ENHANCED THE SECURITY OF LSB BASED TEXT AND IMAGE STEGANOGRAPHY USING AES TECHNIQUE

ABSTRACT:

Steganography is the technique of hiding secret data within an ordinary, non-secret, file or message in order to avoid detection; the secret data is then extracted at its destination. Image steganography is a technique used to hide secret data within an image and it is combined with encryption as an extra step for hiding or protecting data. The main objective of this proposed work is to hide the secret message inside the image using Least Significant Bit (LSB) technique and provide security for the hidden message using Advanced Encryption Standard (AES) Algorithm. Efficiency of algorithm is estimated by Peak Signal to Noise Ratio (PSNR), Mean Square Error (MSE) where higher PSNR value gives the high-quality image. Additionally the for transferring the key the Diffie Hellman's key exchange technique is used to share the key to the receiver securely and the RGB values of each pixel in an image is extracted and saving those values into a CSV (Comma-Separated Values) file .

Keywords- AES, Cryptography, Diffie-Hellman key exchange, Image steganography, LSB, MSE, PSNR Ratio

I. INTRODUCTION

Steganography becomes of greater significance in the digital era as more people are joining cyber space revolution. Computer network requires special means of security as the number of data being exchanged on the internet is increasing. Therefore, confidentiality and data integrity plays a major role to protect against unauthorized access. Information hiding is an emerging research area in modern communication. This includes applications such as watermarking, fingerprinting, copyright protection and steganography. In ancient time, the use of invisible ink or shaving the messengers head and then sending the person one's the hair grows or wooden wax was used as a medium in order to pass the secret communication from sender to receiver. The problem with this way of communication was the information was not secured. With the advancement in the technology, internet came into existence. As a result all the information transfer took place over the web. In the modern era, the word steganography means to hide the secret data in a file so that the third person is unaware of the existing hidden data into the image. The hidden secret information can be in two insertion domains: spatial domain and frequency domain. Our main focus is in the spatial domain because the changes in the cover image are indistinguishable by the human eye. The spatial domain performs the dissimilation in the bits of the pixel of the original image. LSB (Least Significant Bit) technique is one of the spatial domain techniques. In LSB each of the bit of the data i.e. the character or the image are placed in the least significant bit of the cover image so that the distortions brought by the insertion process remain imperceptible by the human eye. In our work we study about the LSB technique (LSB) i.e. embedding the secret data into the cover image and in order to protect and provide security for the stegno-image AES (Advanced Encryption Standard) algorithm is used. We take different images of various formats and try to hide the secret data of varied length into the cover image. The paper is organized as following: section II presents LSB substitution and AES algorithms. Section III presents proposed works LSB with AES algorithm. Section IV presents experimental results. Section V presents general conclusion.

II. METHODS

A. Least Significant Bit

Least Significant Bit is one of the spatial domain techniques where each bit in the text is

substituted from the least significant bit of the original image. It is simple and easy to implement. The specialty of its existence in spatial domain is because the human eye cannot distinguish between the original and encrypted image. LSB can be extended up to 4-bits or 2-bits out of 8-bits, but it may cause distortion in an image due to change in the intensity of an image. LSB substitution comprises of

1. LSB ENCODER
2. LSB DECODER

LSB technique has become the basis of many techniques that hide the secret data within the carrier data. First section explains about the encoding process where the secret data is hided and next section explains about the decoding process where the data gets extracted.

LSB Encoder

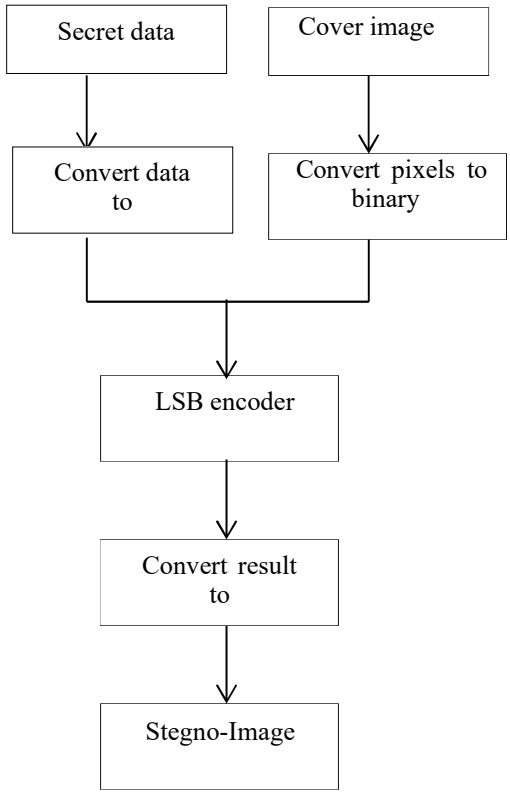

Figure 1: LSB encoder

1. Input the secret data i.e. text or the image which needs to be hidden.
2. Input the cover (original) image of size [256 256]
3. Convert the secret data i.e. asci value of each text or pixel value in case of image into binary representation.
4. Convert the pixel value of cover image into binary representation.
5. Apply LSB encoder; function is to hide each bit of text or image into the least significant bit of each 8 pixel value of cover image.
6. The resultant output is converted back to pixel values to get the stego-image.

LSB Decoder

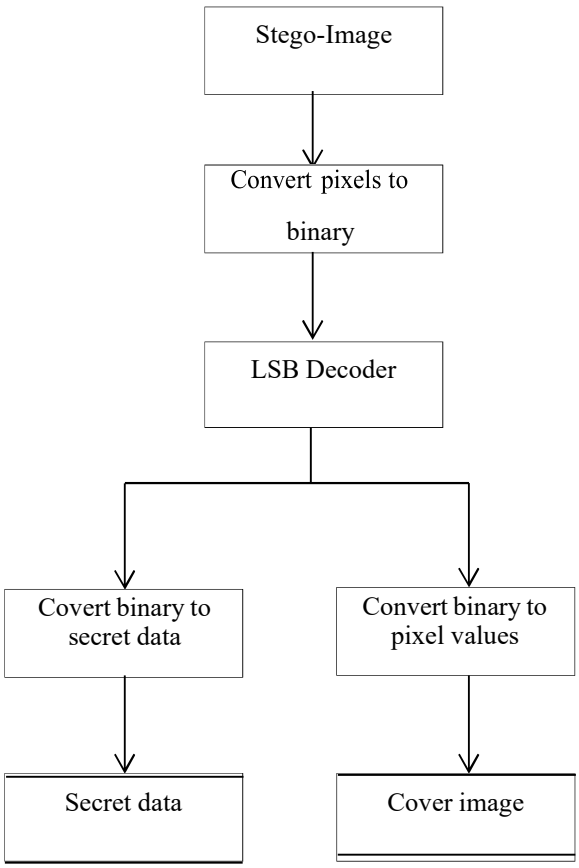

Figure 2: LSB decoder

1. Input the stego image.

2. Convert the pixel value to binary representation.

3. Apply LSB decoder; function is to retrieve the secret data back from the stego-image.

4. The secret data and the cover image are separated together to get the desired output

B. AES (Advanced Encryption Standard)

The AES algorithm is the symmetric algorithm that is secure enough to provide security for confidential data operating on plaintext of 128-bit with variable key length of 128, 192 and 256-bit . The number of rounds performed is 10, 12 and 14 respectively. AES is one of the strongest algorithms until now and we can use only one key at the sender and receiver side, hence the privacy made by the key is secured.

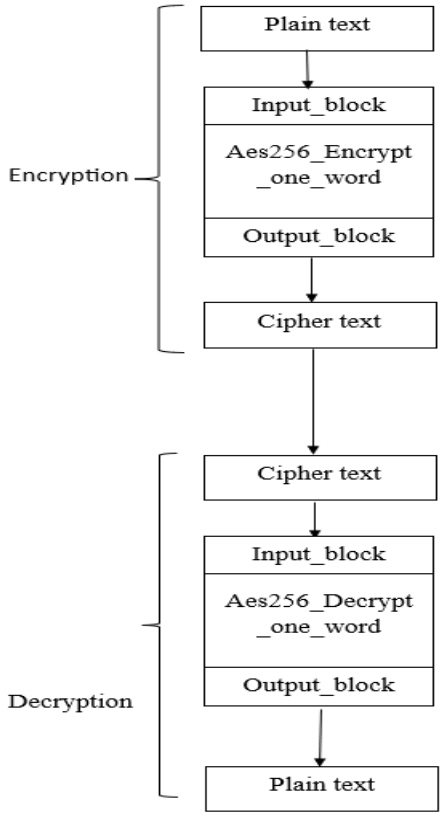

Figure 3: AES Encryption Algorithm

AES ENCRYPTION

All operations in AES are byte oriented. 128 bit plaintext is arranged in 4x4 matrixes with 16bytes. Substitute byte, shift rows, mix columns and add round key are the main steps in AES encryption .

a) Sub Bytes Transformation Sub byte is substitute byte where each byte from the cipher text is substituted from the standard look up table of s-box which contains 256 elements. S-box is shown in fig. First hexadecimal values in byte indicate row index and second
b) hexadecimal value indicate column index.
c) Shift Rows Transformation In shift row transformation, the rows of the matrix are circularly left shifted. Row 0is kept constant; Row 1 is left shifted by 1 byte; Row 2 is shifted by 2 byte and finally last row by 3 byte.
d) Mix Column Transformation Each column in the new formed matrix is multiplied with each column of the predefined matrix.
e) Add Round Key The resultant matrix is xor-ed with the expanded key generated from the initial key.

AES DECRYPTION

Decryption is just the opposite of encryption i.e. the cipher text is converted back to the plain text. Inverse substitute byte, inverse mix column, inverse shift rows and inverse s-box takes place.

a) Inverse Sub Byte Transformation Each byte from the matrix is substituted with the inverse s-box table to get the new matrix.
b) Inverse Shift Rows Transformation The row of the matrix is circularly right shifted.
c) Inverse Shift Rows Transformation The row of the matrix is circularly right shifted.
d) Inverse Add Round Key Transformation The round keys should

be selected in a reverse order and Xored with the state matrix.

III. LSB combined with AES Algorithm

Our approach is to provide security for the secret information hidden inside the cover image performed through LSB technique with the help of AES algorithm. The first step is to read the secret data and the cover image. The next step is to hide the data into the cover image which is done with the help of LSB encoder. To protect the data from the intruder the resultant stego-image is given to AES encryption where each of the pixels get scrambled .To get back the stego-image AES decryption is done and the output of this is provided to LSB decoder where the hidden data is retrieved finally.

General Block Diagram:

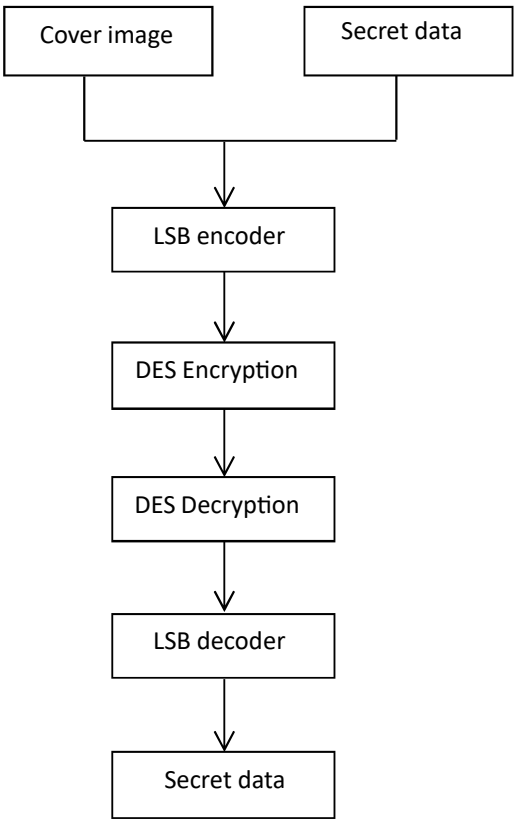

Figure 4: LSB combined with AES algorithm.

1. Read the input cover image of size [256 256].
2. Read the secret data i.e. text or the image.
3. Apply LSB encoder where the bits of the secret data are hidden into the least significant bit of the pixel value of the cover image.
4. The resultant stego-image is given to AES encryption in order to provide security for the hidden data.
5. The function of the AES decryption is to get back the stego image in order to retrieve back the data.
6. The LSB decoder finds each of the data which was embedded in the cover image and the resultant is the secret data.

DIFFIE-HELLMAN KEY EXCHANGE TECHNIQUE:

Diffie–Hellman key exchange (DHKE) is a method of securely exchanging cryptographic keys over a public channel and was one of the first public-key protocols as conceived by Ralph Merkle and named after Whitfield Diffie and Martin Hellman. DH is one of the earliest practical examples of public key exchange implemented within the field of cryptography. Published in 1976 by Diffie and Hellman, this is the earliest publicly known work that proposed the idea of a private key and a corresponding public key.

Working:

- **Setup:** Both parties, usually referred to as Alice and Bob, agree on a public domain parameters. These parameters include a large prime number (p) and a generator (g) that is also a primitive root modulo p. These parameters are known to both parties.
- **Key Generation:** a. Alice selects a random secret number, a, and computes A = g^a mod p. She then sends A to Bob. b. Bob selects a random secret number, b, and computes B = g^b mod p. He then sends B to Alice.

- **Key Exchange:** a. Alice receives B from Bob and computes the shared secret key as K = B^a mod p. b. Bob receives A from Alice and computes the shared secret key as K = A^b mod p.
- **Key Derivation**: Both Alice and Bob now have the same shared secret key, K, which can be used for symmetric encryption or any other cryptographic purposes.

RGB PIXEL VALUES:

The RGB value of a pixel represents the intensity of the red, green, and blue color channels that make up the pixel's color. Each color channel is typically represented by an 8-bit value ranging from 0 to 255, where 0 represents no intensity (no color) and 255 represents full intensity (maximum color). In an RGB image, each pixel is composed of three values: the red (R), green (G), and blue (B) color channels. These values define the contribution of each color channel to the overall color of the pixel. For example, a pixel with an RGB value of (255, 0, 0) represents full intensity red, as the red channel is at its maximum value (255).

Figure 5: Pixel with RGB value of (255, 0, 0)

while the green and blue channels have no intensity (both are 0). Similarly, a pixel with an RGB value of (0, 255, 0) represents full intensity green,

Figure 6: Pixel with RGB value of (0,255,0)

and (0, 0, 255) represents full intensity blue.

Figure 7: Pixel with RGB value of (0,0,255)

IV. EXPERIMENTAL RESULTS

In this work, we used LSB technique in order to hide the secret data i.e. text or image using LSB technique. Further in order to provide and protect the data AES algorithm is applied with the maximum key length of 128 bit. Image steganography is implemented in Visual Studio Code software.

Images of various sizes are used and further resized the image into 256 X 256 pixel values.

Using the above two algorithms, we have hidden the text inside the cover image and is also successful in retrieving back the hidden data, and the RGB values of each pixel in an image is extracted successfully and it is saved for the later verification. We compare the various images with the same text length and also with varied text length and also same in case of secret image.

256-bit length

Image(256x256)	PSNR RATIO(dB)	MSE
Bird	115.592967	27.50
Dog	40.3428344	32.07
Butterfly	47.7234370	31.34
Moon	89.6499633	28.60
Sky	1.67201232	45.89

Table 1: Text inside Image for 256-bit text length.

Table 1 gives the hiding of text inside the original cover image of text length of 256 bit long.

The figure 8 is the screenshot is the user interface of the application.

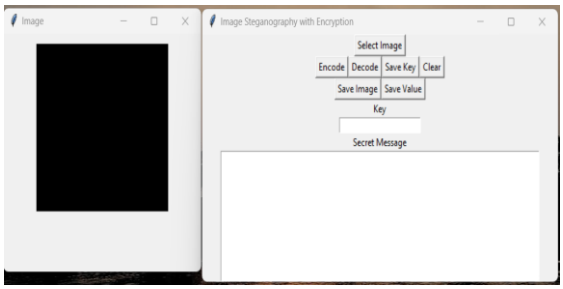

Figure 8

The figure 9 is the screenshot of the screen after the message is encoded successfully.

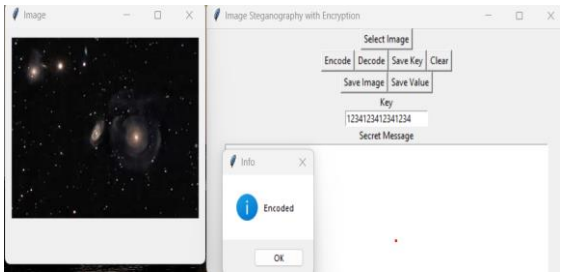

Figure 9

The figure 10 is the screenshot of the screen after decoding which shows the secret message in the message textbox.

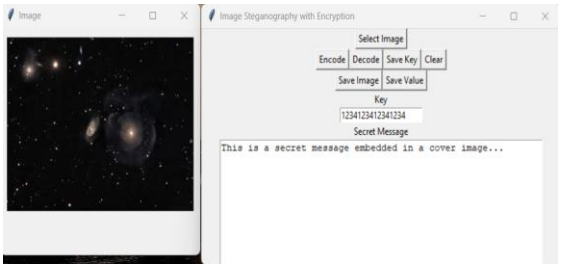

Figure 10

V. CONCLUSION

In this proposed work, a secret data or message is embedded in an cover image with the help of LSB steganography technique and AES cryptography algorithm, which is an NIST standard symmetric algorithm with key length of 16 character that is 256 bit to provide more security for the secret data or information. Based on our implementation of this algorithm using Python in Visual Studio Code environment the secret data embedded in an cover image is successfully retrieved using an aes encryption key and the RGB pixel value of the cover image and the secret image is successfully extracted and saved.

For further one additional encryption algorithm like RSA can be added for additional security.

References:

1. Shahid Rahman, Jamal Uddin, Habib Ullah Khan, Hameed Hussain, Ayaz Ali Khan, and Muhhamad Zakarya “A Novel Steganography Technique for Digital Images Using the Least Significant Bit Substitution Method”, 2022.

2. Chiradeep Gupta and N V Subba Reddy “Enhancement of Security of Diffie-Hellman Key Exchange Protocol using RSA Cryptography”, 2022.

3. U. A. Md. Ehsan Ali, Emran Ali, Md. Sohrawordi and Md. Nahid Sultan “A LSB Based Image Steganography Using Random

Pixel and Bit Selection for High Payload", 2021.

4. Lalit Kumar Gupta, Aniket Singh, Abhishek Kushwaha, and Ashish Vishwakarma, "Analysis of Image Steganography Techniques for Different Image Format", 2021.

5. Arshiya S. Ansari, Mohammad S.Mohammadi, and Mohammad Tanvir Parvez" A Multiple-Format Steganography Algorithm for Color Images,"2020.

6. Jemima Dias, Dr. Ajit Danti "Image Steganography based Cryptography",2020 International Journal of Scientific & Engineering Research.

7. Ali Ahmed and Abdelmotalib Ahmed "A Secure Image Steganography using LSB and Double XOR Operations" IJCSNS International Journal of Computer Science and Network Security, VOL.20 No.5, May 2020.

8. Arun Kumar Singh, "Steganography in Digital Images Using LSB Technique", Journal of Xidian University 2020.

9. Mr. Jawwad A R. Kazi, Mr. Gunjan N. Kiratkar, Ms. Sonali S. Ghogale, Prof. Atiya R. Kazi "A novel approach to Steganography using pixel-based algorithm in image hiding", 2020 International Conference on Computer Communication and Informatics,2020.

10. Sakshi Audhi, Maruska Mascarenhas "Secure Mechanism for Communication Using Image Steganography" 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT).

11. NIST, "Advanced Encryption Standard," http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf, 2001 (accessed March 24, 2017).

12. . Z. Y. Al-Omari and A. T. Al-Taani, "Secure LSB steganography for coloured images using character-colour mapping," 2017 8th International Conference on Information and Communication Systems (ICICS), 2017, pp. 104-110.

13. Beenish Siddiqui, Sudhir Goswami "A Survey on Image Steganography Using LSB Substitution Technique" 2017 International Research Journal of Engineering and Technology (IRJET).

14. Aman Arora, Manish Pratap Singh, Prateek Thakral, Naveen Jarwal proposed a paper by name "Image steganography using Enhanced LSB substitution technique",2016 fourth international conference on Parallel, Distributed and Grid Computing.

15. Arun Kumar Singh, Juhi Singh, Dr. Harsh Vikram Singh "Steganography in Images Using LSB Technique", 2015 International Journal of Latest Trends in Engineering and Technology (IJLTET).