National College of Ireland

Securing secret data using an enhanced blowfish encryption with Image Steganography using Pixel Indicator Technique

MSc Internship Cyber Security
MSc Cyber Security

Farrukh Hassan Abbas
Student ID: x18210180

School of Computing
National College of Ireland

Supervisor: Michael Pantridge XXX

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Farrukh Hassan Abbas
Student ID:	x18210180
Programme:	MSc Cyber Security
Year:	2020
Module:	MSc Internship Cyber Security
Supervisor:	Michael Pantridge
Submission Due Date:	17/08/2020
Project Title:	Securing secret data using an enhanced blowfish encryption with Image Steganography using Pixel Indicator Technique
Word Count:	6929
Page Count:	27

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

I agree to an electronic copy of my thesis being made publicly available on NORMA the National College of Ireland's Institutional Repository for consultation.

Signature:	
Date:	17th August 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Securing secret data using an enhanced blowfish encryption with Image Steganography using Pixel Indicator Technique

Farrukh Hassan Abbas
x18210180

Abstract

In the modern day of technology world, it is crucial to have a secure communication system. Over the past few years, there have been numerous amounts of data breaches, data leaks, intrusion in communication etc been reported to the authorities. This is due to insufficient security measures while the data is being transmitted. In this paper we will be looking at how we can use Cryptography and Steganography together to secure secret messages. Cryptography is the field of encrypting a human readable text into an unreadable gibberish form making it impossible for the intruder to read the plain text. Steganography on the other hand is the field of embedding a secret information into digital resources such as image, audio, or video making the plain text invisible. In this research A forceful combination of enhanced Blowfish algorithm along with Image Steganography using the Pixel Indicator technique have been brought together to improve data security and protect against brute force attacks, man in the middle attacks and data breaches.

1 Introduction

In the security world we think of it with the acronym CIA. Confidentiality, Integrity, and availability. Confidentiality asks the questions who has access to my data, Integrity causes us to ask the question who can modify my data. Availability causes us to ask the question to do I have access to my data. Let's say for example Alice and bob are having a conversation, and man in the middle joins and intercepts the messages from the conversations. The man in the middle can now read whatever is going on in that conversation. This comprises the confidentiality of the conversation. Hence confidentiality of the conversation becomes crucial. It is important to manage who can read what. To overcome this security problem combination of Blowfish encryption algorithm and Image Steganography can be used to make it complex for the intruder in the system to read through the message. Cryptography plays a vital role in our daily life. From online transactions to our emails, different cryptographic algorithms are being used to develop a secure system [1]. Cryptography ensures that the message sent at one end remains confidential and should be received only the intended receiver at the other end. There are 3 key principles of cryptology. Principles of Cryptography:

Encryption: Encryption is the process of converting data into an unreadable form in order to protect its privacy during data transfer, reception and storage. Encrypted data has to be decrypted. Encryption and decryption require a key to be used in the process while data may seem to be scrambled it can still be read and understood by desired recipients.

Authentication: Ensuring that the sender claimed the string of data is in fact the actual sender. This works by the sending computer or device performing some action that the receiver only knows the correct sender can do. When the receiver sees that action, **it authenticates that message.**

Integrity: Proper cryptography also ensures message integrity which means that the messages are communicated accurately and not intercepted or altered during the communication path.

There has been diverse combination of research been done in the field of cryptography and steganography together in order to make complex systems. Different types of encryption algorithms such as AES, DES, RSA along with different steganography techniques (Image, Audio, Video) has been carried out to encrypt and hide the data. It is true that cryptosystems are also vulnerable. Once the attacker has found the secret decryption key the attacker can decipher into plain text. Same goes with the Steganography once the attacker knows which type of steganography techniques has been put to hide the data it will be easy to decode it. Main objective of the proposed system is that a unique combination of Blowfish encryption along with Image Steganography using Pixel Indicator technique is used in order to protect the confidentiality of secret messages against Brute force and Man in the middle attacks making it impossible for the attacker to understand what's going on in the secret conversation. Cryptography on its own to some extent provides a certain level of security but once the attacker has gained access to the secret keys there will be no use of that cryptographic algorithm applied. Same goes with the Steganography once the attacker identifies the type of steganography technique applied, an attacker can break the steganography easily by manipulating it. Which leads us to our solution that we have proposed in this research. The solution consists the use of both steganography and cryptography to maintain the confidentiality of secret messages being exchanged on the internet. In our proposed approach, A plain text is encrypted using the enhanced version of Blowfish algorithm. We first enhanced the Blowfish algorithm by increasing its text size from 64 bit to unlimited text size. Meaning any amount of string can by encrypted using the Blowfish algorithm. Once the text has been encrypted it will then divide into 4 different chunks of data. These 4 chunks of data will then be embedded into 4 different random images using the Pixel Indicator technique.

Having done my research, I did not come across any research where I saw this combination to perform a task. This motivated me come up with a new and unique version of securing the conversations with the use of Blowfish encryption and Image steganography.

2 Related Work

In order to appreciate the paper's novelty, it is crucial to conduct a thorough literature review to understand the related research and efforts of the authors to publish papers in the discipline of steganography and encryption. The existing work can be enhanced and new approaches can be discovered by studying the previous research carried out in the respective fields

2.1 Steganography

The term steganography can be defined as embedding files within an innocent, innocuous, and mild cover work such that this file cannot be detected by anyone without approach to the cover work (Toumazis, 2009) [1]. Furthermore, Channalli et al. (2009) described that steganography is a Greek word for covered writing which means "hiding in plain sight." Hence steganography is defined as the art of inconspicuously hiding data inside data [2]. However, according to Kumar and Pooja (2010), steganography is not only the hiding data or information but it is also an attempt to cover the existence of the embedded file [3].

Some basic features of steganography include unsuspiciousness, undetectability, transparency (clearness), invisibility, and high embedding capacity. Moreover, the algorithm should also be stable (holds the data embedded in the cover), independent of the original cover, and temper resistant (embedding new data, modification of embedded data, as well as prevents deletion) (Khairullah, 2019) [4].

The different types of steganography include audio, video, image, and text steganography. Different techniques are applied for the stenography of these types. The hardest form of stenography is text steganography (Malik et al., 2017) [5] which is used to hide text within the text. Text steganography is difficult because the difference between stenographic text and original text the eyes can easily be detected; therefore extreme measures must be put to prevent human eyes from detecting them. Three methods, namely linguistic method, random-statistical generation, and format-based can be applied here (Doshi et al., 2012). [6]

2.1.1 Image steganography

Although various formats can be used to transmit images, the Joint Picture Expert Group (JPEG) format is one of the most common image formats (Provos and Honeyman, 2003) [7]. Image steganography permits the image and text data to be hidden within an image. The techniques used here include Discrete Cosine Transform, Transform Domain, Spatial Data Embedding, Spread Spectrum, Filtering and Noising, Masking, MSB, and LSB (Kamble et al., 2013). MSB technique is not a good choice for steganography system as it makes the steganographic image to be suspicious to human.

Steganographic techniques are usually divided into two groups: spatial and frequency domain [8]. The message is embedded in image pixels' [9] Least Significant Bit (LSB) [10] in the first group. This technique is vulnerable to attacks such as compression

and low-pass filtering but, it is easy to implement and has a high capacity. For example, Raja [10] demonstrated various LSB using the Optimal Pixel Adjustment Process (OPAP) to improve the quality of stego-image with low computational complexity. Moreover, imperceptibility and sensitivity issues in the spatial domain can also be improved in this hiding method. The message is embedded into frequency coefficients of images in the second group [11]. The imperceptibility and robustness issues found in the spatial domain can be overcome by this hiding technique.

JPEG is a typical technique for compression of image [12]. JPEG is used to hide data in many steganography techniques such as Outguess, JP Hide Seek, and JSteg. Discrete Wavelet Transform (DWT) has also been applied in recent studies apply due to its broad application in the imperceptibility or capacity [8]. To make secure steganography encoding on the JPEG images, Akbarzadeh, Fard, and Varasteh [13] had suggested a GA evolutionary process. To maximize the quality of the watermarked image, R. Elshafie, N. Kharma and R.Ward [14] presented a parameter optimization using GA. A method to embed data into Discrete Wavelet Transform [15] coefficients operating a mapping function based on the Genetic Algorithm [14] in 4x4 blocks on the cover image is proposed in this paper. After embedding the message to maximize the PSNR OPAP is also implemented in the proposed method.

2.1.2 Pixel Indicator Technique (RGB) vs Least Significant Bit (LSB)

PVD-based steganographic or LSB substitution techniques have been applied by many researchers to develop some efficient colour image steganographic methods. Al-Qahtani et al (2009) have improved the security of the colour steganographic scheme where the secret information bits have not been hidden into each colour pixel in sequential order [16]. A secret pseudorandom value in the embedding process decides payload capacity and embedding sequence of the bits of the secret message into each colour plane adaptively [17]. This indirect approach defined by them certainly enhances the security level. Another technique of colour image steganography which is based on LSB substitution was observed in which the secret message bits are not embedded directly into orders, but are hidden with reference to an indicator colour plane [17]. Parvez & Gutub has recommended another secret key-based colour image steganography [18]. In this technique, some pre-defined secret key is used to spread out the secret message over each colour plane. Nagaraj et al. suggested a modified modified PVD-based steganography method [19]. They used modulus 3 function with PVD in their scheme to transform secret message bits into colour pixels. Later, Prema & Manimegalai [20] recommended a technique to use modified PVD in colour image steganography. In their method, non-overlapping blocks of two consecutive pixels are obtained by the decomposition of an RGB colour image. Two consecutive colour pixels form three different pairs, known as (B,R), (G,B), and (R,G). Based on the difference in colour component pairs, the secret message is embedded. While preserving the acceptable visual quality of the stego-image, they have enhanced the hiding capacity. A block-based smart pixel adjustment process was developed by Yang & Wang [21] in which considers a block of two colour pixels during the embedding process of secret message. However, hiding capacity is not intense in their scheme. Swain [22] has proposed an adaptive PVD-based colour image steganography in which block level of each colour hides the plane the secret message. During the message-embedding process, the horizontal and vertical edges are exploited in each block. The above-mentioned colour

image steganographic techniques work essentially on a colour plane rather than on colour pixels

One of the most easiest and common algorithms used in steganography is the Least Significant Bit (LSB) algorithm. However, it has a major drawback to being destroyed easily by the attackers: therefore, the author (Kanhe et al., 2015) of the paper "Robust Audio Steganography based on Advanced Encryption Standards in Temporal Domain" introduced a technique of integrating the LSB with AES algorithm [23]. One of the best algorithms for sustaining the confidentiality and integrity of the data, the AES algorithm is used to encrypt data, and then the LSB algorithm is used to hide this encrypted data into the cover file.

The author used the above method so that attacker would not be able to access the original data as it will be encrypted. It happened because the AES algorithm will encrypt the original data if the attacker tries to crack the LSB algorithm and get data. The proposed program was aimed at proving that the quality of the file will not be compromised by implementing encryption and steganography. Signal to Noise Ratio (SNR) plot was used to take and analyse the five different sets (Kanhe et al., 2015). The tests showed that SNR values are not affected to a greater extent by integrating the LSB and AES algorithm [23].

2.2 Encryption

Encryption is a technique to scramble data so that it can be understood by only authorized parties. It is a method of transforming the plaintext to ciphertext, in technical terms. We can say that readable text is altered into a text which appears random through the process of encryption. An encryption key is required to undergo encryption. The encryption key is a set of mathematical values that are known to both the recipient and the sender of an encrypted message.

Although encrypted data seems to be random, encryption proceeds in a logical and predictable way so that data can be easily decrypted into plain text by a party receiving it and having the key used to encrypt the data. Secure encryption will be truly complex enough to make it highly unlikely that a third party can decrypt the ciphertext by brute force, that is by guessing.

2.2.1 Blowfish Encryption:

Blowfish encryption algorithm was introduced by Bruce Schneier to replace DES and IDEA. Blowfish encryption algorithm uses symmetric key cryptography to encrypt 64bit block with a variable length ranging from 32 to 448bits which makes it an ideal match for both domestic and exportable use. [24] Blowfish encryption algorithm properties include. It being fast comparatively from other encryption algorithms. Its compact, meaning it can execute in less memory, it is simple meaning use of simple operations such as XOR, Addition etc. Most Importantly Blowfish is labelled as one of the secure encryption algorithms. [25] Serge Vaudenay famous French Cryptographer, in his Ph.D. thesis examined the weak keys in Blowfish. He identified that there is a class of keys that can be detected although it is not broken [26]. S, Joshna performed a comparative Analysis

of Symmetric key algorithms where they analysed AES, Blowfish, and DES algorithm. They concluded that blowfish algorithm has better performance compared to DES, and 3DES. They also stated that symmetric key algorithms are faster than asymmetric key algorithms as well as symmetric key has more secure key encryption. In the table below we can see comparison of various encryption algorithms regarding level of security and vulnerable attacks [27].

Table 1: Comparison of encryption algorithms (Semwal and Sharma, 2017) [27].

Algo & Yr	**Block Size (Bits)**	**Key Len (Bits)**	**No of Rounds**	**Security level**	**Attacks vulnerable**
DES (1977)	64	56	16	Not adequate	Brute force Differential Attack, Men in Middle Attack
3-DES (1978)	64	112-168	48	Vulnerable	Brute force, Differential Attack
CAST-128	40-128	128	12-16	Vulnerable	64 bit version is vulnerable to linear attack
IDEA (1991)	64	64-128	5-8	Vulnerable	linear Attack
AES (2000)	128	128-256	10-14	Excellent	Side Channel Attack
Blowfish (1993)	64	32-448	16	High	Not yet but prone to Key related attacks Boomerang attack
RSA (1977)	Not Fixed	>=1024	Nil	Very High	Brute force & Timing Attack

3 Methodology

Cryptography is one way to secure the communication of confidential data, but the encrypted text can attract an attacker to perform various cryptographic attacks to gain access to the plain text. On the other side there is steganography, steganography is a security mechanism which is used to protect the privacy of the secret messages. Unlike cryptography it does not scramble the words, making it meaningless to the intruder but instead it embeds the secret data into digital media such as image, video or audio etc [28].

"*Hence compared with cryptography, the steganographic process prevents an unintended recipient from suspecting that secret data are being transmitted over a public channel through meaningful cover media.*" [29]

Having studied and research the previous literature we can see that diverse amount of research has been done on combination of different cryptographic algorithms as well as steganographic modes.There are different techniques that can be used in order to hide the data inside the images. Among them, Least Significant Bit (LSB) is one of the well known and widely used techniques. The reason LSB is used widely is because it gives us the flexibility to change the lowest bits to be our message and we will have almost imperceptible change on the actual way the file (Image, audio, video) looks. For example, if we change a number from 400,234 to 400,229 in the image it is not going to have a massive effect because the number is so big that in the grand scheme it makes no difference. LSB image steganography is a popular approach due to its better imperceptibility and payload capacity. Even Though its simple, traditional LSB is vulnerable as the probability of detecting the secret message is high [22]. Therefore, we are using modified version of image steganography known as RGB Pixel Indicator Technique that can increase imperceptibility and robustness of LSB image steganography. Rather than storing the message bits directly in LSB positions, each message bit is encoded and placed in one of the 3 channels of RGB pixel. The selection of a channel is done based on MSB bits of that RGB pixel, whereas even and odd parity values of that selected channel bias the encoding process of the message bit. The demonstrated system proposed will make sure that the information that is being exchanged over the networks are confidential and secure.

The main goal of the proposed system is to stand apart from all the previous methodologies, this system will make sure that there are not any noticeable changes in the quality of the cover images used. To achieve the hypothesis of this paper a set of diverse tools, algorithms software's and matrices are being used.

To develop our proposed system Python is used as the programming language. The reason python is used is because of its clean and readable syntax. It has versatile set of libraries for Data Analysis, Machine learning and Web development framework. To encrypt the plain text Blowfish encryption algorithm was used and developed in python environment. While conduction the literature review, it was observed that the use of Blowfish algorithm was minimum. Having done the research on blowfish, it showed that blowfish has not been broken yet and is labelled as a secure encryption algorithm [25]. In the literature above we learned that researchers have combined different combination of Cryptographic algorithms with Steganographic techniques to ensure the Confidentiality, Integrity and Availability of a data. The decision of choosing Blowfish as an encryption algorithm is unique, as Blowfish algorithm along with Pixel Indicator Technique has never been used before. The results of current approach will help us understand and learn more about how combination of Blowfish algorithm with Pixel Indicator can improve the world of information security to secure secret messages.

We observed in the literature review above, that first the cipher text needs to be converted to ASCII value and then these ASCII values can be converted into a binary. we observed in the previous proposed systems that once cipher text is obtained it is directly hidden inside an Image, audio or video file. What makes the proposed system apart from previous systems are that when cypher text is obtained it is divided into 4 chunks then these 4 chunks will be embedded into 4 different images files using Pixel Indicator Technique. The

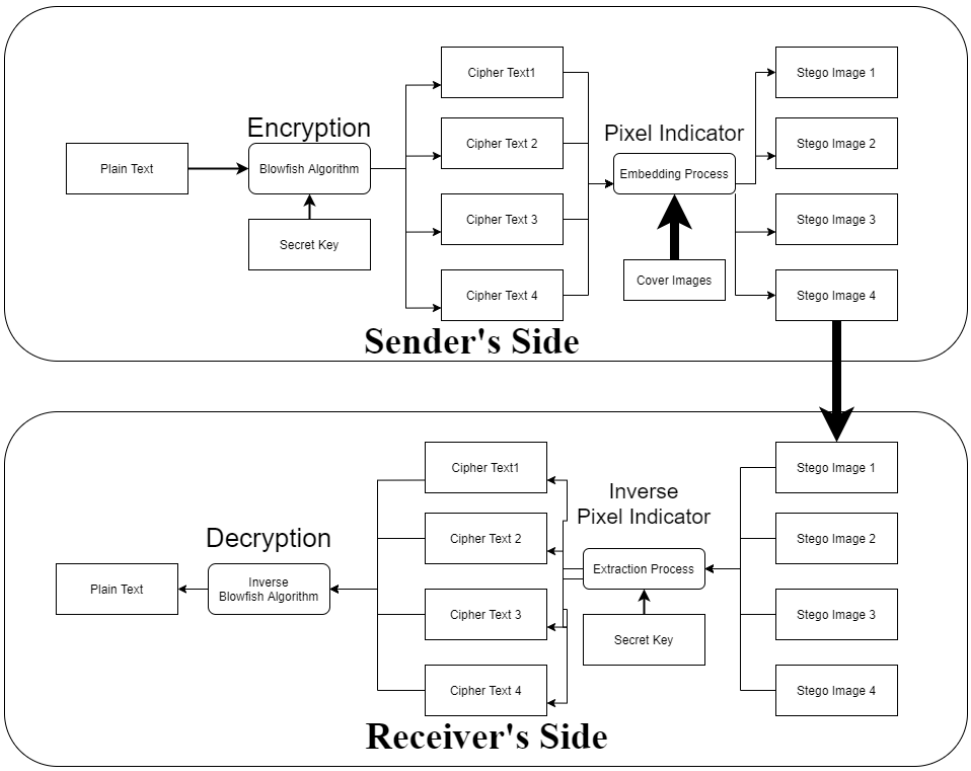
architecture of proposed system will ensure a better security in terms of Confidentiality, Integrity and Availability. For example, even if an attacker gets access to the 4 Stego images he/she first needs to sort the sequence of the images, then the steganography technique, and then the decryption of Blowfish algorithm. This will only leave the attacker with frustration because we are talking about the complexity of proposed system.

In the proposed system at first Least Significant Bit (LSB) technique was being used to embed the cipher text inside the Images. After researching and learning more about LSB technique it was concluded that LSB technique is not suitable for the current systems and is vulnerable to many common attacks. Which lead to decision of choosing a better Steganographic technique such as Pixel Indicator Technique. RGB Pixel Indicator Technique can increase imperceptibility and robustness of LSB image steganography. From literature review, it was concluded that Pixel Indicator Technique is a potential technique to hide the cipher text into image files. In order to develop the proposed system, the main challenge that I faced was working with images. It was hard to process the difference between raw images and stego images. But as mentioned earlier python is a versatile language and contains a lot image libraries which can be used to process and analyse our stego images. An article by geeksforgeeks "Working with Images in Python" helped me a lot and made me understand of how images work in python [30] [31]. To help process and analyse the quality of raw and stego images, Image Quality matrices is used. IQM is a characteristic of an image that measures the processed image degradation by comparing to an ideal image [32]. Following parameters are used to measure the quality between original (raw) image and stego-image. Peaks Signal to Noise Ratio (PSNR), Mean Square Error (MSE), Signal to Noise Ratio (SNR), Structural Similarity Index Measure (SSIM). These terms are explained in detail in Evaluation section.

4 Design Specification

In this section, the architectural view of the system developed will be demonstrated. A step by step block diagram has been created to understand the developed system properly. The figure below demonstrates the architecture of how the proposed system is going to work. The two major components of this proposed system are Blowfish algorithm and Pixel Indicator Technique which helps us gain the confidentiality of the secret data. There are two sides in this system. The sender's side and the receiver's side. At first we will encode the secret key and combine with the secret message (Plain text). Then combined data is encrypted with the enhanced version of Blowfish encryption.

Data encryption occurs via 16 round networks. Each round consists of a key dependent permutation and a key and data substitution. All operations are XORs and addition on 32-bit words. Once the message has been encrypted via Blowfish algorithm it is then splitted into 4 parts and will be embedded into 4 different images to get the 4 different stego-images. By using Pixel Indicator Technique, the secret text will be embedded into the image making it ready to send it to the receiver. All these operations are done in the Senders side.

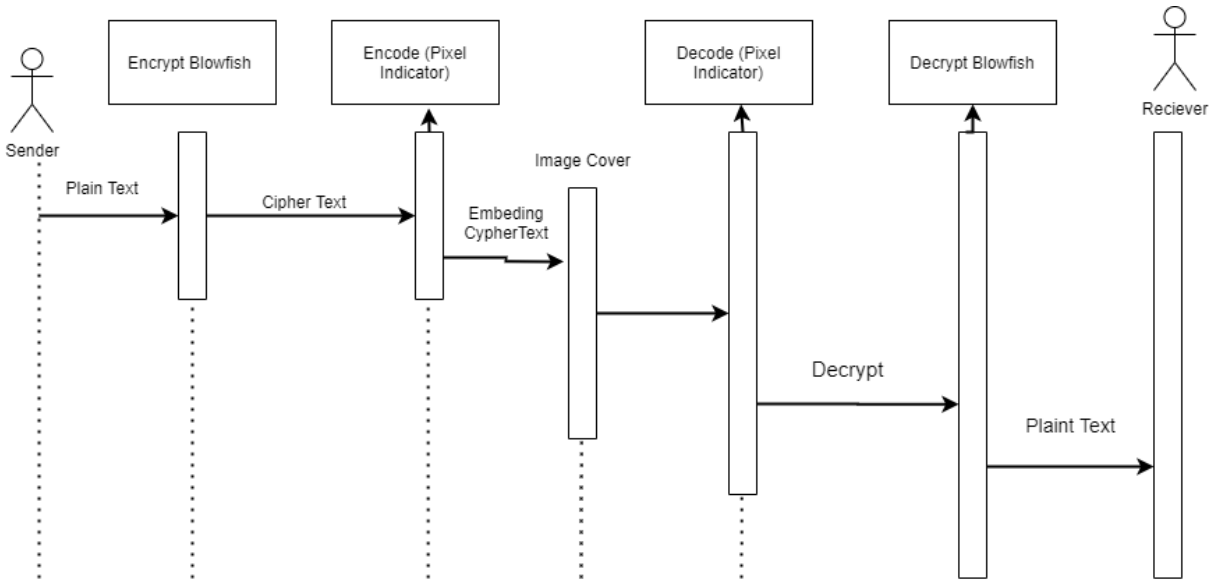


At the receiver's side, the focus for the receiver is to extract the Secret message out of the Stego-image. To achieve this aim, the receiver must perform all the operations in a reverse manner as compared to the sender side. First the receiver should extract the Encrypted text from the stego-image by providing secret key. Once the Encrypted text has been released/extracted by applying reverse Blowfish algorithm, the plain text can be released.

4.1 Sequence Diagram

To detail how operations are carried out in proposed system we will be using Sequence Diagram. It will show the interaction among the objects in sequential order. The 4 major steps have been displayed in the sequence diagram below.

1. Encrypt: Plain text is encrypted using Blowfish algorithm
2. Encode: Cipher text is Encoded into raw images.
3. Decode: The receiver decodes the secret text from the images.
4. Decrypt: Once secret text is decoded; it is time to decrypt it using the Blowfish algorithm.



5 Implementation

This section consists of all the details about the technologies, methods, techniques and algorithms that are used in order to put together the proposed system. We will be talking in details of how we performed certain actions to achieve our confidentiality goals.

Encryption:
The first stage of our Implementation phase is to encrypt the plain text using enhanced Blowfish encryption algorithm. The steps involved in the working of system are as follows.

1. Enter the data to encrypt.
2. Enter the secret key.
3. Secret key will be encoded using **password_embed ()** function
4. Encrypt data will be encoded using base 64 encode method.
5. Combine Encrypt data and Encoded Password.
6. Combined data will be divided into multiple parts and length of each parts will be 4.
7. (It allows to encrypt the data with no limitation on length of data.)
8. Again, combined data will be encoded using manually written **encode ()** function.
9. (Here we are using encoding function for conversion of any symbol/characters/integers into ASCII value.)
10. Encoded data will be encrypted using blowfish encryption algorithm.

11. Encrypted data will split into 4 pieces as encrypted_data1, encrypted_data2, encrypted_data3 & encrypted_data4.

12. It will ask user to provide 4 images path along with extension. 1

13. Later, we will hide these 4 encrypted data pieces into 4 PNG images sequentially using RGB Pixel Steganography Technique.

14. It will display the time required to encrypt data.

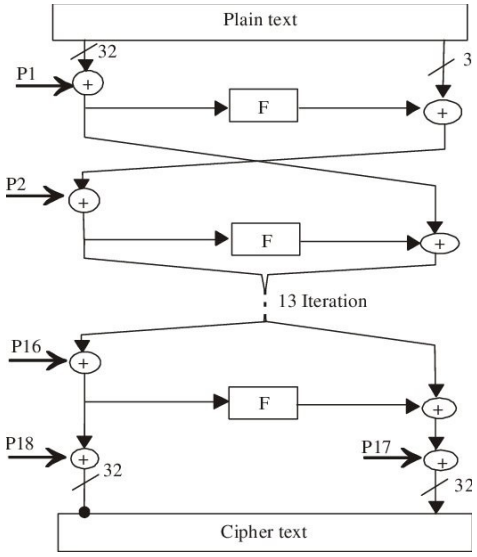


Figure 1: Fiestal structure of Blowfish Algorithm.

The above diagram displays the how Blowfish Encryption works. Blowfish uses Fiestel network structure, which has 16 rounds for encryption and decryption [30]. By performing the actions mentioned in the diagram above the ciphertext is achieved.

Hide:

- For each character in the data, its ASCII value is taken and converted into 8-bit binary
- Three pixels are read at a time having a total of $3 \times 3 = 9$ RGB values. The first eight RGB values are used to store one character that is converted into an 8-bit binary.
- The corresponding RGB value and binary data are compared. If the binary digit is 1 then the RGB value is converted to odd and, otherwise, even.
- The ninth value determines if more pixels should be read or not. If there is more data to be read, i.e. encoded or decoded, then the ninth pixel changes to even. Otherwise, if we want to stop reading pixels further, then make it odd.
- Repeat this process until all the data is hide into the image.

Decode:

- Again, three pixels are read at a time. The first 8 RGB values give us information about the secret data, and the ninth value tells us whether to move forward or not.
- For the first eight values, if the value is odd, then the binary bit is 1, otherwise it is 0.
- The bits are concatenated to a string, and with every three pixels, we get a byte of secret data, which means one character.
- Now, if the ninth value is even then we keep reading pixels three at a time, or otherwise, we stop.

Decryption:

1. Enter the secret key.
2. Secret key will be encoded using **password_embed ()** function
3. It will ask user to provide 4 stego_images paths.
4. Cipher data will be retrieved from all 4 stego_images.
5. We will combine all cipher data from images.
6. Combined data will be decrypted using blowfish decryption method.
7. After that decrypted data will be decoded using manually written decode () function.
8. We will compare the encoded password with decoded data and then we will retrieve decoded data from it.
9. Again, decoded data will be decoded using base 64 decode method.
10. Finally, we will get decoded data as plaintext data.
11. It will also display the time required to decrypt data.

6 Evaluation

IMAGE QUALITY MATRICES:

In the development of image processing algorithms, IQM (Image Quality Measurement) plays an important role. To evaluate the performance of processed image, IQM can be utilized. Image Quality is defined as a characteristic of an image that measures the processed image degradation by comparing to an ideal image. We have considered following image quality parameters.

1. **Mean Squared Error (MSE):**

   In statistics, the mean squared error (MSE) [13] of an estimator (of a procedure for estimating an unobserved quantity) measures the average of the squares of the errors that is, the average squared difference between the estimated values (Stego image) and what is estimated (cover image). It's important to note that a value of 0 for MSE indicates perfect similarity. A value greater than one implies less similarity and will continue to grow as the average difference between pixel intensities increases as well.

   The MSE is a measure of the quality of an estimator—it is always non-negative, and values closer to zero are better.

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2$$

   Where, m and n are width and height of image.

2. **Structural Similarity Index Measure (SSIM):**

   The SSIM method is clearly more involved than the MSE method, but the gist is that SSIM attempts to model the perceived change in the structural information of the image, whereas MSE is actually estimating the perceived errors

   The SSIM value can vary between -1 and 1, where 1 indicates perfect similarity.

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu^2 - y + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

   Where μx ,μy, σx, σy and σxy are the local means, standard deviations, and cross-covariance for images x, y. $C_1 = (k_1 L)2$ and $C_2 = (k_2 L)^2$ .Two variables to stabilize the division with weak denominator. L is the dynamic range of the pixel-values $k_1 = 0.01$ and $k_2 = 0.03$ and by default.

   The above equation is used to compare two windows (i.e. small sub-samples) rather than the entire image as in MSE. Doing this leads to a more robust approach that is able to account for changes in the structure of the image, rather than just the perceived change.

   The parameters to equation include the (x, y) location of the N x N window in each image, the mean of the pixel intensities in the x and y direction, the variance of intensities in the x and y direction, along with the covariance.

3. **Peak Signal-to-noise Ratio (PSNR):**

   It's the ratio between the maximum possible power of an image and the power of corrupting noise that affects the quality of image. The larger the value of PSNR, the more efficient is a corresponding compression method.

$$PSNR = 10\ log_{10}(\frac{(L-1)^2}{MSE}) = 20\ log_{10}(\frac{L-1}{RMSE})$$

   Here, **L** is the number of maximum possible intensity levels (minimum intensity level supposed to be 0) in an image.

4. **Root Mean Square Error (RMSE):**

   RMSE (Root Mean Squared Error) is the error rate by the square root of MSE.

$$RMSE = \sqrt{MSE}$$

   The root-mean-square error (RMSE) is a frequently used measure of the differences between values (sample or population values) predicted by a model or an estimator and the values observed.

5. **Entropy Difference:**

   Image entropy is an important indicator for evaluating the richness of image information; it represents the property of combination between images. The larger the combination entropy of an image, the richer the information contained in the image. The entropy of an image is

$$H = -\sum_{i=0}^{L-1} \pi \ log_2 \ \pi$$

   Where H is the entropy, L is the overall gray-scales of image, pi is the probability of gray level i.

   We calculate entropy difference using the following formula:

$$H_{diff} = H_{stego} - H_{original}$$
$$H_{original} = Entropy\ of\ original\ image.$$
$$H_{stego} = Entropy\ of\ stego\ image.$$
$$H_{diff} = Entropy\ difference\ between\ Stego\ image\ and\ original\ image.$$

6. **Normalised Cross Correlation**

   Normalized cross correlation is the simplest but effective method as a similarity measure, which is invariant to linear brightness and contrast variations. NC (Normalized Cross Correlation) measures the comparison of the stego image and raw image. NC is expressed as follows:

$$NC = \frac{\sum[a(i,j) - Mean(a)[b(i,j) - Mean(b)]}{\sqrt{(\sum[a(i,j) - Mean(a)]^2 * [b(i,j) - Mean(b)]^2)}}$$

7. **Average Differences:**

   AD is simply the average of difference between the raw image (x(i, j)) and stego image (y(i, j)). It is given by the equation

$$AD = \frac{1}{MN} \sum_{i=1}^{M} \sum_{J=1}^{N} (x(i,j) = y(i,j))$$

8. **Maximum Difference:**

   MD is the maximum of the error signal (difference between the raw image and stego image).

$$MD = MAX|x(i,j) - y(i,j)|$$

∗ **Step 1: Writing python code for all above matrices.**

- **Step 3: Display of all Cover images and Stego images with variable message size**
- **Step 4: Histogram analysis of all cover images and Stego images with variable message size**
- **Step 5: Histogram analysis of all cover images and Stego images with composition of Red, Blue & Green colours.**
- **Step 6: Entropy of all Cover images and Stego images with variable message size**
- **Step 7: Normalised Cross Correlation of all Cover images and Stego images with variable message size**
- **Step 8: Time required to encrypt and decrypt the data with variable message size**

1. Dimensions of Raw_Image1: (469, 612, 3) >> (Height, Width, No_of_Channels)
2. Dimensions of Raw_Image2: (478, 868, 4)
3. Dimensions of Raw_Image3: (570, 570, 3)
4. Dimensions of Raw_Image4: (926, 1309, 3)

Case study 1

* **Step 2: Finding values of all image matrices wrt each image**

1. Different Image quality parameters with variable message size of raw_image1 cover Image and its stego_image1.
2. Different Image quality parameters with variable message size of raw_image2 cover Image and its stego_image2.
3. Different Image quality parameters with variable message size of raw_image3 cover Image and its stego_image3.
4. Different Image quality parameters with variable message size of raw_image4 cover Image and its stego_image4.

Image 1	**Image Quality Parameter**							
Message size	MSE	SSIM	PSNR	RMSE	Entropy Difference	Normalised Cross Correlation	Average Difference	Maximum Difference
1000	0.0142	0.9999	66.597	0.119	0.0065	1.00	0.004	1.00
5000	0.0781	0.999	59.20	0.279	0.0312	1.00	0.026	1.00
10000	0.1630	0.999	56.00	0.403	0.0545	1.00	0.054	1.00
Best value	Lower (Close to zero)	Higher(Close to +1)	Higher Value (>40)	Lower (close to zero)	Lower (close to zero)	Higher (close to 1)	Lower (close to zero)	Lower value

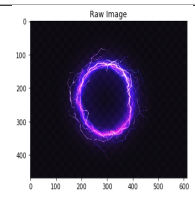
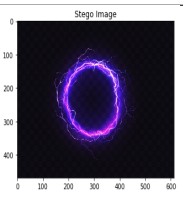
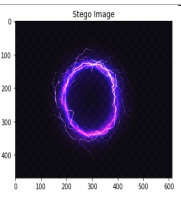
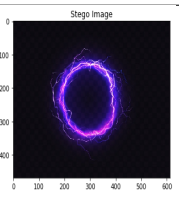
Image 2	**Image Quality Parameter**							
Message size	MSE	SSIM	PSNR	RMSE	Entropy Difference	Normalised Cross Correlation	Average Difference	Maximum Difference
1000	0.0098	0.9999	68.178	0.0994	0.0013	1.0	0.0047	1.00
5000	0.0531	0.999	60.87	0.2304	0.0062	1.0	0.0132	1.00
10000	0.111	0.999	57.64	0.334	0.006	1.0	0.0279	1.00
Best value	Lower (Close to zero)	Higher(Close to +1)	Higher Value (>40)	Lower (close to zero)	Lower (close to zero)	Higher (close to 1)	Lower (close to zero)	Lower value

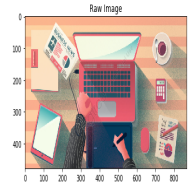
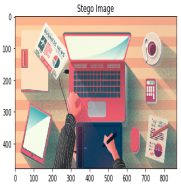
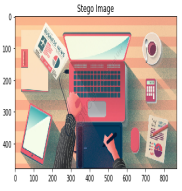
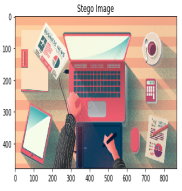
Image 3	**Image Quality Parameter**							
Message size	MSE	SSIM	PSNR	RMSE	Entropy Difference	Normalised Cross Correlation	Average Difference	Maximum Difference
1000	0.0126	0.999	67.124	0.112	0.0041	0.999	0.0042	1.00
5000	0.0681	0.999	59.795	0.261	0.0164	0.999	0.0227	1.00
10000	0.1412	0.999	56.63	0.375	0.028	0.999	0.0472	1.00
Best value	Lower (Close to zero)	Higher(Close to +1)	Higher Value (>40)	Lower (close to zero)	Lower (close to zero)	Higher (close to 1)	Lower (close to zero)	Lower value

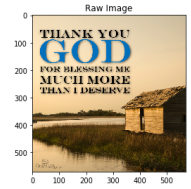
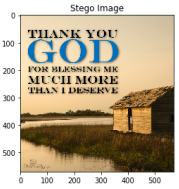
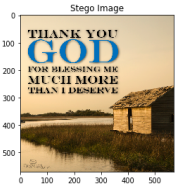
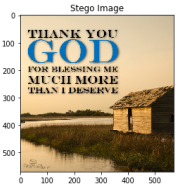
Image 4	**Image Quality Parameter**							
Message size	MSE	SSIM	PSNR	RMSE	Entropy Difference	Normalised Cross Correlation	Average Difference	Maximum Difference
1000	0.0033	0.999	72.826	0.0582	0.0015	1.0	0.0011	1.00
5000	0.018	0.999	65.52	0.134	0.0068	1.0	0.006	1.00
10000	0.0381	0.999	62.316	0.195	0.011	1.0	0.0127	1.00
Best value	Lower (Close to zero)	Higher(Close to +1)	Higher Value (>40)	Lower (close to zero)	Lower (close to zero)	Higher (close to 1)	Lower (close to zero)	Lower value

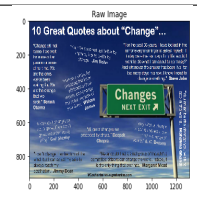
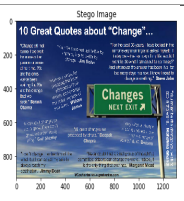
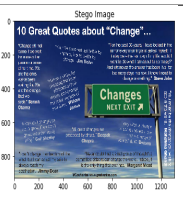
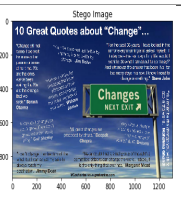
Case study 2

* **Display of all Cover images and Stego images with variable message size**

	Raw Image	Stego Image with variable message size		
Message Size		1000	5000	10000
Image1				

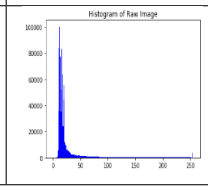
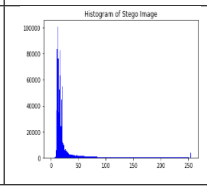
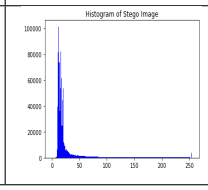
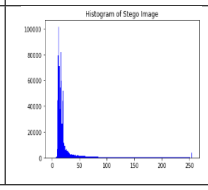
	Raw Image	Stego Image with variable message size		
Message Size		1000	5000	10000
Image2				

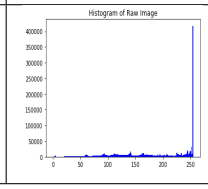
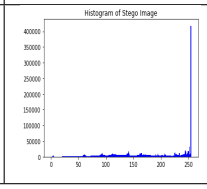
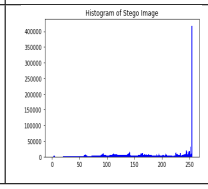
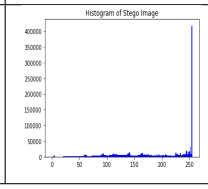
	Raw Image	Stego Image with variable message size		
Message Size		1000	5000	10000
Image3				

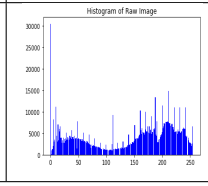
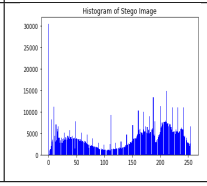
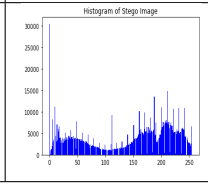
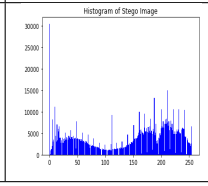
	Raw Image	Stego Image with variable message size		
Message Size		1000	5000	10000
Image4				

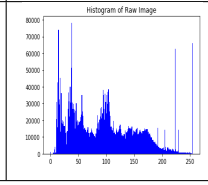
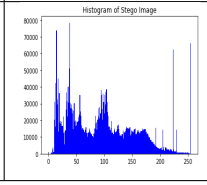
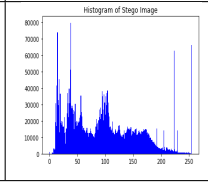
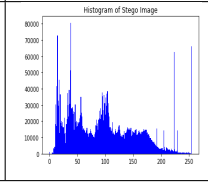
Case study 3

* **Histogram analysis of all cover images and Stego images with variable message size**

	Raw Image	Histogram analysis with variable message size		
Message Size		1000	5000	10000
Image1				

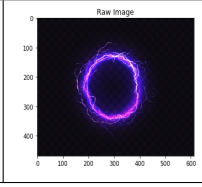
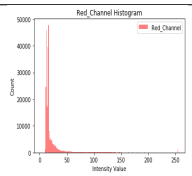
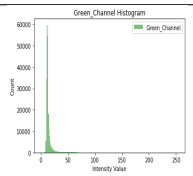
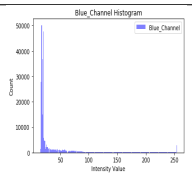
	Raw Image	Histogram analysis with variable message size		
Message Size		1000	5000	10000
Image2				

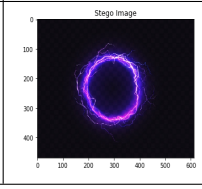
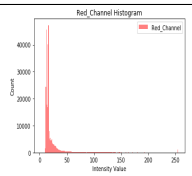
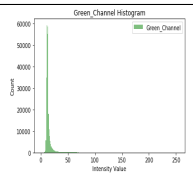
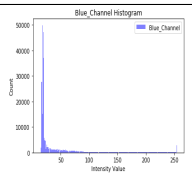
	Raw Image	Histogram analysis with variable message size		
Message Size		1000	5000	10000
Image3				

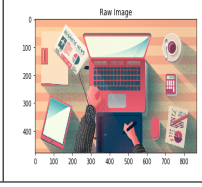
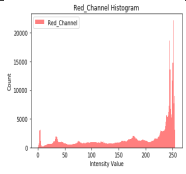
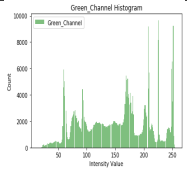
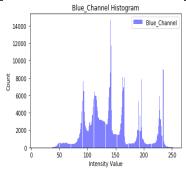
	Raw Image	Histogram analysis with variable message size		
Message Size		1000	5000	10000
Image4				

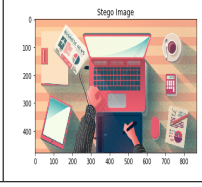
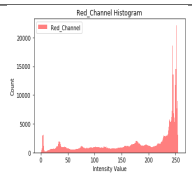
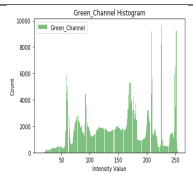
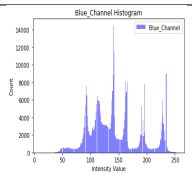
Case study 4

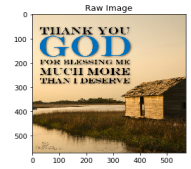
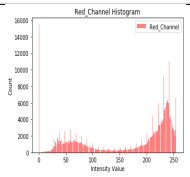
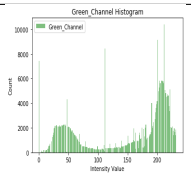
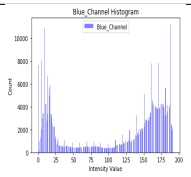
* **Histogram analysis of all cover images and Stego images with composition of Red, Blue & Green colours**

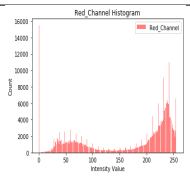
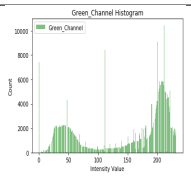
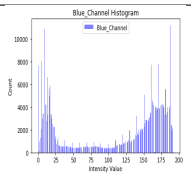
	Raw Image	Histogram analysis with variable message size		
Message Size		Red_Channel	Green_Channel	Blue_Channel
Image1				

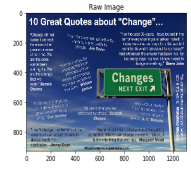
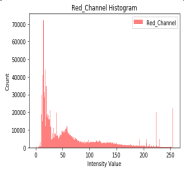
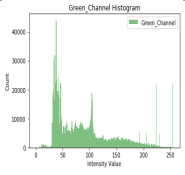
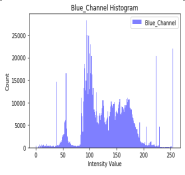
	Stego Image	Histogram analysis with variable message size		
Message Size		Red_Channel	Green_Channel	Blue_Channel
Image1				

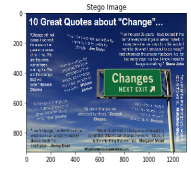
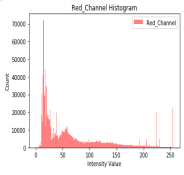
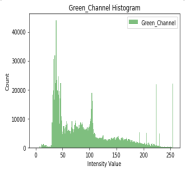
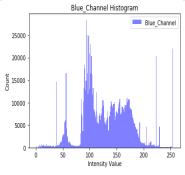
	Raw Image	Histogram analysis with variable message size		
Message Size		Red_Channel	Green_Channel	Blue_Channel
Image2				

	Stego Image	Histogram analysis with variable message size		
Message Size		Red_Channel	Green_Channel	Blue_Channel
Image2				

	Raw Image	Histogram analysis with variable message size		
Message Size		Red_Channel	Green_Channel	Blue_Channel
Image3				

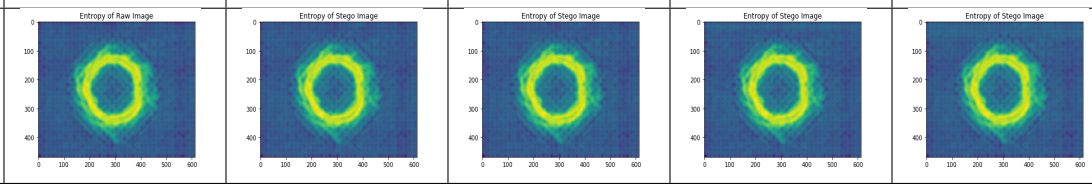
	Stego Image	Histogram analysis with variable message size		
Message Size		Red_Channel	Green_Channel	Blue_Channel
Image3				

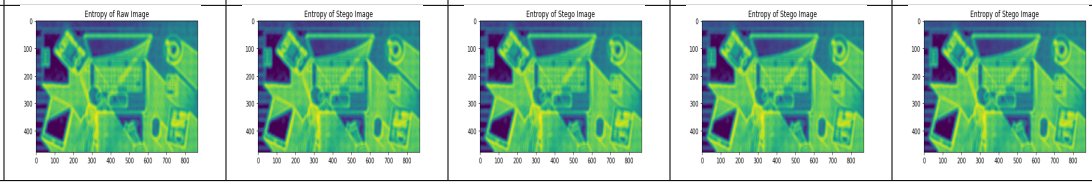
	Raw Image	Histogram analysis with variable message size		
Message Size		Red_Channel	Green_Channel	Blue_Channel
Image4				

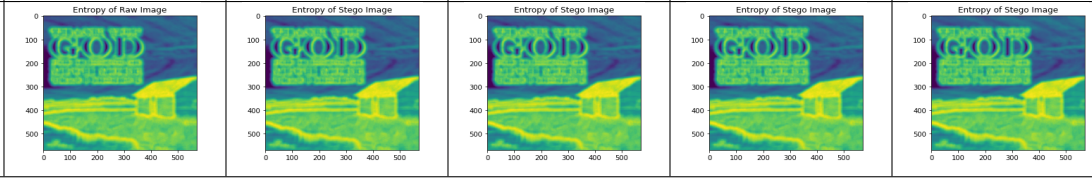
	Stego Image	Histogram analysis with variable message size		
Message Size		Red_Channel	Green_Channel	Blue_Channel
Image34				

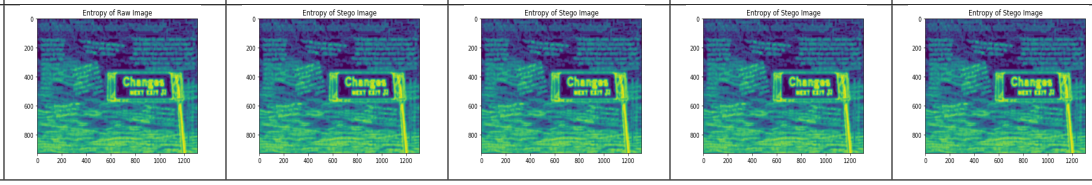
Case study 5

* **Entropy of all Cover images and Stego images with variable message size**

	Raw Image 1	Stego Image with variable message size			
Message Size		500	1000	5000	10000
Image1					

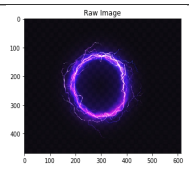
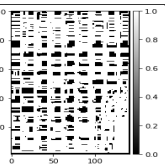
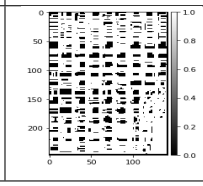
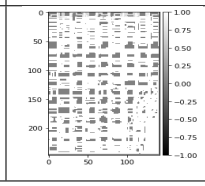
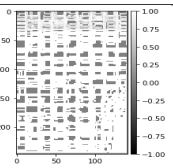
	Raw Image 2	Stego Image with variable message size			
Message Size		500	1000	5000	10000
Image2					

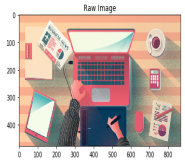
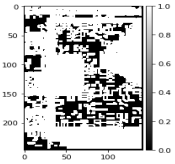
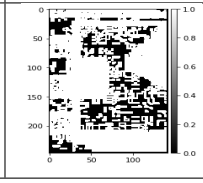
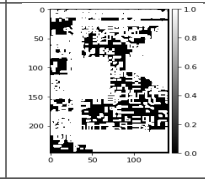
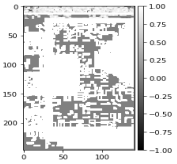
	Raw Image 3	Stego Image with variable message size			
Message Size		500	1000	5000	10000
Image3					

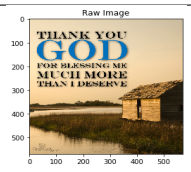
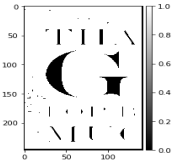
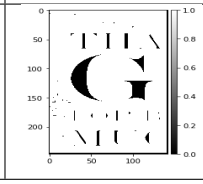
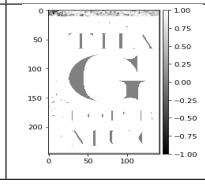
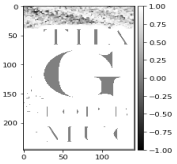
	Raw Image 4	Stego Image with variable message size			
Message Size		500	1000	5000	10000
Image4					

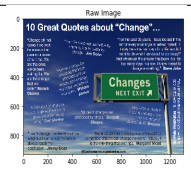
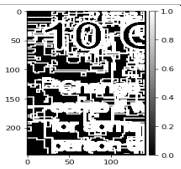
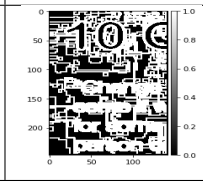
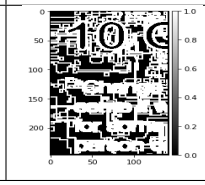
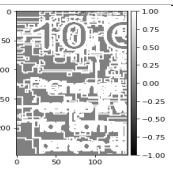
Case Study 6

* **Normalised Cross Correlation of all Cover images and Stego images with variable message size**

	Raw Image 1	Stego Image with variable message size			
Message Size		500	1000	5000	10000
Image1					

	Raw Image 2	Stego Image with variable message size			
Message Size		500	1000	5000	10000
Image2					

	Raw Image 3	Stego Image with variable message size			
Message Size		500	1000	5000	10000
Image3					

	Raw Image 4	Stego Image with variable message size			
Message Size		500	1000	5000	10000
Image4					

Case Study: 7

* **Time required to encrypt and decrypt the data with variable message size**

Message Size	Method	Time required to Encrypt & Decrypt the Data with variable message			
		500	1000	5000	10000
Time Required	Encryption	0.18 sec	0.846 sec	1.451	7.753 sec
	Decryption	0.117 sec	0.179 sec	0.2698	0.422 sec

Discussion:

We have selected 4 different colour images of different size namely raw_image1.png, raw_image2.png, raw_image3.png & raw_image4.png for experiment and have considered different image quality parameter for analysis of our proposed method. Above table (Case Study1) shows image quality parameters with variable message size. We can compare value of different image quality parameters with their best value from table. Raw image with its stego image has shown in above figures (Case Study 4). Figures (Case Study 3) shows histogram analysis of raw image with stego image. Values of image quality parameters are very close to their best value with message size 500, 1000, 5000 and 10000. But values of image quality parameter with message size 10000 is slightly deviated with their best value. From histogram analysis we say that stego images with variable message size are very similar to that of selected raw images. Our proposed method works best with message size less than or equal to 10000.

A method for image Steganography was proposed implemented and tested, it was shown from the obtained results that stego image always has a good quality and has a best value of their image quality parameters, thus it is difficult to guess whether the stego image is differ from the original one.

All LSB methods of hiding message in colour image are simple but they are not highly secure and they need an extra work to add an encryption tool to increase the security of hidden information, thus the proposed method can be used to hide messages within a colour image taking in consideration achieving best performance by providing a high secure of message hiding and extracting within a minimum time of processing.

7 Conclusion and Future Work

As the world of technology keeps developing, we will be depending on these technologies more and more. It is necessary to ensure that these technologies are secure and sound. Users need to worry about how their data is being treated. A simple conversation or message can be captured easily on the web therefore it is going to harm the confidentiality. In this research, a robust and unique technique have been proposed to secure the communication of data with the use of steganography and cryptography combined.

An enhanced version of Blowfish encryption algorithm is used to encrypt the secret messages, with the help of Pixel Indicator technique the encrypted text is embedded inside raw images.

To conclude and evaluate the performance of our proposed technique, A set a of statistical tests were carried out between the raw images and the stego images in order to identify the quality of both the images and figure out if we can extract the cipher text by performing statistical attacks.

- The results of visual test were as follows, the Steg-image and Cover Image looked exactly the same meaning it cannot be detected by a human eye whether which image is Stego image and which is raw image. The receiver was able to extract the encrypted text and decrypt it easily without any data loss which means the decrypter can easily recover the cypher text using the secret key.
- The Histogram analysis of stego image and cover image were intact and no difference was found between the images. Which means we are secure against any histogram steg analysis attacks.
- On running the PSNR and MSE tests our proposed algorithm showed a good behaviour. Even though the proposed method can encrypt any size of text, but it works best with 10,000 or less text size in the in the raw image while it won't affect the quality of steg-Image. The quality of Steg-image remains same without any distortion if the text size is 10,000 characters.
- The quality of result images proposed approach prove that it is superior, fast, and efficient compared to the traditional LSB approach.
- The proposed system successfully covers all the features of CIA Triad methodology.

The proposed system only validates text and it will only embed that into images. However, regarding future work researchers can change file formats to Audio, Video, or even Networking protocols such as TCP, HTTP. I suggest using Networking protocols as a mode of steganography as it is a unique way of embedding the data and it will be interesting to see how it turns out. Overall, a diverse combination of encryption algorithms along with steganographic techniques can be used and tried to provide a confidentiality to the input.

References

[1] A. Toumazis, "Steganography," *University of Cambridge*, 2009.

[2] C. Shashikala and J. Ajay, "Steganography an art of hiding data," *International Journal on Computer Science and Engineering*, vol. 1, 12 2009.

[3] A. Kumar and K. Pooja, "Steganography-a data hiding technique," *International Journal of Computer Applications*, vol. 9, no. 7, pp. 19–23, 2010.

[4] M. Khairullah and M. A. S. Ratul, "Steganography in bengali unicode text," *SUST Journal of Science and Technology*, 2018.

[5] A. Malik, G. Sikka, and H. K. Verma, "A high capacity text steganography scheme based on lzw compression and color coding," *Engineering Science and Technology, an International Journal*, vol. 20, no. 1, pp. 72–79, 2017.

[6] R. Doshi, P. Jain, and L. Gupta, "Steganography and its applications in security," *International Journal of Modern Engineering Research (IJMER)*, vol. 2, no. 6, pp. 4634–4638, 2012.

[7] N. Provos and P. Honeyman, "Hide and seek: An introduction to steganography," *IEEE security & privacy*, vol. 1, no. 3, pp. 32–44, 2003.

[8] M. P. R. Kamble, M. P. S. Waghamode, M. V. S. Gaikwad, and M. G. B. Hogade, "Steganography techniques: A review," *International Journal of Engineering*, vol. 2, no. 10, 2013.

[9] H. Inoue, A. Miyazaki, and T. Katsura, "An image watermarking method based on the wavelet transform," in *Proceedings 1999 International Conference on Image Processing (Cat. 99CH36348)*, vol. 1. IEEE, 1999, pp. 296–300.

[10] K. Raja, K. Kumar, S. Kumar, M. Lakshmi, H. Preeti, K. Venugopal, and L. M. Patnaik, "Genetic algorithm based steganography using wavelets," in *International Conference on Information Systems Security*. Springer, 2007, pp. 51–63.

[11] C.-K. Chan and L.-M. Cheng, "Hiding data in images by simple lsb substitution," *Pattern recognition*, vol. 37, no. 3, pp. 469–474, 2004.

[12] R. El Safy, H. Zayed, and A. El Dessouki, "An adaptive steganographic technique based on integer wavelet transform," in *2009 International Conference on Networking and Media Convergence*. IEEE, 2009, pp. 111–117.

[13] A. M. Fard, M.-R. Akbarzadeh-T, and F. Varasteh-A, "A new genetic algorithm approach for secure jpeg steganography," in *2006 IEEE International Conference on Engineering of Intelligent Systems*. IEEE, 2006, pp. 1–6.

[14] D. R. ElShafie, N. Kharma, and R. Ward, "Parameter optimization of an embedded watermark using a genetic algorithm," in *2008 3rd International Symposium on Communications, Control and Signal Processing*. IEEE, 2008, pp. 1263–1267.

[15] P.-Y. Chen, H.-J. Lin *et al.*, "A dwt based approach for image steganography," *International Journal of Applied Science and Engineering*, vol. 4, no. 3, pp. 275–290, 2006.

[16] A. Gutub, A. Al-Qahtani, and A. Tabakh, "Triple-a: Secure rgb image steganography based on randomization," in *2009 IEEE/ACS International Conference on Computer Systems and Applications*. IEEE, 2009, pp. 400–403.

[17] A. A.-A. Gutub *et al.*, "Pixel indicator technique for rgb image steganography," *Journal of emerging technologies in web intelligence*, vol. 2, no. 1, pp. 56–64, 2010.

[18] M. T. Parvez and A. A.-A. Gutub, "Vibrant color image steganography using channel differences and secret data distribution," *Kuwait J Sci Eng*, vol. 38, no. 1B, pp. 127–142, 2011.

[19] V. Nagaraj, V. Vijayalakshmi, and G. Zayaraz, "Color image steganography based on pixel value modification method using modulus function," *IERI Procedia*, vol. 4, pp. 17–24, 2013.

[20] C. Prema and D. Manimegalai, "Adaptive color image steganography using intra color pixel value differencing," *Aust J Basic Appl Sci*, vol. 8, no. 3, pp. 161–167, 2014.

[21] C.-Y. Yang and W.-F. Wang, "Block-based colour image steganography using smart pixel-adjustment," in *Genetic and Evolutionary Computing*. Springer, 2015, pp. 145–154.

[22] G. Swain, "Adaptive pixel value differencing steganography using both vertical and horizontal edges," *Multimedia Tools and Applications*, vol. 75, no. 21, pp. 13 541–13 556, 2016.

[23] A. Kanhe, G. Aghila, C. Y. S. Kiran, C. H. Ramesh, G. Jadav, and M. G. Raj, "Robust audio steganography based on advanced encryption standards in temporal domain," in *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2015, pp. 1449–1453.

[24] B. Schneier, "Schneier on security: The blowfish encryption algorithm," in *Schneier.com*, 2020, available: https://www.schneier.com/academic/blowfish/. [Accessed: 14- Aug- 2020].

[25] S. Vaudenay, "On the weak keys of blowfish," in *International Workshop on Fast Software Encryption*. Springer, 1996, pp. 27–32.

[26] T. Nie and T. Zhang, "A study of des and blowfish encryption algorithm," in *Tencon 2009-2009 IEEE Region 10 Conference*. IEEE, 2009, pp. 1–4.

[27] J. S, "Symmetric key algorithms: A comparative analysis," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 4, pp. 15 772–15 775, 09 2016.

[28] P. Semwal and M. K. Sharma, "Comparative study of different cryptographic algorithms for data security in cloud computing," in *2017 3rd International Conference on Advances in Computing, Communication & Automation (ICACCA)(Fall)*. IEEE, 2017, pp. 1–7.

[29] S. Prasad and A. K. Pal, "An rgb colour image steganography scheme using overlapping block-based pixel-value differencing," *Royal Society open science*, vol. 4, no. 4, p. 161066, 2017.

[30] G. Manikandan, N. Sairam, and M. Kamarasan, "A new approach for improving data security using iterative blowfish algorithm," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 4, no. 6, pp. 603–607, 2012.

[31] “Working with images in python,” [Accessed: 16- Aug- 2020]. [Online]. Available: https://www.geeksforgeeks.org/working-images-python/

[32] “Image quality metrics- matlab & simulink,” [Accessed: 16- Aug- 2020]. [Online]. Available: https://www.mathworks.com/help/images/image-quality-metrics.html