Data Hiding and Data Protection of Patients’ records in a Multi-Specialty Hospital

Gokul Gopakumar
Department of Computer Science and Engineering
Vellore Institute of Technology, Vellore
Vellore, Tamil Nadu, India
gokul.ggs2012@gmail.com

Kaipu Surya Prathap Reddy
Department of Computer Science and Engineering
Vellore Institute of Technology, Vellore
Vellore, Tamil Nadu, India
suryakaipu98@gmail.com

Anuj Suresh
Department of Computer Science and Engineering
Vellore Institute of Technology, Vellore
Vellore, Tamil Nadu, India
anuj.suresh2019@vitstudent.ac.in

Edara Hementh Kumar
Department of Computer Science and Engineering
Vellore Institute of Technology, Vellore
Vellore, Tamil Nadu, India
hemanthkumar.edara2019@vitstudent.ac.in

Surya Narayan Sahu
Department of Computer Science and Engineering
Vellore Institute of Technology, Vellore
Vellore, Tamil Nadu, India
surya.official49@gmail.com

***Abstract*—In the last 20 years, the healthcare industry has seen a significant increase in the use of medical information and its applications. In order to deliver top-notch healthcare services on a worldwide scale, the ability to access patient information utilizing web-based technologies from any place is becoming increasingly important. Medical data communicated over open networks must be protected in the current era, especially when important considerations like patient confidentiality are taken into account. Particularly sensitive, patient medical information requires the highest level of security during both storage and transfer. Linking these records to patient medical information, such as X-ray or scan images, is frequently necessary (CAT, MRI, etc.). We plan to employ LSB-based image steganography for data concealment and cryptography (the AES approach) for data encryption, even though there are a number of security solutions that can encrypt the information and prevent unauthorized access to the data.**

Keywords—healthcare, confidentiality, data concealing, steganography, encryption

I. Introduction

A technology that makes this feasible is the transfer of electronic patient records (EPR) through the Internet to hospitals and foreign places over great distances. The transmission of secure EPR is required in order to reduce the danger of a network security breach and stop data access by unauthorized end users, but, since EPR is a very confidential medical record.

Health care workers have a moral and legal obligation to safeguard patient information, which may contain some of the most intimate data about an individual. Electronic medical records (EMR) pose a larger risk of information leakage to the public than other formats, such paper-based data. Working with medical imaging raises three security issues: confidentiality, reliability, and availability.

Most clinical picture archiving and communication systems employ the digital imaging and communications in medicine (DICOM) standard image format (PACS). For PACS and DICOM, there are now a number of security measures that provide adequate protection for data storage and transfer. These features include encrypted transmission, firewalls, passwords, private keys, and public keys.

II. Literature Reviews

1. Using Image Steganography for Providing Enhanced Medical Data security

This study makes a case for the novel image steganography approach for concealing medical data. The suggested picture steganography technique illustrates how sensitive medical or patient information can be secured and encrypted while maintaining the quality and imperceptibility of the stego images. Multiple layers of encryption are provided for the medical data using swapped Huffman tree encoding. By comparing the histograms of the stego image with the cover image, it can be shown that the hidden data is still undetectable.

2. An improved k-nearest neighbor algorithm and its application to high resolution remote sensing image classification

The K-nearest neighbor (KNN) classification approach is widely used in data mining techniques. It is widely utilized in a variety of domains due to its ease of implementation, clarity of theory, and outstanding classification performance. When training samples are distributed unevenly or the sample size of each class is highly variable.

3. An exhaustive survey on security and privacy issues in Healthcare 4.0

This study includes a comprehensive overview of the literature and an analysis of cutting-edge suggestions to uphold security and privacy in Healthcare 4.0. In order to inform the research and application communities, it also examined the blockchain-based solution. In an organized approach, multiple taxonomies for examining various security and privacy challenges in Healthcare 4.0 are also provided. Next, the study explores and discusses the benefits and drawbacks of various security and privacy solutions.

4. Data Hiding Scheme for Medical Images

This research presents a novel method for blind watermarking a medical image that is resistant to attacks like brightening or contrast enhancing. After geometrical attacks like scaling and translation, we can successfully recover data by utilizing image moments. The outcomes demonstrate that using homogeneity allows for greater extraction accuracy than using the straightforward LSB method, which only selects regions close to borders.

III. Methodology

We must keep the secret image and text in the database in a way that no one else can discover them in order to achieve secrecy. In this method, a cover picture would be utilized as a ruse in which the hidden image and hidden message would be concealed. The AES Algorithm, which makes up the cryptographic component of the process, would be used on the sender's end to encrypt the secret picture. LSB Based Image Steganography would be used to cover the secret text message in this encrypted secret image. Additionally, LSB-Based Picture Steganography would be used to conceal the cover image while preserving access to the encrypted secret image and embedded secret text.

The steganographic or data-hiding portion of the process would consist of these two phases. After completing the aforementioned procedures, the steno-image is obtained. It is then divided into 16 pieces, indexed, and conveyed to the recipient via image segmentation.

These sub-images would be retrieved one at a time during decryption and combined depending on their indices. The LSBs of the combined cover picture would then be used to retrieve the encrypted image. The secret text would then be extracted from the LSBs of this encrypted picture.

The decryption technique would also be used to separate the encrypted secret picture from its original form. The hidden picture and the secret message will thus be revealed to the recipient through the cover image.

The decryption technique would also be used to separate the encrypted secret picture from its original form. The hidden picture and the secret message will thus be revealed to the recipient through the cover image.

After the entire process is complete, we would check to see if the receiver had indeed received the right secret image and secret text by comparing the newly acquired information to the information that was originally given by the sender.

*A. **Analysis and Design***

The solution proposed in this project divides the whole workflow into four components. The secret image is initially divided into several pieces, and the next several steps are then taken:

- The encryption/decryption phase (AES encryption on the secret picture): During this phase, the secret image is encrypted using the AES technique. Even if the attacker manages to obtain this encrypted image via steganalysis, the sender uses the AES technique to render the secret image unintelligible. The receiver will next use the AES algorithm's decryption process to recover the original secret picture from its encrypted state.

- The text-based steganography embedding phase – Using an LSB-based picture steganography technique, the secret text message is concealed inside the encrypted secret image (obtained from the preceding stage) (text-based steganography). Important text data that we wish to convey securely is included in ciphertext (e.g., CVV of a card or a PIN). In this stage, we incorporate our encrypted secret image with our secret text. The secret text information in this image must first be extracted from the encrypted image by the recipient when he receives it.

- The hiding phase (KMCG method for image steganography) - Kekre's Median Codebook Algorithm (KMCG algorithm) is a codebook generating technique used for picture steganography. Since utilizing traditional LSB-based steganography to conceal the complete encrypted picture in the cover image is not possible, we will use the KMCG technique to conceal the hidden image behind the cover image. The receiver's end receives the generated picture. The encrypted picture will then need to be extracted from the cover image by the recipient using the same KMCG process, which will require turning the complete image into a codebook.

- The K-nearest neighbor supervised technique may be employed during the segmentation step, which involves splitting the picture and piecing it back together. All subimages have SIFT features derived from them. Each feature's KNN is discovered using a k-d tree. Geometrically consistent feature matches are found using RANSAC. The latter related elements of the picture matching are discovered.

These make it challenging for an intruder to decode all of the picture components because they are out of sequence, encrypted, and concealed inside of a decoy image.

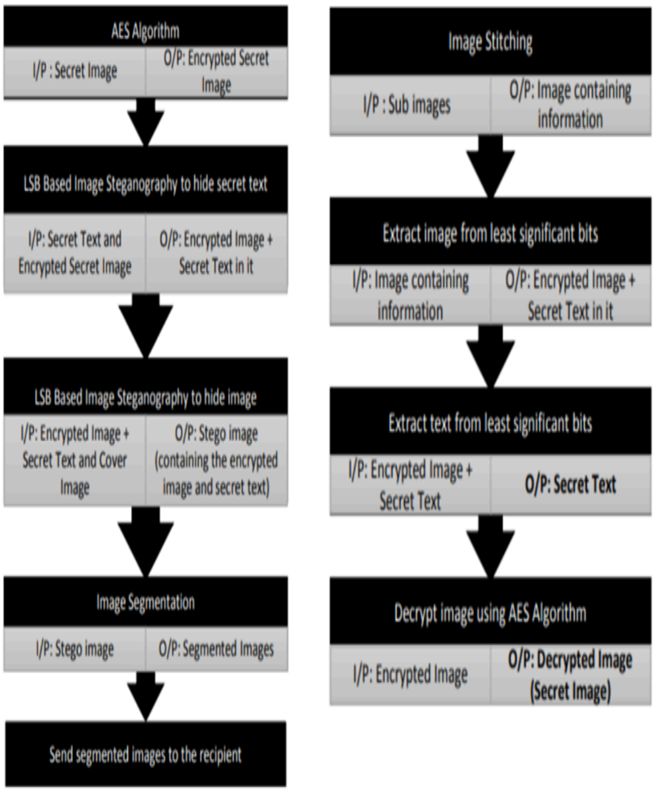

AES Algorithm
I/P : Secret Image
O/P: Encrypted Secret Image
LSB Based Image Steganography to hide secret text
I/P: Secret Text and Encrypted Secret Image
O/P: Encrypted Image + Secret Text in it
LSB Based Image Steganography to hide image
I/P: Encrypted Image + Secret Text and Cover Image
O/P: Stego image (containing the encrypted image and secret text)
Image Segmentation
I/P: Stego image
O/P: Segmented Images
Send segmented images to the recipient
Image Stitching
I/P : Sub images
O/P: Image containing information
Extract image from least significant bits
I/P: Image containing information
O/P: Encrypted Image + Secret Text in it
Extract text from least significant bits
I/P: Encrypted Image + Secret Text
O/P: Secret Text
Decrypt image using AES Algorithm
I/P: Encrypted Image
O/P: Decrypted Image (Secret Image)

Fig 1 : Flowchart of the components

B. *Architecture*

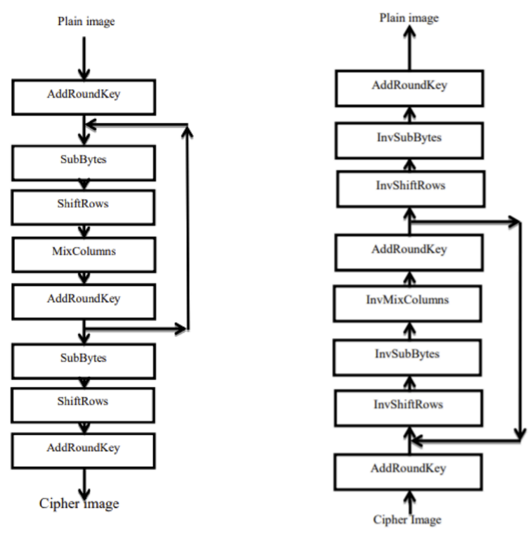


Fig 2 : Encryption and Decryption phase
(Sender and Receiver Side)

• Senders side

Phase of encryption: The AES method was used to encrypt and decode images. First, a known size was achieved by resizing the provided hidden picture (in our case 480 x 480). Following that, the picture was split into Nx16 chunks. Both the transmitter and the receiver employ a 128-bit key. This key has to be securely exchanged between the sender and the receiver. The key can be seen as a series of 8-bit blocks numbered x [0], x [1],...x [15]. We process 16 pixels in each cycle of the AES encryption algorithm. The AES algorithm employs a round function made up of the following four byte-related transformations:

1. **SubBytes step:** Using an 8-bit substitution box, each value in the state matrix (original picture) is changed to a subbyte (S-box). This guarantees the cipher's non-linearity. The inverse function and an invertible affine transformation are used to create the S-box. Each byte in the state S is replaced with an 8-bit lookup table using the formula b I j] = S. (a[i, j]).

2. **Shift Rows step:** The bytes in each row are moved to the left by an incrementally larger offset. The top row stays the same. The second row's bytes are cyclically shifted one byte to the left. The third and fourth rows are also relocated by corresponding offsets of two and three.

3. **Combine Columns:** The four blocks of each state's column are combined using an invertible linear transformation, such as multiplication and bitwise XOR. This method emits four bytes after receiving four bytes as input. An easy XOR operation represents addition. The polynomial multiplicand is modulo irreducible. A constant polynomial, c, is multiplied by each column in the state (x).

4. **Execute Round Key step:** A combined state and sub key is used. A sub key is discovered using the primary key. By XORing a byte from the state with the corresponding byte from the sub key, the sub key is added.

We obtain an encrypted version of the secret image we wished to share with another user after this encryption stage with the AES technique.

Phase of Embedding: The secret message is concealed in the secret encrypted cypher picture during the embedding phase, which follows the encryption of the key image. To point to the conclusion of the sentence, we append the null character to the key text. Since these bits hold the smallest amount of information in an image and flipping them won't significantly alter the picture's look, LSB-based image steganography includes concealing data in these bits. The key message is first transformed into its ASCII equivalent. The 8-bit binary equivalent of those ASCII values is then created. These bits are then sequentially substituted for the smallest significant bits of the pixel values of the encrypted cipher picture. As a result, the key image and secret text are successfully concealed.

The message that has got to be hidden is reflected in the final bit of each pixel value using LSB replacement steganography. Consider an 8-bit grayscale bitmap picture, where each pixel is represented by one byte that corresponds to a color value in the grayscale. Assume that the primary eight grayscale pixels in the original image have the following values:

0 1 0 1 0 0 1 0

0 1 0 0 1 0 1 0

1 0 0 1 0 1 1 1

1 1 0 0 1 1 0 0

1 1 0 1 0 1 0 1

0 1 0 1 0 1 1 1

0 0 1 0 0 1 1 0

0 1 0 0 0 0 1 1

We would change the LSBs of these pixels to have the following new values in order to conceal the letter Z, whose binary equivalent of the ASCII [90] code is 01011010.

0 1 0 1 0 0 1 0

0 1 0 0 1 0 1 1

1 0 0 1 0 1 1 0

1 1 0 0 1 1 0 1

1 1 0 1 0 1 0 1

0 1 0 1 0 1 1 0

0 0 1 0 0 1 1 1

0 1 0 0 0 0 1 0

In a similar way, we first divide the image's pixels into three planes (in our example, the RGB plane), then we replace each plane's LSB with the 8-bit binary value of each text character's ASCII code. This method likewise successfully achieves data concealment while maintaining the design of the image. After completing this step, we've a secret text-encrypted cipher picture, thus we proceed to the subsequent stages.

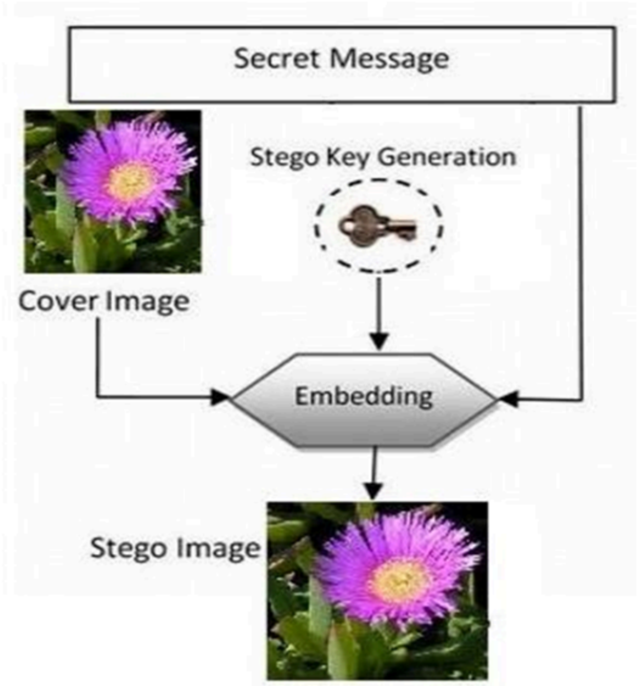


Fig 3 : Secret Text Embedding Procedure

Hiding Phase : The encrypted secret picture containing the secret text is concealed inside the cover image using the same procedure as LSB-based image steganography, which requires concealing the secret message within the secret image. The red component of the cover picture conceals the binary representation of the pixel values of the encrypted image with the secret text, which is then followed by the blue and green components.

The KMCG algorithm, also known as Kekre's Median Codebook Generation Algorithm, was used to apply steganography.

The attacker may use steganalysis to uncover the hidden information if the cover data is corrupted. The typical LSB method's embedding capacity is computed using:

(W * H)/8 = capacity

W * H * 3/8 = [for RGB picture] capacity

where W denotes the image's width, and H its height. Consider a grayscale picture with the dimensions W = 200 and H = 150. The embedding capacity is 3750 bits as a result. The embedding capacity would be 11250 bits if the picture were RGB.

The least significant bit (LSB) of each sampling point can be replaced with binary data in LSB coding using algorithms like the KMCG, which create a compressed codebook of the cover image and then allow the replacement of the bits to hide image data in it. This allows for the encoding of a large amount of data.

This is how the algorithm works:

a. Split the picture into 22 pixel blocks.

b. Create the first training set cluster using rows with a 12-value-per-pixel frame (4 pixels in each block and 3 RGB planes so total of 12 values per window).

c. Apply the KMCG codebook generation method to the original cluster to produce a codebook with 2048 code vectors.

d. Include a column for beginning index position in the codebook (CB).

e. Sort the CB data using a dictionary.

f. Only show the final column of data in the sorted CB.

g. Add the last index position column to the Stego CB once again .

h. Sort the Stego CB.

i. Reconstruct the Stego cover picture using the sorted CB data (as this image now contains the encrypted secret image with the embedded secret text; hidden in it).

j. Send the recipient the rebuilt picture and the updated final index position.

Segmentation Phase : Next, we separate the cover picture with the encrypted secret image and the hidden text into several portions (16 to be specific). The receiver is then provided each component separately.

• Receiver's side

Phase of stitching: The original cover picture containing the concealed secret encrypted image and the embedded secret text is recovered at the receiver's end by stitching back the sub images based on their index values (0 to 15).

Extraction Phase: Using the same KMCG technique as on the sender's end, the least significant bits of the red, blue, and green components are extracted, eight bits at a time, for the complete size of the secret image. The KMCG algorithm's stages are as follows when used at the receiver's end:

a. Receiver will create training vectors by dividing the received picture into blocks.

b. The codebook is still the only source for the collection of unique training vectors. Utilizing the final index position that was obtained, arrange the codebook entries.

c. Take the bits in the codebook and extract the secret information.

We now have access to the encrypted secret picture and the hidden text that it contained.

Phase of Retrieval: The least significant bits are extracted from the secret encrypted picture that we acquired through the aforementioned stages, eight bits at a time, until we reach a number that represents a zero (because we added a zero to the end of each text word at the sender's end). The resultant 8-bit binary integers are then translated to decimal form. The ASCII values of the characters are represented by these decimal representations. The secret message (such as a patient's records) that was contained in the encrypted secret picture is then extracted by converting them back to their character representation.

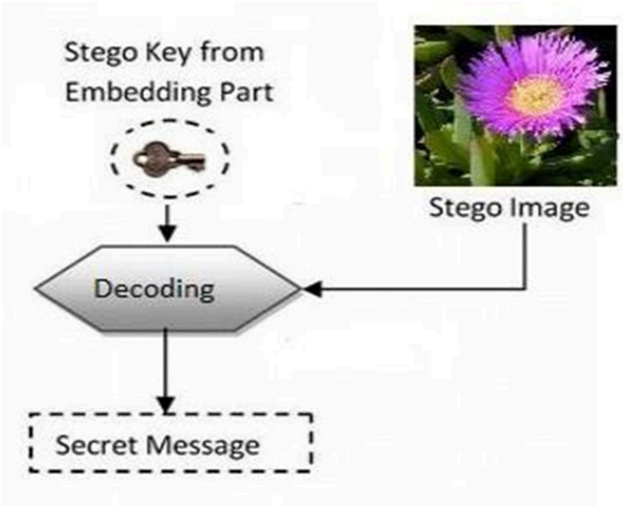


Fig 4: Secret Text Extraction Process

Phase of decryption: The stages of the AES algorithm are performed in the opposite sequence from the encryption process when decrypting a secret picture.

1. We undo the changes done to the picture on the sender's side in the Add Round Key step first.

2. Next, the polynomial function's inverse is applied to the picture in the Inverse Mix Columns step. By doing so, the columns of the picture are restored to their initial placement.

3. The image's rows that were mixed up on the sender's side before transmission are then reversed.

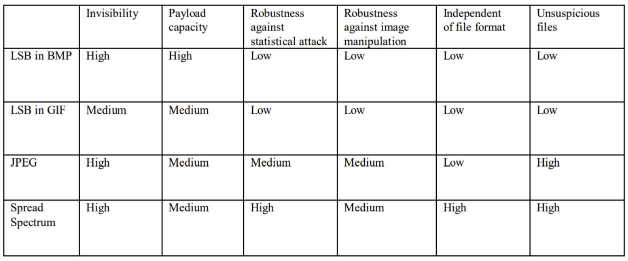
4. After that, inverse substitution is used by getting the 8-bit table lookup's inverse. We get the hidden picture on the receiver's end by repeating the process.

IV. Results

We have discovered a safe method of text and picture communication by combining AES encryption and LSB-based steganography. Because AES data encryption is a more mathematically sophisticated and beautiful cryptographic technique with a range of key length options, we chose to employ it. AES is significantly more secure than DES's 56-bit key since it offers 128-bit, 192-bit, or 256-bit key options. Despite being a symmetric method, no one has yet been able to decrypt it without a key.

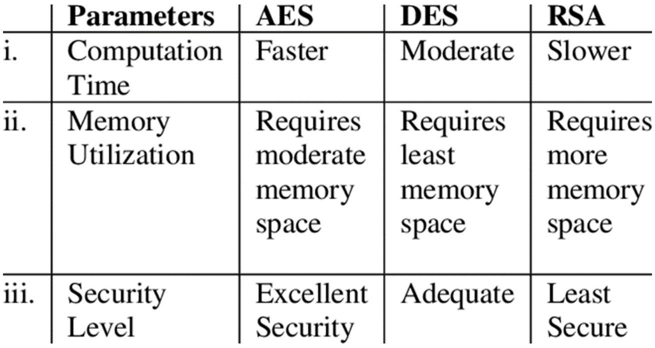
We also conducted a study on alternative algorithms. It was shown that LBG (Linde-Buzo-Gray) and KPE (Proportionate Error Algorithm) provide lower mean square errors (MSE) than KMCG (Kekre's Median Code), demonstrating that they are superior for steganographic applications. KPE and LBG are slower than KMCG, though. By employing KMCG, a significant amount of time is saved overall at the expense of a modest increase in message distortion.

We then further separated the final image into 16 pieces after using the KMCG and AES algorithms so that it could be transmitted to the receiver across various channels. Therefore, even if someone successfully intercepts a channel and has access to partial pictures, they won't be able to acquire the complete image and therefore won't be able to receive our secret image or secret text.



	Invisibility	Payload capacity	Robustness against statistical attack	Robustness against image manipulation	Independent of file format	Unsuspicious files
LSB in BMP	High	High	Low	Low	Low	Low
LSB in GIF	Medium	Medium	Low	Low	Low	Low
JPEG	High	Medium	Medium	Medium	Low	High
Spread Spectrum	High	Medium	High	Medium	High	High

Fig 5: Comparison of Image Steganography Techniques



	Parameters	AES	DES	RSA
i.	Computation Time	Faster	Moderate	Slower
ii.	Memory Utilization	Requires moderate memory space	Requires least memory space	Requires more memory space
iii.	Security Level	Excellent Security	Adequate	Least Secure

Table 1: Comparison of Encryption Techniques

Testing and Validation:

TEST CASE	
TEST CASE ID:TC_001	**Test Designed by:** **Surya Narayan Sahu**
Test Priority: High	Test Designed date: 18 August 2022
Module Name: Image Upload	Test Executed by : Surya Narayan Sahu
Test Title: Image Upload types	

SN No	Scenario	Result	Status
1	Upload Jpeg file	“Successfully Uploaded” is Shown	Pass
2	Upload BMP file	“Successfully Uploaded” is Shown	Pass
3	Upload JPG file	“Invalid Type”	Pass

TEST CASE	
TEST CASE ID:TC_002	**Test Designed by:** Kaipu Surya Prathap Reddy
Test Priority: High	Test Designed date: 18 August 2022
Module Name: LSB Encryption	Test Executed by : Edara Hementh Kumar
Test Title: LSB encryption	

SN No	Scenario	Result	Status
1	JPEG file is uploaded of size<5Mb	“Successfully Uploaded” is Shown	Pass
2	JPEG file is uploaded of size>5Mb	“File size exceeds 5MB” is Shown	Pass

TEST CASE	
TEST CASE ID:TC_003	**Test Designed by:** Gokul Gopakumar
Test Priority: High	Test Designed date: 18 August 2022
Module Name: Decryption	Test Executed by : Anuj Suresh
Test Title: Decryption	

SN No	Scenario	Result	Status
1	Decryption of a encoded image	We get the original plaintext	Pass
2	Decryption of Encoded image	The original image with information is shown	Pass

V. Acknowledgement

We would like to express our sincere gratitude to Dr. G. Viswanathan, the VIT University's cherished Chancellor for allowing us to complete the project.

We want to thank our mentor, Dr. Archana T , for her advice and ideas, which allowed us to finish the project on schedule. Words can't adequately explain how grateful we are to the professors and personnel that helped us with the project and gave us assistance.

VI. Conclusion

In this paper, we have derived a best optimal way to send images and some text secretly by using LSB-based steganography. By dividing the image into 16 parts, and sending them through different channels, we have made attackers' task even more troublesome. By sending images in this way, we can prevent unauthorized personnel from accessing our images or texts via intercepting attacks. And hiding in a cover image does not raise as many questions as weird-looking images would. If someone applied steganalysis, and retrieved the images, they would not be able to know the matrix operations we applied using AES encryption. And hence, they would be unable to get any kind of data even if they somehow manage to get access to the photos

VI. Future Work

Future work can be done to enable us to use various steganographic methods and offer even higher security levels. We can also proceed by utilizing asymmetric encryption techniques like Diffie-Hellman, ECC, El Gamal, etc. Asymmetric algorithms will increase security since they employ different keys to encrypt and decrypt messages. By dividing the image into RGB planes in the transmitting step and then using stitching techniques in the receiving step, segmentation techniques can be used to further increase the security of our system. We can also improve the current program by providing coloured images of the patients using RGB planes.

VII. References

[1] Usman, M. A., & Usman, M. R. (2018, January). Using image steganography for providing enhanced medical data security. In *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)* (pp. 1-4). IEEE.

[2] Ying Li and Bo Cheng, "An improved k-nearest neighbor algorithm and its application to high resolution remote sensing image classification," 2009 17th International Conference on Geoinformatics, 2009, pp. 1-4, doi: 10.1109/GEOINFORMATICS.2009.5293389.

[3] Hathaliya, J. J., & Tanwar, S. (2020). An exhaustive survey on security and privacy issues in Healthcare 4.0. *Computer Communications*, *153*, 311-335..

[4] Rodriguez-Colin, R., Claudia, F. U., & Trinidad-Blas, G. D. J. (2007, February). Data hiding scheme for medical images. In *17th International Conference on Electronics, Communications and Computers (CONIELECOMP'07)* (pp. 32-32). IEEE..

[5] Yang, Y., Xiao, X., Cai, X., & Zhang, W. (2020). A secure and privacy-preserving technique based on contrast-enhancement reversible data hiding and plaintext encryption for medical images. *IEEE Signal Processing Letters*, *27*, 256-260.

[6] Parah, S. A., Ahad, F., Sheikh, J. A., Loan, N. A., & Bhat, G. M. (2017). A New Reversible and high capacity data hiding technique for E-healthcare applications. *Multimedia Tools and Applications*, *76*(3), 3943-3975.

[7] Sajedi, H., & Yaghobi, S. R. (2020). Information hiding methods for E-Healthcare. *Smart health*, *15*, 100104.

[8] Al-Dmour, H., & Al-Ani, A. (2016). Quality optimized medical image information hiding algorithm that employs edge detection and data coding. *Computer methods and programs in biomedicine*, *127*, 24-43.

[9] Sanivarapu, P. V., Rajesh, K. N., Reddy, N. V., & Reddy, N. (2020). Patient data hiding into ECG signal using watermarking in transform domain. *Physical and Engineering Sciences in Medicine*, *43*(1), 213-226.

[10] Larrucea, X., Moffie, M., & Mor, D. (2021). Enhancing GDPR compliance through data sensitivity and data hiding tools. *JUCS-Journal of Universal Computer Science*.

[11] Rajendran, S., Kulkarni, V., Chaudhari, S., & Gupta, P. K. (2020). An update on medical data steganography and encryption. In *Recent Trends in Image and Signal Processing in Computer Vision* (pp. 181-199). Springer, Singapore.

[12] Huang, H. C., Fang, W. C., & Lai, W. H. (2012, May). Secure medical information exchange with reversible data hiding. In *2012 IEEE International Symposium on Circuits and Systems (ISCAS)* (pp. 1424-1427). IEEE.

[13] Sreejith, R., & Senthil, S. (2017, April). A novel tree based method for data hiding and integrity in medical images. In *2017 IEEE International Conference on Electrical, Instrumentation and Communication Engineering (ICEICE)* (pp. 1-4). IEEE.

[14] Sharma, N., Anand, A., & Singh, A. K. (2021). Bio-signal data sharing security through watermarking: a technical survey. *Computing*, *103*(9), 1883-1917.

[15] Karakis, R., & Guler, I. (2018). Steganography and medical data security. In *Cryptographic and Information Security* (pp. 627-660). CRC Press.