  Dim = (pix,pix)
7.  MConv = np.ones(Dim) * 1/(pix**2)  // create convolution matrix
8.  LocalMedian = conv2(ImgGray,Mconv) // calculate local median
9.  LocalDevStd = conv2((ImgGray - LocalMedian) .^2 , MConv))  // calculate local standard deviation
10. ImgOut1 = LocalMedia <= K0*GlobalMedian  // compare to create binary matrix of dark regon
11. ImgOut2 = LocalDesvStd <= k2* GlobalDesvStd // compare to create binary matrix of low contrast
12. ImgOut = BitAND(ImgOut1, ImgOut2) // Binary matrix of the choose pixels
```

Fig. 4. Pseudo-code for Statistical Method - Choice of Low-contrast Area.

```
1.  Def Func_LSB(msg): // define LSB function
2.      for CharIn in(msg):
3.          o = CharIn  //// read the encrypted message
4.          for Column in (Img):  // traverse the columns
5.              for Row in (Img): // traverse the rows
6.                  if ImgOut(Row,Column) == (TRUE) // check if the pixel is choose to lsb
7.                      for in range (8)
8.                          O & lsb(Img) // put the information in the 8 pixels
```

Fig. 5. Pseudo-code for LSB Method.

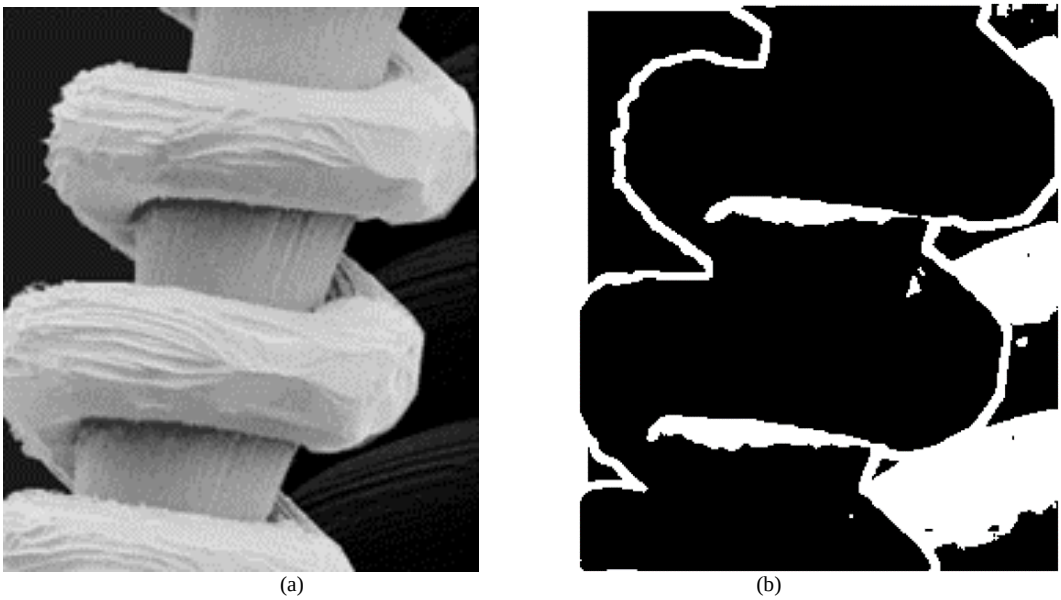
III. Results

The first step to verify the algorithm's effectiveness is to verify that the statistical algorithm effectively identifies the areas with low contrast. Fig. 6 shows in part (a) the original image with an area of low contrast and in part (b) a black and white image, where the white parts are the areas of the pixels chosen to perform the information hiding process using the LSB algorithm. For this part, it was only necessary to follow the steps of equations (1) and (2).

As a result, the stego-image does not look the same in its least significant bits as the original cover image, although this is not as visible to the human eye. The image loses its natural entropy, i.e., the bright part of the image is removed. This characteristic serves as an indication to discover whether a stego-image has hidden information in a steganalysis process.

The information encryption and hiding process were tested using the AES-CBC algorithm for encryption and the enhanced LSB algorithm that mixes o embeds information in the low contrast areas; this test was made using different sizes of plaintext files. Fig. 7 shows a bar graph that presents the execution time depending on the size of the information to be encrypted and hidden. The tests were performed on a PC with an 8-core Core i7 with 16 Gigabytes of RAM and a Geforce GTX 610 video card.

It can be seen how the application can store files of different sizes in the stego-image, up to a limit of one Megabyte, the size of the book Don Quixote in plaintext format, for which the processing time was approximately one minute. On the other hand, the time for information smaller than 200 kilobytes is less than 10 seconds, so it could be said that the process is agile for small texts.



(a) (b)

Fig. 6. (a) Original Image. (b) Image with Low-contrast Areas in White [24].

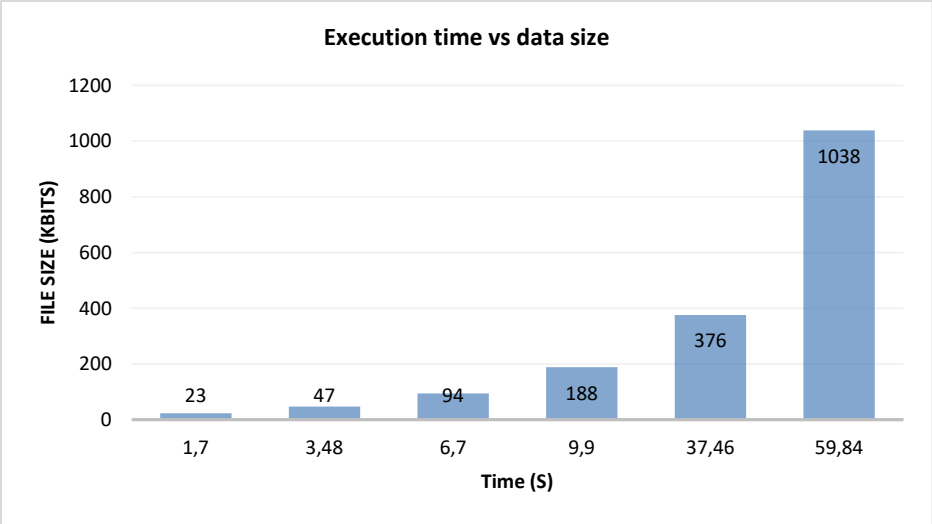


Fig. 7. Performance Graph of the Algorithm Measured on Text Files of Different Sizes.

IV. Conclusion

It was verified that the step-by-step implementation of a hybrid algorithm that combines cryptographic and steganography techniques for sending plaintext files over a digital image in BMP format gives a double layer of security. So, if the stego-image is revealed to have hidden information, it is impossible for the person or entity intercepting the message to know plaintext content because it is encrypted.

On the other hand, it was determined that using a statistically enhanced steganography technique to choose the lower contrast areas to hide the encrypted information only in these zones gives the hybrid algorithm an extra layer of security, making the entire algorithm more robust. The choosing lower contrast areas algorithm makes that the entropy of the image is only affected in the areas chosen by it. This feature makes it difficult to detect concealed information. It adds an extra layer of security since, besides having the session key to decrypt the ciphertext, the value of two constants, k_0 and k_2, must be present so that when performing the decryption process, the information only is taken from these areas. Therefore, it is verified that only modifying the least significant bit does not affect the statistical selection criteria with which the pixels in which the encrypted information was hidden were chosen.

Regarding the cryptographic technique implemented to encrypt the information before hiding it, the standardized AES-CBC algorithm was used, which was automated using the Cryptodome library, achieving greater security in the encryption of the information. However, it became evident that it would be possible to experiment with different combinations of standardized modes for AES in the Cryptodome library for future work. Seeking to maximize the entropy in the information and therefore generate fewer possible patterns in the LSB algorithm, as well as specifying how the key and the comparison constants would be exchanged.

Finally, the algorithm's performance was analyzed regarding the time used for the encryption and embedded process, resulting in fast usability for small plaintext files below 200 kilobytes. It is a good performance considering that a 100 kilobytes text is the entire chapter of any chapter literature text.

Acknowledgment

The Universidad Distrital Francisco José de Caldas supports this work through the research group SIE -Embedded Informatics Security- which belongs to the Technological Faculty. SIE has dedicated to working in cryptography and applied steganography. Currently, the bases are being generated to implement this type of algorithm in stand-alone applications, which is the final purpose of the workgroup.

References

[1] M. S. Taha, M. S. Mohd Rahim, S. A. Lafta, M. M. Hashim, and H. M. Alzuabidi, “Combination of Steganography and Cryptography: A short Survey,” in IOP Conference Series: Materials Science and Engineering, Jun. 2019, vol. 518, no. 5. doi: 10.1088/1757-899X/518/5/052003.

[2] A. O. Vyas and S. v Dudul, “An Overview of Image Steganographic Techniques,” International Journal of Advanced Research in Computer Science, vol. 6, no. 5, pp. 67–72, 2015, Accessed: Jun. 30, 2022. [Online]. Available: http://www.ijarcs.info/index.php/Ijarcs/article/viewFile/2483/2471

[3] S. G. Shelke and S. K. Jagtap, “Analysis of spatial domain image steganography techniques,” in Proceedings - 1st International Conference on Computing, Communication, Control and Automation, ICCUBEA 2015, Jul. 2015, pp. 665–667. doi: 10.1109/ICCUBEA.2015.136.

[4] G. C. Kessler, “An Overview of Cryptography,” Jun. 2010. [Online]. Available: www.garykessler.net/library/crypto.html.

[5] D. Laishram and T. Tuithung, “A Survey on Digital Image Steganography: Current Trends and Challenges,” May 2018. Accessed: Jun. 30, 2022. [Online]. Available: https://ssrn.com/abstract=3171494.

[6] B. Jana, M. Chakraborty, T. Mandal, and M. Kule, “An Overview on Security Issues in Modern Cryptographic Techniques,” May 2018. Accessed: Jun. 30, 2022. [Online]. Available: https://ssrn.com/abstract=3173527.

[7] A. Jan, S. A. Parah, M. Hussan, and B. A. Malik, “Double layer security using crypto-stego techniques: a comprehensive review,” Health and Technology, vol. 12, no. 1, pp. 9–31, Jan. 2022, doi: 10.1007/s12553-021-00602-1.

[8] S. Almuhammadi and A. Al-Shaaby, “A Survey on Recent Approaches Combining Cryptography and Steganography,” in Computer Science & Information Technology (CS & IT), Feb. 2017, pp. 63–74. doi: 10.5121/csit.2017.70306.

[9] G. S. Charan, Nithin Kumar S S V, Karthikeyan B, Vaithiyanathan V, and Divya Lakshmi K, “A novel LSB based image steganography with multi-level encryption,” in 2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Mar. 2015, pp. 1–5. doi: 10.1109/ICIIECS.2015.7192867.

[10] K. S. Seethalakshmi, Usha B A, and Sangeetha K N, “Security enhancement in image steganography using neural networks and visual cryptography,” in 2016 International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS), Oct. 2016, pp. 396–403. doi: 10.1109/CSITSS.2016.7779393.

[11] S. Mangela, N. Daddikar, T. Bargode, and P. N. Tatwadarshi, “Advance steganography using dynamic octa pixel value differencing,” in 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Mar. 2017, pp. 1–7. doi: 10.1109/ICIIECS.2017.8275989.

[12] M. Abdur, R. Ahmed, M. Adnan, and A. Ahmed, “Digital Image Security: Fusion of Encryption, Steganography and Watermarking,” International Journal of Advanced Computer Science and Applications, vol. 8, no. 5, 2017, doi: 10.14569/IJACSA.2017.080528.

[13] T. Morkel, “Self-sanitization of digital images using steganography,” in 2015 Information Security for South Africa (ISSA), Aug. 2015, pp. 1–6. doi: 10.1109/ISSA.2015.7335073.

[14] P. Sethi and V. Kapoor, “A Proposed Novel Architecture for Information Hiding in Image Steganography by Using Genetic Algorithm and Cryptography,” Procedia Computer Science, vol. 87, pp. 61–66, 2016, doi: 10.1016/j.procs.2016.05.127.

[15] M. E., A. A., and F. A., “Data Security Using Cryptography and Steganography Techniques,” International Journal of Advanced Computer Science and Applications, vol. 7, no. 6, 2016, doi: 10.14569/IJACSA.2016.070651.

[16] R. Indrayani, H. A. Nugroho, R. Hidayat, and I. Pratama, “Increasing the security of mp3 steganography using AES Encryption and MD5 hash function,” in 2016 2nd International Conference on Science and Technology-Computer (ICST), Oct. 2016, pp. 129–132. doi: 10.1109/ICSTC.2016.7877361.

[17] I. Algredo-Badillo, F. R. Castillo-Soria, K. A. Ramírez-Gutiérrez, L. Morales-Rosales, A. Medina-Santiago, And C. Feregrino-Uribe, “Lightweight Security Hardware Architecture Using DWT and AES Algorithms,” IEICE Transactions on Information and Systems, vol. E101.D, no. 11, pp. 2754–2761, Nov. 2018, doi: 10.1587/transinf.2018EDP7174.

[18] V. Sharon, B. Karthikeyan, S. Chakravarthy, and V. Vaithiyanathan, “Stego Pi: An automated security module for text and image steganography using Raspberry Pi,” in 2016 International Conference on Advanced Communication Control and Computing Technologies

(ICACCCT), May 2016, pp. 579–583. doi: 10.1109/ICACCCT.2016.7831706.

[19] A. Odeh, K. Elleithy, and M. Faezipour, "Fast real-time hardware engine for ZWC text steganography," in 2014 4th International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE), May 2014, pp. 1–5. doi: 10.1109/VITAE.2014.6934454.

[20] D. Das and M. Dutta, "Security Enhancement on Application Oriented Steganographic Schemes with Crypto-Encryption: A Technical Review," TIU Transactions on Intelligent Computing (TTIC), vol. IV, Dec. 2020, [Online]. Available: https://www.researchgate.net/publication/358280401.

[21] P. Martí Méndez Naranjo and D. Fernando Avila Pesantez, "Cryptography application experience to improve security in a steganographic method in images," Revista Espacios, vol. 40, no. 38, Nov. 2019, Accessed: Jul. 01, 2022. [Online]. Available: https://www.researchgate.net/publication/350153611.

[22] S. Raniprima, B. Hidayat, and N. Andini, "Digital image steganography with encryption based on rubik's cube principle," in 2016 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC), Sep. 2016, pp. 198–201. doi: 10.1109/ICCEREC.2016.7814972.

[23] S. L. Chikouche and N. Chikouche, "An improved approach for lsb-based image steganography using AES algorithm," in 2017 5th International Conference on Electrical Engineering - Boumerdes (ICEE-B), Oct. 2017, pp. 1–6. doi: 10.1109/ICEE-B.2017.8192077.

[24] R. C. Gonzalez and R. E. Woods, Digital Image Processing, Third. Pearson, 2008.

[25] R. Roy and S. Changder, "Image steganography with block entropy based segmentation and variable rate embedding," in 2014 2nd International Conference on Business and Information Management (ICBIM), Jan. 2014, pp. 75–80. doi: 10.1109/ICBIM.2014.6970937.

[26] E. J. G. et al. , Edwar Jacinto Gómez et al., "Implementation of a Crypto-Steganographic System Based on the Aes-Cbc Algorithm," International Journal of Mechanical and Production Engineering Research and Development, vol. 10, no. 3, pp. 15059–15068, Jun. 2020, doi: 10.24247/ijmperdjun20201435.