ISSN: 0974-5823 Vol. 7 No. 1 January, 2022

International Journal of Mechanical Engineering

Information Hiding Using Steganography

Hasan Al-Furiji
Department of Physics, College of Science, University of Misan, Iraq.

Nsaif Jasim Hadi
Department of Petroleum, College of Engineering, University of Misan, Iraq.

Ali Dhahir Mohsin
Department of Petroleum, College of Engineering, University of Misan, Iraq.

Abstract **- Steganography refers to the practice of concealing the fact that correspondence is taking place by encasing the data in other data. Steganography technology used in various applications has different specifications. There are many ways to hide data, including reverse data hiding, which is a strategy that allows data to be embedded within an image and then recovered, as well as an exact copy of the host image. This technology is ideal for both medical and military uses because it is lossless. Security is increased in this diagram by masking the STImage inside another image, which improves security while reducing distortion. The data in the image is obscured using the Least Significant Bit (LSB) technology, which preserves the original STImage consistency. We used multi-level embedding in this proposed framework, where the hidden text is first inserted into an image. STImage is then obscured once more inside another image.**

Keywords – Cover Image (COImage), Embedding Data (EmData), LSB method, Steganography, Stego-Image (ST Image).

INTRODUCTION

Due to the exponential growth of the World Wide Web, information security has become one of the most significant connectivity considerations. Cryptography began as a means of securing the confidentiality of data. Unfortunately, keeping the contents of a message confidential isn't often enough; often it's often important to maintain the message's life hidden, and the term responsible for this is known as Steganography [1].

Steganography is a centuries-old art of incorporating personal data into other data using a set of laws and techniques. Unauthorized users would be unable to see or read about the EmData as a result. The term steganography refers to the use of a hidden path to transmit data. If both methods are used, Cryptography and Steganography is used to encrypt the connection [2]. The primary distinction between Cryptography and Steganography is that Cryptography is concerned with maintaining the secrecy of the message's contents, while Steganography is concerned with maintaining the secrecy of the message [3]. Multimedia objects, such as images, audio, and video, are perhaps the most common cover media. The emphasis of this paper is on photographs as cover media. Watermarking and fingerprinting [4] are two other innovations that are closely related to Steganography.'

The main application of the Steganographic approach is in the field of covert contact. It can be used by intelligence services all over the world to share highly sensitive data in covert media. For example, undercover officers can use Image Steganographic software to hide a valuable secret map in an image, and a second officer can easily download the image and recover the hidden map [5].

The manuscript description is as follows. The second section looks at certain elements of reversible data embedding (RDE) and the methods that have been used to EmData in media files so far. The LSB technologies used in embedding and the techniques used in this technique to store data are described in Section Three. The presenter in Section Four gives a quick overview of the new RDE technology. Section 5 defined a technique for dramatically improving efficiency. Section Six contains the findings of these experiments. Section Seven ends with a review of the approaches discussed as well as any research proposals for the future.

REVERSIBLE DATA HIDING (RDH)

The RDH scheme is a technology that allows EmData to hide an image and then retrieved it as required, as well as an exact copy of the original host image. Any conventional RDH systems use additive modular computational methods and a diffuse range to achieve their results [6]. Honsinger's method, which uses the 256 addition paradigm as an invertible operation, is an example of one of these schemes. RDH algorithms focused on computational units have noticeable salt and pepper artifacts and delay watermark retrieval, despite the fact that some of these schemes are robust. Graph transforming strategies have been suggested to improve the robustness of the reversible watermark and reduce the optical effect of salt and pepper on the graphs. The block diagram of the image replaces the embedding goal in this diagram [7]. The circle representation diagram suggested by Vlees Chouwer is a clear example of a

diagram. While this method of steganography produces a higher-quality embedded image, it has a lower embedding capability [8]. Methods that losslessly compress a series of chosen features from an image and embed the payload in the space saved due to the compression fall under a separate range of data hiding schemes. This kind has a greater embedding capability than the previous two types [9]. Celik's simplified LSB embedding algorithm, which is based on grouping pixels and EmData bits into the state of each group[10], is another scheme of this sort. Tian suggested a slightly different scheme that involved changing the difference between two pixel values while keeping the mean constant. His method splits the image into pairs of pixels, with one piece of data embedded in each pair. Not all pairs are appropriate for hiding. Expandable pairs should only be generalized if they do not cause flow or downstream errors [9].

LEAST SIGNIFICANT BIT (LSB) TECHNIQUE

This term only uses the least significant pixel alpha fraction. No color value is changed in this case. The message length must be written in the image before the message is embedded. Bit 0 from the first 32 pixels is taken. To know the message length contained in the image[12], the bits must be specifically organized inside an integer vector. The 32-pixel pixels save the bits needed to reconfigure the byte value to produce the original string. A message containing a limit of 1.24.996 characters may be stored on an image of 1 million pixels (or 1 megapixel). / 8 = 1,34,996. ((100,000 - 31)). Although a 1 mega-pixel image is known to be a low-resolution image, many of its characters can be saved as a text message (TxTMsg) [13]. The scale limit message can be determined with the formula n=(P-32)/8 by means of a pixel image of 1 bit per pixel. If we extend storage positions to the smallest bit of all four of the ARGB system's components (pixels 0, 8, 16 and 24) the storage space would increase to n = (4P-8)/ 8. Here, n is the maximum duration for the message And P is the pixel count. Confidential data was initially incorporated into the original image using the LSB substitution technology on the sender side. This is why STImage is produced. Another COimage has now been selected to cover the STImage. Thus multiple encryption is performed here in order to increase the security[14]. At the receiver side, the image which is received from the sender is first decrypted to get STImage. STImage is decrypted to get the original secret message. System design of this proposed framework is shown in Fig 1.

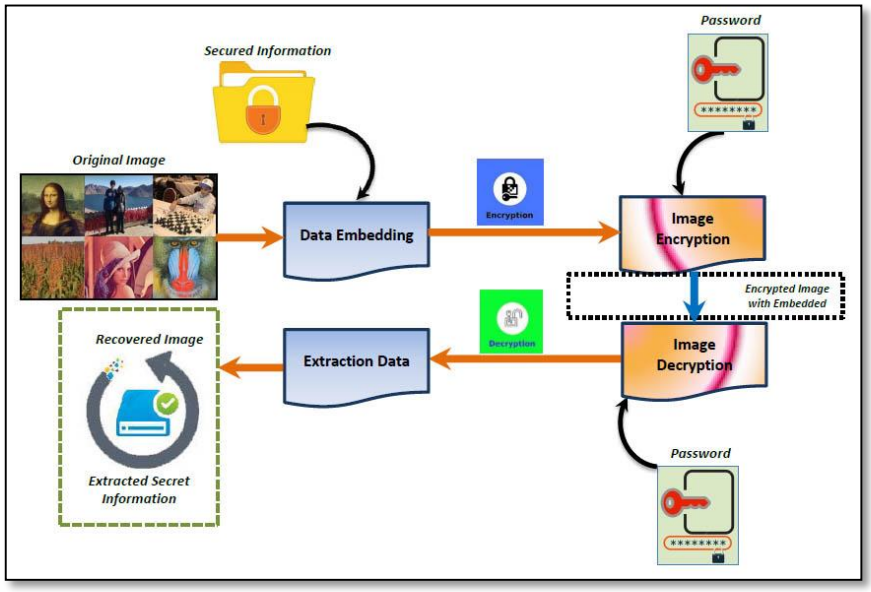

FIG. 1
INFRASTRUCTURE DESIGN OF THE SYSTEM

The secret message M (ScrMsg)will be arranged to create a simulated k-bit image for the extraction process to integrate the n-bit ScrMsg M to the far right of the LSB of cover image C, provided STImage S, embedded messages can be retrieved directly.

The following steps are an algorithm for embedding a TxTMsg
Stage 1: Both the TxTMsg to be hidden into COImage and COImage are reading.
Stage 2: Calculate the length of the TxTMsg.
Stage 3: Convert the length in binary and embed it into the COImage.
Stage 4: Convert TxTMsg in binary into 8-bits.
Stage 5: Following the length bits, start embedding the message bits.
Stage 6: Replace LSB of COImage with each bit of ScrMsg one by one.
Stage 7: Get STImage

The following steps are TxTMsg retrieving algorithm:-

Stage 1: Read the STImage.
Stage 2: Retrieve the length of the message.
Stage 3: For each pixels of STImage, Compute LSB.
Stage 4: Bits are retrieved and each 8 bit is converted into a character.
Stage 5: Get the ScrMsg.

PROCESSING STAGE

1. Embedding a Message

Describes how the meaning is embedded in an image. The image is embedded using LSB technology in this case. Fill in the blanks with the significant text you want to appear in the image. Paste text from the clipboard or type the text to be used in the Add Message dialog box. Finally, the text will be included in the image in this module, and a STImage will be generated. STImage is created by hiding secret text in an image. Use this STImage as the next module's input.

2. Embedding Image into an Image

It describes about embedding the output of above module with an image. In order to hide an image inside another image, user needs to click Open source image button. To embed, user needs to give password of length 8 characters which is also hided in that image. The path of the particular file will be displayed in a text field. You have to give the path for storing encoded image. For embedding, user should click on Make Stegano-Image.

3. Retrieving the Image File

It is about decoding the STImage which contains the original message. To open the image with secret data, click open button and go to the same path where the encoded image is stored. A text box will show the complete path about the embedded image. Receiver should enter the same password given by the sender for decoding the STImage. On clicking the decoding button, decoding starts our STImage and COImage will be separated. The retrieved COImage will be saved in the user preferred directory.

4. Retrieving the Secret Data

The method of extracting the ScrMsg from an image extracted from the previous module. In order to extract the ScrMsg from the STImage, LSB replacement algorithm was utilized. This STImage contains the original secret message. Click Open button and follow the same path where decoded STImage is stored. Complete path will displayed in text box which is in front of Destination File button. Click on Decode button. The message will be retrieved and the retrieved message will be displayed in the Text Area. Thus, as a result the ScrMsg is successfully encrypted and decrypted using multilevel encryption. Security is enhanced using this multiple encryption techniques and data is hidden inside the image using LSB replacement algorithm.

FINDINGS ANALYZING

This result analysis represents the experimental results for different image formats. Different images are taken for embedding and the performance of the proposed system is analyzed.

FIG. 2
COIMAGE

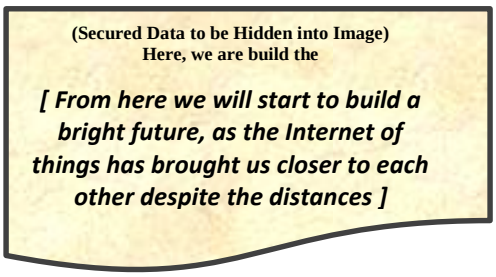

FIG. 3

THE MESSAGES REQUIRED HIDING

FIG. 4
IMAGE TO HIDE DATA

The result of embedding ScrMsg in COImage. This result is obtain by taking same COImage of different format For this we have calculated PSNR, MSE. After that, the PSNR and MSE are determined as follows:

$$PSNR = 10log_{10}\frac{Max^2}{MSE} \quad (1)$$

$$MSE = \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(\alpha_{ij} - \beta_{ij}) \quad (2)$$

Here, α_{ij} is the COImage pixel where the coordinates (i, j), and β_{ij} is the STImage pixel where the coordinate is (i, j). M and N represents image size. The various between COImage and the STImage it is indicated by the larger PSNR value, and its more invisible to the human eye.

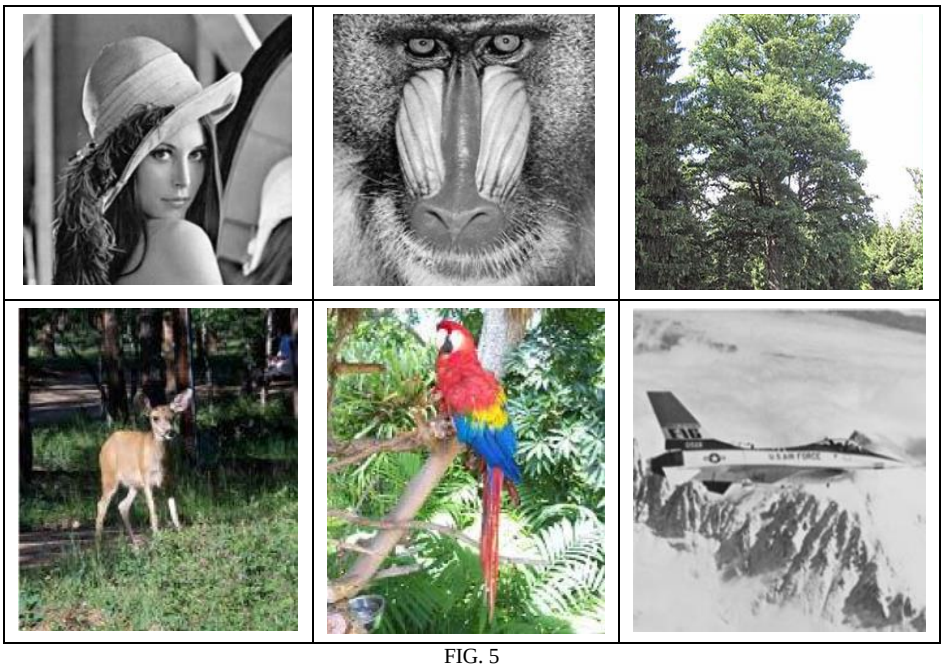

FIG. 5
THE IMAGES FOR TESTING

TABLE 1
COMPARISON OF DIFFERENT TYPES OF IMAGES

Testing Images	**PSNR (1)**	**PSNR (2)-LSB**	**MSE (1) Dynamic Histogram Shifting**	**MSE (2) Least Significant Bit**
Deer	47	53	0.040	0.0401
Fighter Jet	41	50	0.0418	0.0217
Parrot	45	53	0.3112	0.0433
Lena	42	51	0.0371	0.0221

Different Test Images have been taken to test the efficiency of our proposed system. If PSNR value of an output image is high then that image is having low distortion. MSE value is a reciprocal of PSNR. If MSE value of an image is low then that image is having low distortion. By comparing the PSNR and MSE values of our Least Significant Bit and Difference Histogram Shifting technique, proposed technique is very efficient than the DES technique.

CONCLUSION

The suggested framework is modular and follows a clear flow for quick comprehension. This was done to ensure that further modifications will be quickly implemented without requiring significant improvements to the program. When additional modules are needed, they can be easily installed. The platform is designed in a modular fashion. Many of the components were checked individually before being combined to form the full structure. The device was then checked for a variety of image formats and found to be functional.

A multilevel encryption was used to improve confidentiality, and the LSB technique was used to EmData into the picture in this proposed system. In the future, additional modules will be added to increase the embedding capability even further.

REFERENCES

[1] Saxena, A.K., Sinha, S., & Shukla, P, "Design and development of image security technique by using cryptography and steganography: a combine approach," *International Journal of Image, Graphics and Signal Processing*, Vol. 11, No. 4, 2018, 13-21.

[2] Cheddad, A., Condell, J., Curran, K., & Mc Kevitt, P, "Digital image steganography: Survey and analysis of current methods," *Signal processing*, Vol. 90, No. 3, 2010, pp. 727-752.

[3] Roy, C.Y., & Goel, M.K. (2016). Review on image steganography. *Indian Journal of Science and Technology*, Vol. 9, No. 47, pp. 1-5.

[4] Chauhan, S., Kumar, J., & Doegar, A, "Multiple layer text security using variable block size cryptography and image steganography," *In 3rd International Conference on Computational Intelligence & Communication Technology (CICT)*, 2017, 1-7.

[5] Nilizadeh, A., & Nilchi, A.R.N, "A novel steganography method based on matrix pattern and LSB algorithms in RGB images," *In 1st Conference on Swarm Intelligence and Evolutionary Computation (CSIEC)*, 2016, 154-159.

[6] Rahmani, P., & Dastghaibyfard, G. (2018). An efficient histogram-based index mapping mechanism for reversible data hiding in VQ-compressed images. *Information Sciences*, Vol. 435, pp. 224-239.

[7] Pradhan, A., Sahu, A.K., Swain, G., & Sekhar, K.R, "Performance evaluation parameters of image steganography techniques," *In International Conference on Research Advances in Integrated Navigation Systems (RAINS)*, 2016, 1-8.

[8] Hussain, M., Wahab, A.W.A., Idris, Y.I.B., Ho, A.T., & Jung, K.H, "Image steganography in spatial domain: A survey," *Signal Processing: Image Communication*, Vol. 65, 2018, pp. 46-66.

[9] Chan, C.K., & Cheng, L.M, "Hiding data in images by simple LSB substitution," *Pattern recognition*, Vol. 37, No. 3, 2004, pp. 469-474.

[10] Ahmadi, S.D., & Sajedi, H, "Image steganography with artificial immune system," *In Artificial Intelligence and Robotics (IRANOPEN)*, 2017, pp. 45-50.

[11] Kasana, G., Singh, K., & Bhatia, S.S, "Data hiding using lifting scheme and genetic algorithm," *International Journal of Information and Computer Security*, Vol. 9, No. 4, 2017, pp. 271-287.

[12] Hussain, M., Abdul Wahab, A.W., Javed, N., & Jung, K.H, "Hybrid data hiding scheme using right-most digit replacement and adaptive least significant bit for digital images," *Symmetry*, Vol. 8, No. 6, 2016.

[13] Al-Janabi, S., & Al-Shourbaji, I, "A hybrid image steganography method based on genetic algorithm," *In 7th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)*, 2016, pp. 398-404.

[14] Khanam, F.T.Z., & Kim, S. (2017). Enhanced joint and separable reversible data hiding in encrypted images with high payload. *Symmetry*, Vol. 9, No. 4, pp. 50.