Comparative Study of Image Encryption and Image Steganography using

Cryptographic Algorithms and Image Evaluation Metrics

Surya Teja Chavali[1], Charan Tej Kandavalli [2], Sugash T M[3], Dr. Prakash G.[4]

Department of Computer Science and Engineering[1-4]
Amrita School of Engineering, Bengaluru,
Amrita Vishwa Vidyapeetham, India.
1 bl.en.u4aie19014@bl.students.amrita.edu, 2
bl.en.u4aie19013@bl.students.amrita.edu, 3
bl.en.u4aie19062@bl.students.amrita.edu, 4
g_prakash@blr.amrita.edu

Abstract. In this paper, steganography is carried out by converting the message hidden into ciphertext using renowned cryptographic algorithms like Advanced Encryption Standard (AES) and Triple Data Encryption Algorithm (TDEA or Triple DEA). The embedding technique produces a Stego image. Considering this as the first level of encryption, we then encrypted the whole stego image using AES and TDEA algorithms, giving us the second layer of encryption. The encrypted stego image will be returned in enc format, unreadable without the proper key. We have good resistance to brute force, statistical, differential attacks due to the experimental results of two tiers of protection. In addition to the encryption and decryption process, this paper also talks about Image evaluation metrics like PSNR value and SSIM value, performed on stego images, which use ciphertext from AES and TDEA algorithms gives the statistical information.

Keywords: Image Encryption & Decryption, Image Steganography, Hashing, SHA – 256, MD5, AES, Triple DES, PSNR, and SSIM.

1 Introduction

Our transactions, communications, personal information, and private data are all in desperate need of security. As cyber-attacks become more common, we must protect our data using well-established and secure cryptographic methods or approaches. Cryptography studies secure communications mechanisms that only the sender and authorized receiver may use or access the data using a key. Steganography conceals data in a non-suspicious manner. The method of hiding data inside an image file is called Image Steganography. The cover picture is chosen for this purpose, while the stego image is acquired following steganography. In this paper, we employed well-known cryptographic techniques such as the Advanced Encryption Standard (AES) and the Triple Data

Encryption Algorithm (TDEA or Triple DES) for encrypting the message that needs to be hidden in the cover image.

We considered a text message a private key, applied AES encryption for that message, and used TDEA encryption for the exact plain text. This image steganography is done embedding the ciphertext by systematically altering the pixel values so that the actual data of the cover image is not much disturbed and not observable for a person other than the sender and authorized receiver. Then, we performed some Image evaluation metrics on images with stego images with ciphertext from AES and Stego images with ciphertext from TDEA. We then compared analyzed the performance of these two kinds of images. Image encryption may be described as the act of encoding a secret image with the assistance of an encryption algorithm so that unauthorized users cannot access it. Once we had our Stego Image ready, we utilized the same AES TDEA encryption techniques to encrypt the entire image using a private key as the second level of encryption. We return the encrypted picture in "enc" format when Image Encryption is completed, which cannot be accessed without the actual key used in the encryption procedure. With this, our second and last layer of encryption is now complete. The comparison and analysis part discusses the Image evaluation metrics carried out on stego images. We considered metrics like Peak Signal – To Noise Ratio (PSNR) and the Structural Similarity Index (SSIM) to give us the statistical information between the cover (original) and stego images when encoded with different ciphertexts from different algorithms used.

2 Literature Survey

Different secret text embedding techniques along with DES encryption have been considered previously, such as in [14]. The authors have used K – means clustering to group all the 3-pixel values in the image (R, G, B) and then store the secret text in partitions using LSB and MSB. Unlike this, we have found and characterized a research gap for this application. As a reason, we have used a unique approach, we have used ASCII to bit conversions, for carrying out the process of steganography.

In [16], the authors have considered a fuzzy logic system, to make wise approximations, as to where and how is the textual data is placed and used neural networks to hide the data in the embedding process. This shows that the scope of steganography is extended to the field of deep learning too.

Several comparative techniques have been considered for understanding image distortion before. In [6], Ching-Chiuan Lin has proven that grayscale images obtained after the decryption process are almost the same as the cover images considered for steganography. We have aimed at dealing with RGB images without compromising on any channel and on any pixel value.

Some of the previous works in the domain, have highlighted the similarity of images considering only a single image format. For example, in [7], the authors have explicitly discussed that the mean percentage of correct classification is approximately 95.3%.

In [18], the authors have carried out the evaluation metrics part using MSE and PSNR, showing LSB Substitution with some amount of encryption is the better one compared to PVD. Whereas, we have mentioned SSIM which is more robust and accurate than MSE in comparing the pixel intensity values of two images (discussed more in Section 5).

3 Preliminaries

3.1 Triple DES

After 1990 users started using Triple DES. The reason for using Triple-DES against DES was because of an exhaustive key search that was performed successfully on DES. In those days, users were not ready to switch the cryptographic algorithm because of the cost and time of replacing it; instead, they built a Triple-DES variant of the original DES [11]-[13].

The Normal key size of the DES algorithm is 56, but in Triple-DES, the length of the key size is increased by three times, i.e., the key size of Triple-DES is 168. This increased key size provides more protection against exhaustive searches. 64 bits of data constitute each block, which is undergone a 3 – time encryption using the DES encryption technique

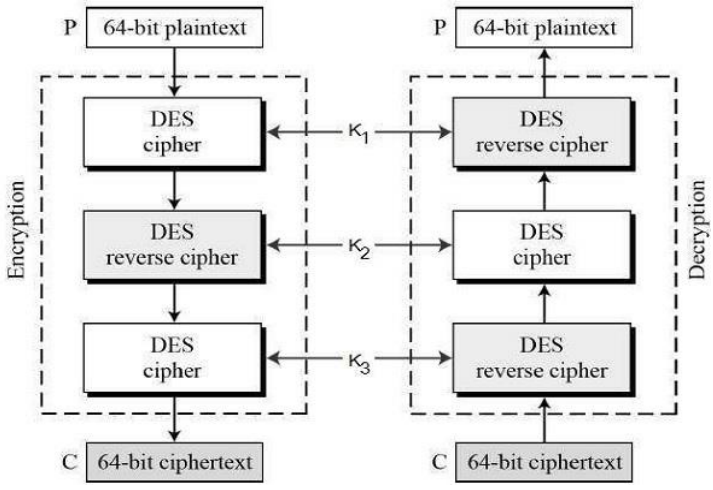

Fig. 1. Structural working of the Triple-DES algorithm

The architecture first uses a single DES to encrypt the plaintext block using key K1. The result of the previous step is decrypted with key K2, and finally, again, the plaintext block is encrypted with key K3 to produce ciphertext. The ciphertext is then decrypted by reversing the encryption algorithm, i.e., K3 is used to decrypt, then K2 is used to encrypt, and lastly, K1 is used to decrypt. Although triple-DES systems are significantly safer and more protected than single-DES, they are much slower to operate.

3.2 Advanced Encryption System (AES)

Advanced Encryption Standard (AES) is the most common and commonly used symmetric encryption technique. It is much faster than triple DES. The DES key size was too tiny and needed to be replaced. When the processing power increased, it was

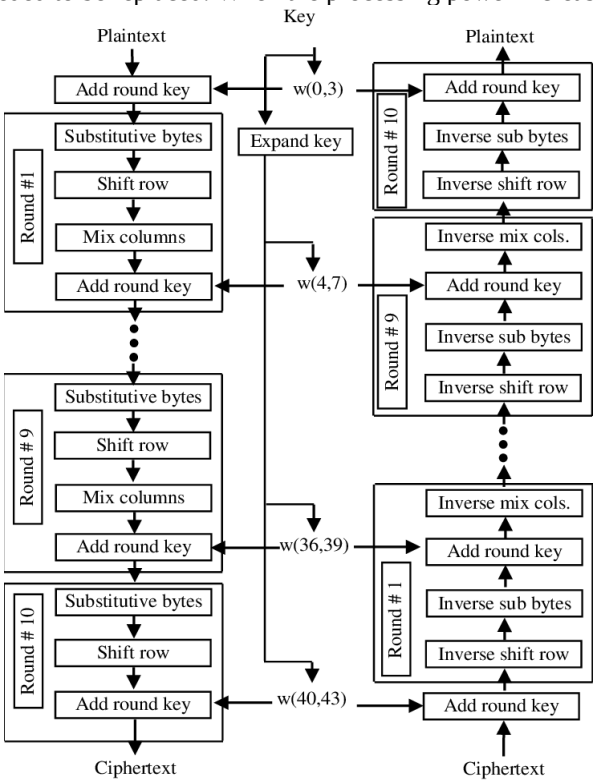

Fig. 2. Structural Working of the AES encryption technique

assumed to be susceptible to an exhaustive key search attack. To overcome this weakness, Triple DES was developed; nevertheless, it was sluggish. Because of these reasons, users switched to AES, and they found it is more efficient than early cryptographic algorithms. AES uses an iterative approach rather than a Feistel cipher approach. The number of iterations in AES is governed by the key length and is adjustable. AES uses ten rounds for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys.

The different 128-bit key round key is used in each round, calculated from the native AES key. AES does all operations in bytes instead of bits. Moreover, it is built on a 'substitution–permutation network.'

Each round comprises four operations in the encryption of plaintext in the AES algorithm. They are as follows."

- Byte Substitution – By using S-Box provided in the design, the 16 input bytes are substituted.
- Shift rows – Each of the matrix's four rows are shifted to the left in Shift rows.
- Mix Columns - Each four-byte column is altered using a specific mathematical algorithm. This function accepts 32 bits from one column as input and replaces them with four new bytes.
- Add Round Key - The 16 bytes of the matrix are considered 128 bits, and they are XOR-ed with the 128 bits of the round key. The 128 bits are translated into 16 bytes in regular rounds, and we repeat the process. However, the ciphertext is produced at the last round after adding the round key.

The decryption of ciphertext in the AES algorithm is similar to encryption, but the difference is that the operations in each block are reversed. As with DES, AES [8] security is only assured if adequately implemented and standard key management is employed.

3.3 Steganography

The technique of concealing a message in a non-secret object without changing its characteristic properties is defined as the art of steganography. Its goal is to hide and defraud the message. It is a sort of covert communication in which messages are concealed through the use of any medium. The key contrast between cryptography and steganography is that cryptography renders the data illegible or hides the meaning, and steganography hides the data's presence. A message can be concealed in several different ways. Some bytes in a file or image are redundant, and they can be substituted with a message without affecting the original message. Different types of steganography exist in images, Steganography in Audio Steganography in Video Steganography in Documents [16], etc.

In our project, we used steganography in digital images. Because digital images arrive in various formats, the algorithms used to process them differ substantially. The approach used in the project is the Least Significant bit [9] and [15] technique. The attacker searches the file for the least essential data and replaces it with the hidden message, usually damaging the code. There are many such techniques, such as Platte based technique in which, using digital photographs as malware carriers, the attackers

encrypt the message and then hide it in a broad palette of the cover image. Techniques like Secure Cover Selection in which the carrier image's blocks must be compared to specified malware blocks can also be used. Nevertheless, the advantages of Least-Significant-Bit (LSB) stenographic data embedding are that it is simple to comprehend, simple to execute, and produces stego-images [1]-[2] with hidden data.

4 Implementation

4.1 Image Steganography

Steganography incorporated within images, in one way, is thought of implementing using a clustering technique, where the pixels are clustered into three regions (R, G, B). Our execution is governed using a novel conversion technique, unlike what is used in [14]. In implementing our project, we used Python as the programming language. In Python, the 'Cryptodomex' module calls certain essential functions for the project like AES, DES, SHA256, etc. Image Encryption and Steganography have been implemented in various forms to analyze different results.

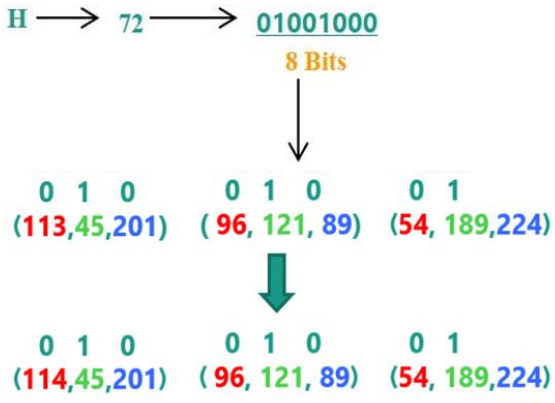

Fig. 3. Encoding the given secret text into the original image to generate a stego image

In order to understand steganography, let us consider a message to be hidden inside the image - "Hello." First, we need to convert each character to its respective ASCII value. Unlike the approach mentioned in [19], where the ASCII of all pixels is converted to a matrix for further encoding, the ASCII value in our case is converted to its respective 8 – bit binary number. Each pixel contains three values corresponding to Red, Green, and Blue, respectively. From the image, we take out three pixels which give nine values. These nine values are modified according to 8 – bit binary values by following specific conditions.

If the bit is 0 – change its respective pixel intensity value as even and if the bit is 1 – change its respective pixel intensity value as odd. If the last pixel intensity value is

even, it conveys the message continues, or else if it is odd, it tells that the message is terminating.

To carry out the above process, the user has to enter a password in the encryption part, which is nothing but the key in this process. This particular key is hashed using the SHA – 256 algorithm. We adopted the Cipher Block Chaining mode in the AES algorithm to perform encryption in the later stages. An initialization vector (IV) of a certain length is utilized in cipher block chaining. By combining this with a single encryption key, individuals and organizations may securely encrypt and decrypt huge quantities of plaintext. Cipher block chaining is a method for encrypting and decrypting significant plaintext inputs that involve building a cryptographic chain in which each ciphertext [5] block is dependent on the previous one. To begin a cipher blockchain, XOR is considered the first block of the many, with an IV, which is a one-of-a-kind conversion function with a fixed length, to generate a pseudorandom or random output. The ciphertext block, an encrypted text format that can be decrypted with the correct key, is created by encrypting the XOR output with a cipher key.
CBC decryption works in a similar but separate manner. The process does not start with the last ciphertext block, as it does with similar decoding algorithms. In reality, because all inputs are present, it might all happen simultaneously.
After combining the second ciphertext block with the cipher key, the output is XORed with the initial ciphertext block to generate the second plaintext block. The IV is substituted with the preceding ciphertext block during the decoding process.

Padding is a critical method to be applied during encryption and decryption. Padding is a cryptographic phrase that refers to various processes that all entail appending data to the start, middle, or conclusion of communication before encryption.
This step ensures that our secret message is encrypted and appropriately hidden inside the image in the manner mentioned in the above explanations.

We should always consider all of our design and implementation corner cases. While the receiver may be confident that the data is secure enough while the transmission is taking place, attackers who are clever enough might still be able to retrieve the file and decode the text inside it. Therefore, we have considered Image Encryption a viable step, which is discussed in the latter.
In the above procedure, the algorithms AES and T – DES are resisting the brute force approach as the key size is high, i.e., 256 and 192-bit keys for AES and T – DES respectively. The length of the encryption key governs whether or not a brute-force attack is practical, with longer keys being significantly more difficult to breach than smaller ones. Applying brute-force methods, AES 256 is essentially impenetrable. Whereas a 56-bit DES key can be broken within a day, given current computing capability, breaking AES will take billions of years. At the time of writing this paper, the fastest supercomputer is the Fugaku supercomputer, which can computer 442 Peta Flops (Flops - floating-point operations per second), unlike the Intel Core i9 which can perform only close to a100 Giga Flops. Even though, this fastest supercomputer itself takes (~)30 trillion trillion trillion years (which is still impossible). It would be unwise

for hackers to launch such an attack on a normal PC. However, no encryption system is completely secure.

4.2 Image Encryption

Image Encryption is one of the techniques used in digital image protection, and its primary premise is to encrypt the digital content included in a digital image. As a result, the appearance and the original digital image are completely independent encrypted images, preventing direct viewing of the digital image's content. We are precisely replicating this process in Python. Let us now discuss the structure of this part.

At first, the user is requested to enter a password (key for the AES algorithm), which is bound to the image file. The key is then hashed using the SHA – 256 algorithm. Key hashing is necessary to provide a more secure and customizable technique for obtaining data. The key is then passed into the digest function. A hash function computes a message digest function, a finite size numeric encapsulation of the contents of a message. Encrypting a message digest can be used to establish a digital signature. Next, the image file is directly passed into the encryption function, which processes the entire data – each pixel value gets encrypted, and the file is transformed into another image. However, directly sending this image to the receiver may not be the most feasible step here, as the attacker might still trap and disrupt the image, even though he cannot retrieve the hidden data.

Therefore, we have converted the encrypted image file to a generic encoded file with an extension '. enc' (known as an enc file), which protects the file against illegal access or aids in configuring the file for a specific purpose of Internet communication. The .enc file generated for our stego image [3] contains information that an average human cannot understand (as shown in Figure 4). You can't open or install ENC files like you can decrypt APK files since they're encrypted using powerful cryptographic algorithms.

This particular file is then sent to the receiver for further proceedings. The above-mentioned is also performed using the Triple-DES algorithm with MD5 as the hashing technique for the key. We have also considered a method where the hidden test is not encrypted using an algorithm and is directly stored in the image. These procedures will help us analyze the results to conclude what a viable technique to consider in all three of the methods mentioned above is.

5 Results and analysis

The final output obtained after encrypting the image, the receiver gets a file that looks like the below image.

Fig. 4. The generic encoded file (.enc), seen at the receiver side after the final step of encryption

Fig. 4. clearly portrays that no unauthorized person (attacker/hacker) will understand this file format, as it consists of characters and symbols that are meaningless.

The proposed approaches – visibility, security, durability, and capacity are all measured using evaluation measures. Since we are comparing the same image before and after steganography, PSNR and SSIM are the best suitable, unlike other evaluation methods. [18]. The metrics most often used are discussed below and the functionality they measure.

First is the 'Peak Noise to Signal Ratio,' which gives us the suggested steganography method's quality, robustness, and transparency are determined using the Peak Signal to Noise Ratio (PSNR) [6]-[7]. PSNR is the ratio of the cover image's maximum quality representation to the stego image's maximum quality representation. It is formulated as below:

$$PSNR = 10\log_{10}((L-1)^2/MSE)$$

The subsequent evaluation to be considered is the Structural similarity index (SSIM) between the original image and the stego image. It is used to measure the similarity between two given images. It is a complete reference measurement that takes two images— the cover image (original image) and a processed image—from the same image capture. Typically, the image is compressed after it has been processed. It may be obtained, for example, by storing image data as a JPEG (of whatever quality level) and

then reading it back in. It is a value between 0 and 1, where 0 indicates the similarity between 2 images is nil, and 1 shows where the images are precisely the same. This metric for determining quality is dependent on the computation of three primary factors: brightness (luminance), contrast, and the structural or correlation term. This index is the result of multiplying these three factors together. It is calculated as the below formula:

$$SSIM(x,y) = [l(x,y)]^{\alpha}.[c(x,y)]^{\beta}.[s(x,y)]^{\gamma},$$

where:

l – indicates the luminance between two images (used to differentiate the brightness of the two images),

c – indicates the contrast between two images (used to differentiate the brightest and darkest region of two images),

s – displays the structure (used to assess the local luminance pattern between two photos to identify the similarity and dissimilarity of the images),

α, β, γ – are the positive constants.

SSIM and FSIM are normalized from a representation standpoint, while MSE and PSNR are not. As a result, SSIM and FSIM can be dealt in a more straightforward manner than MSE and PSNR. PSNR and MSE measure the absolute errors, but SSIM and FSIM are based on perception and saliency. When the noise level rises, the recovery quality of the output image suffers as well. SSIM error map shows the area which is

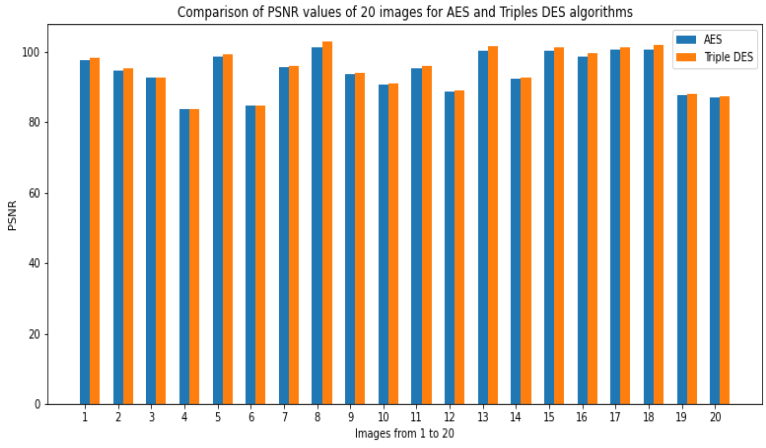

Fig. 5. The comparative depiction of PSNR values for 20 images, when the encryption was done using AES and TDEA algorithms

more affected by noise. Hence it is easy to reconstruct the distorted image. As a result, we may conclude that, from a human visual standpoint, SSIM and SSIM are superior to MSE and PSNR measurements.

Fig. 5. shows the comparison between the PSNR values for all the 20 sample images when performed steganography using AES and Triple DES. We can infer that Triple-DES has slightly higher PSNR values than the images where the text is encrypted with AES

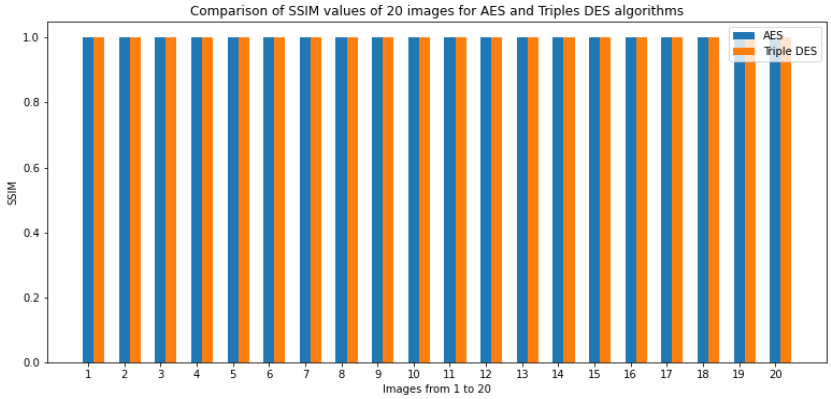

Fig. 6. The comparative depiction of SSIM values for 20 images, when the encryption was done using AES and TDEA algorithms

Fig. 6. tells us that the SSIM values for all 20 images are above 0.9 (exactly the same up till the 6th decimal point) for both the encryption techniques, which indicates the similarity between the original and the stego image is almost the same.

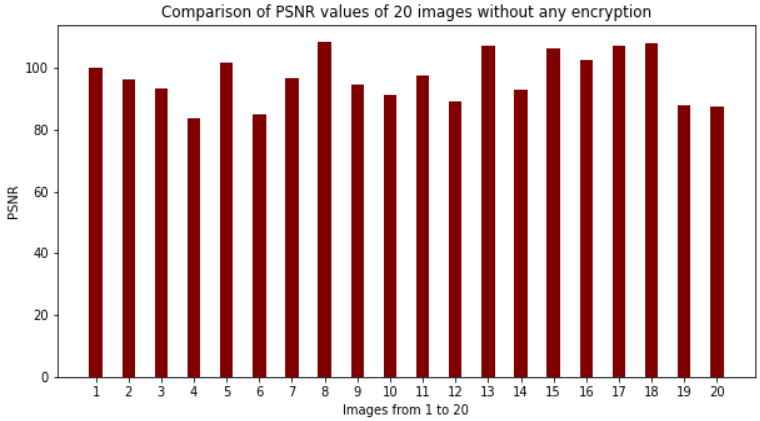

Fig. 7. The comparative depiction of PSNR values for 20 images, when the secret message was not encrypted at all and was directly placed into the image

The higher is the PSNR and SSIM [10] values, the more is the quality and the similarity of the two images, respectively. To better understand our model, we have considered a total of 20 high-quality images, performed the above methods discussed in the implementation part, and later evaluated them using the PSNR and SSIM metrics.

We have also considered a scenario where the user might want to hide the text without encryption. In such a case, Fig. 7. shows that the PNSR values range from 80 to 100, indicating a high variation if the secret message is not encrypted using any cryptographic algorithm. Keeping in mind that the sender sometimes has to use different image formats, we have also performed the above operations listed in the implementation part for three different images format – 'JPG / JPEG,' 'PNG,' 'TIFF'.

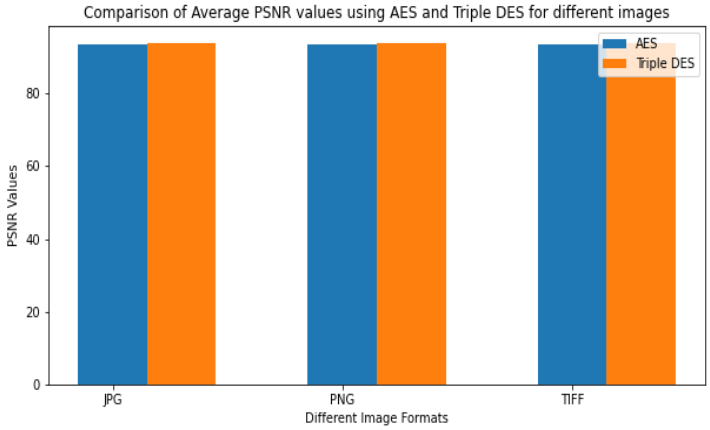

Fig. 8. The comparative depiction of Average PSNR values for different image file formats – JPG / JPEG, PNG, TIFF, when the message is encrypted using AES and TDEA

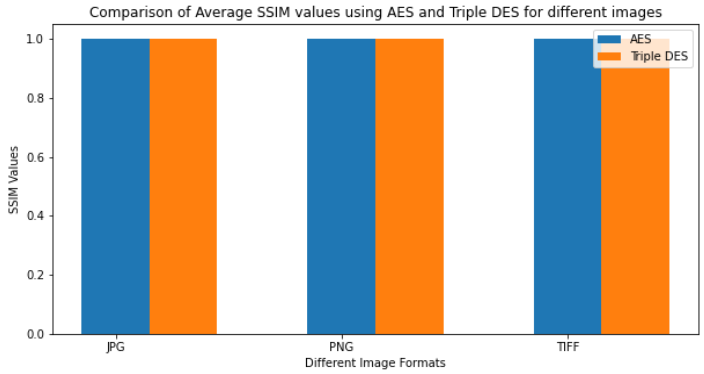

Fig. 9. The comparative depiction of Average PSNR values for different image file formats – JPG / JPEG, PNG, TIFF, when the message is encrypted using AES and TDEA

In every image format that has been considered, Triple-DES encrypted stego images have a slight edge over the AES ones in terms of their corresponding PSNR values. As far as the SSIM values are concerned, the difference in the case of both the algorithms is very minimal.
The above analysis shows us that the model is very robust and flexible, as the user can choose any of the three mentioned file formats, and the difference is not noticeable at all (as shown in Figure 8 and Figure 9).

6 Conclusion and Future Scope

Image encryption models are by far the most considered ones in the secure image processing domain. Image steganography began to mix with the science of computer vision with the introduction of Image steganography, which is also being studied by traditional computer vision researchers. The regions of picture steganography are broadened by combining research fields.
To the best of our knowledge, we have demonstrated Image Steganography using a unique logic of converting the letters of the text into their 8-bit binary equivalent via their ASCII codes, using two different encryption techniques – AES and Triple DES, to obtain a stego image which has a similarity index. Also, to ensure an advanced level of security, we have executed Image Encryption on top of this stego image so that no unauthorized user (attacker/hacker) will be able to decrypt the image and later decode the text. Moreover, the generated file from the image encryption is in '. enc', a standard encoding file for secure file transfers. Any attacker trying to trap this file in between the transmission will not be able to understand what type of a file this is (i.e., a .doc file, pdf file, image file, video file, etc.). We have later evaluated our work using Image Evaluation metrics such as PSNR and SSIM and deduced that Triple DES is the safest and most viable algorithm for Image Steganography.
Nonetheless, the results obtained from the graphs are of high significance and non-trivial. Also, AES is very close to the results obtained from Triple-DES encrypted stego images. With the help of this project, one can easily send secured messages that too in an image format, with cryptographic algorithms of their choice. For the user to have a real-time experience, we have also created a UI / UX interface to make it look much more appealing.

The proposed project is still under development, i.e., as a part of future enhancement [4], one can extend this paper to implement other encryption techniques such as RSA, Blowfish and Two fish Ciphers, etc. A person trying this paper can also evaluate their images using different metrics such as RMSE[17], DSSIM, FSIM, etc.

References

1. Malathi P, Manoaj M, Manoj R, Vaikunth Raghavan, Vinodhini R E, October. 2017. Highly Improved DNA Based Steganography. https://doi.org/10.1016/j.procs.2017.09.151

2. Mathivanan P, Balaji Ganesh A, November. 2020. QR code-based color image stego-crypto technique using dynamic bit replacement and logistic map. https://doi.org/10.1016/j.ijleo.2020.165838
3. Chin-Nung Yang, Shen-Chieh Hsu, Cheonshik Kim, November. 2016. Improving stego image quality in image interpolation-based data hiding. https://doi.org/10.1016/j.csi.2016.10.005
4. Shounak Shastri, V. Thanikaiselvan, March. 2019. Dual image reversible data hiding using trinary assignment and center folding strategy with low distortion. https://doi.org/10.1016/j.jvcir.2019.03.022
5. A.H.M. Kamal, Mohammad M. Islam, January. 2017. Enhancing embedding capacity and stego image quality by employing multi predictors. https://doi.org/10.1016/j.jisa.2016.08.005
6. Ching-Chiuan Lin, February. 2011. An information hiding scheme with minimal image distortion. https://doi.org/10.1016/j.csi.2011.02.003
7. Aladine Chetouani, Azeddine Beghdadi, Mohamed Deriche, Aug. 2010. Statistical Modeling of Image Degradation Based on Quality Metrics. https://doi.org/10.1109/ICPR.2010.180
8. Inas Jawad Kadhim, Prashan Premaratne, Peter James Vial, Brendan Halloran, February. 2019 . Comprehensive survey of image steganography: Techniques, Evaluations, and trends in future research. https://doi.org/10.1016/j.neucom.2018.06.075
9. Osama F. Abdel Wahab, Aziza I. Hussein, Hesham F.A. Hamed, Hamdy M. Kelash, Ashraf A.M. Khalaf, 23 March. 2021. Efficient Combination of RSA Cryptography, Lossy and Lossless Compression Steganography Techniques to Hide Data. https://doi.org/10.1016/j.procs.2021.02.002
10. Yucel Inan, 17 October. 2018. Assessment of the Image Distortion in Using Various Bit Lengths of Steganographic LSB. https://doi.org/10.1051/itmconf/20182201026
11. Alexander Wong, May. 2012. Perceptual Structure Distortion Ratio: An image quality metric based on robust measures of complex phase order. https://doi.org/10.1109/CRV.2012.15
12. Yang Ren-Er, Zheng Zhiwei, Tao Shun, Ding Shilei, Jan. 2014. Image Steganography Combined with DES Encryption Pre-processing. https://doi.org/10.1109/ICMTMA.2014.80
13. Manoj Kumar Ramaiya, Naveen Hemrajani, Anil Kishore Saxena, Feb. 2013. Security improvisation in image steganography using DES. https://doi.org/10.1109/IAdCC.2013.6514379
14. Bhagya Pillai, Mundra Mounika, Pooja J Rao, Padmamala Sriram, Sept. 2016. Image steganography method using K-means clustering and encryption techniques. https://doi.org/10.1109/ICACCI.2016.7732209
15. Nidhi Menon, Vaithiyanathan, Dec. 2017. A Survey on Image Steganography. https://doi.org/10.1109/TAPENERGY.2017.8397274
16. Manohar N, Peetla Vijay Kumar, May 2020. Data Encryption & Decryption Using Steganography. https://doi.org/10.1109/ICICCS48265.2020.9120935
17. S. Sravani, R. Ranjith, July. 2021. Image Steganography For Confidential Data Communication. https://doi.org/10.1109/ICCCNT51525.2021.9579814
18. Asha Asok, Poornima Mohan, April. 2019. Implementation and Comparison of different Data Hiding Techniques in Image Steganography. https://doi.org/10.1109/ICOEI.2019.8862750
19. Anusha M., Bhanu K.N., Divyashree D, July. 2020. Secured Communication of Text and Audio using Image Steganography. https://doi.org/10.1109/ICESC48915.2020.9155715