

Secure Data Transmission using Cryptography, Image
processing and Steganography



Rahul Kishor Sogam

10612997

Supervisor: Prof. Salah Aberkane

This dissertation is submitted for the degree of
MSc in Information system and Computing

August, 2023

Declaration

'I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.'

Signed: Rahul Kishor Sogam

Student No: 10612997

Date: 29th August 2023

Acknowledgment

I am deeply indebted to my supervisor, Salah Aberkane, for their invaluable guidance, unwavering support, and expert advice throughout the entire journey of this research. Their mentorship and encouragement have been instrumental in shaping the direction of this work. My deepest appreciation goes to my family and friends for their continuous encouragement, patience, and understanding during the demanding phases of this thesis. Their belief in my abilities has been a constant source of motivation. I extend my gratitude to Dublin Business School for providing the resources and conducive environment necessary for conducting research. The access to libraries and other sources has greatly enriched my learning experience.

Abstract

In today's digital era data plays a crucial role, users are passing the information from one end to another without ensuring the security of the data. It is paramount to ensure the confidentiality, integrity and security of the data while it is being transmitted over any of the platforms. Many platforms use various types of security algorithms like cryptography, steganography but it is found that is not enough to make the secure data transfer. Attackers are always two steps ahead and find one or another way to violate the data security. Hence this report presents a comprehensive approach that combine cryptography using AES algorithm with QR code generation and least significant bit Steganography to achieve a robust framework and enhanced data security. This enhanced data security methodology addresses the challenges of secure data transmission across digital platforms.

The foundation of proposed methodology lies in the implementation of AES cryptography, which is widely recognized and utilized robust secure cryptography algorithm. AES is employed to transform plain text secret message into cipher text using encryption operation. This encryption ensures the confidentiality and protection from unauthorized access of data. The further implementation of Quick Response code enhances the secure data transmission and hiding data into an image format. This QR codes serve as bridge between digital and physical mediums and enables the sufficient data exchange while maintaining the encryption integrity.

To add one more level of security with AES and QR, LSB steganography is also added. In this layer of security LSB uses least significant bit of cover image pixels data and subtly embed the encrypted data into it. This covert channel not only complements the encryption but also adds the level of obscurity. This makes challenging for unauthorized users to detect the presence of the information. Proposed methodology is evaluated through rigorous testing to assess its security, effectiveness and its efficiency in real world scenarios. The experimental results and demonstrate the successful integration of AES cryptography, QR code operations and LSB steganography. This research contributes to the advancement of secure data transmission techniques.

Table of Contents

Table of figure.....	6
Chapter 1.....	7
Introduction.....	7
1.1 Introduction.....	7
1.2 Importance of Data security.....	8
1.3 Research Problem.....	8
1.4 Research Questions.....	9
1.5 Research Objective.....	9
1.6 Research Roadmap.....	9
Chapter 2.....	11
Literature Review:.....	11
2.1 Different ways of Cryptography.....	11
2.2 Problems with Different Approaches in cryptography.....	12
2.3 QR code generation and protection.....	12
2.4 Problems with Different approaches in QR code encryption.....	13
2.5 Problem's resolution.....	13
2.6 Steganography approaches.....	14
2.7 Problems with Various steganography approaches.....	15
2.8 Solution to the problems.....	16
Chapter 3.....	17
Methodology.....	17
3.1 AES Encryption and Decryption Process:.....	19
3.2 QR Code Generation and Scanning:.....	22
3.3 Steganography Process Using LSB.....	25
Chapter 4.....	29
Implementation and Design of a System:.....	29

4.1 Proposed System.....	30
4.2 Application User Interface.....	33
Chapter 5.....	39
Discussion:	39
Chapter 6.....	40
Conclusion:.....	40
6.1 Summary	40
6.2 Limitation:.....	40
6.3 Future work:	40
References.....	41

Table of figure

Figure 1 Research Roadmap	10
Figure 2 Encryption process	18
Figure 3 Decryption Process	18
Figure 4 Generate OTP	21
Figure 5 Setting Key for AES	21
Figure 6 Message Encryption using AES	22
Figure 7 Message Decryption using AES	22
Figure 8 Google ZXing library	23
Figure 9 Generate QR from cipher	24
Figure 10 QR Scanning and Reading Data.	24
Figure 11 LSB method.....	27
Figure 12 Decode Stego Image.....	28
Figure 13 Sender role and flow using SecureIT platform.....	31
Figure 14 Receiver role and flow using SecureIT platform.....	32
Figure 15 Login Page.....	33
Figure 16 Sign up Page.....	33
Figure 17 Home Page.....	34
Figure 18 Encryption page.....	34
Figure 19 Output of AES encryption	35
Figure 20 Download encoded image.	35
Figure 21 Original Cover Image 22 Stego Encoded Image	35
Figure 23 Decoding image.....	36
Figure 24 Download extracted QR	36
Figure 25 Extracted QR	37
Figure 26 Scanning QR Image.....	37
Figure 27 secret message extracted from QR	38
Figure 28 Decrypted Secret Message.....	38

Chapter 1

Introduction

1.1 Introduction

In today's linked world, where communication has become an essential component of modern communities and enterprises, the need for safe transmission of information has never been more essential. Companies of every kind and size rely largely on the secure flow of private data to keep their operations running, preserve consumer privacy, and secure their intellectual property. However, the growing sophistication of cyber-attacks creates substantial hurdles to the integrity, confidentiality, and availability of transmitted data.

Data transmission via networks creates inherent weaknesses that malevolent actors can exploit. Interception, intercepting, and data manipulation are examples of cyberattacks that have the ability to violate information confidentiality and impair vital systems. As a result, effective security procedures to protect data confidentiality, integrity, and authenticity during transmission is urgently needed. Data security has become a daily worry for the government, consumers, and business owners, resulting in a rising need for data privacy.

Cryptography, one of the most prevalent approaches for achieving data security is encryption and decryption. Encryption transforms plain text into cipher text, whereas decryption transforms cipher text back into plain text [1]. AES encrypts and decrypts the plain text. It is frequently used to protect data. It uses fixed-size data chunks and provides high security. Through a series of complex substitution, permutation, and mixing processes, AES converts input data (plaintext) into distorted output (ciphertext) using a key (128, 192, or 256 bits). The complexity of AES's transformations and reliance on the encryption key contribute to its security. It is utilized in numerous applications, including secure communication and file encryption. Once the Encrypted text is ready QR code is generated out of that text. QR code is basically a two-dimensional code which is generally capable of encoding various types of information such as binary, numeric, alpha-numeric, kanji and control code. It is also capable of encoding the information in both horizontal and vertical ways. QR code technology has proven to be effective even when the code is substantially broken. This is possible because QR codes, which are focused on the Reed-Salomon Codes, have error correction levels [2].

Steganography is considered as a way to conceal the sensitive information using another medium also known as cover medium. These cover objects could be any multimedia files such as image, audio, video however images are widely used as a cover object to hide the

sensitive information [3]. Various algorithms and techniques are available to hide the sensitive information using cover medium such as Least Significant Bit (LSB), Palette-based, spread spectrum, Transform domain, Adaptive steganography etc. These algorithms are mainly used to embed the information in cover medium and extract the same whenever it is needed.

Thus, the motive of this research is to use AES cryptography, QR code generation and steganography. To support this novel idea one platform is designed using which users can securely transfer the data to each other and perform the encoding and decoding of the data at their end with ease. QR code has many benefits of its own such as smaller in size, High precision values and speed of reading the data. The hidden information is described by the QR Code, which is a machine-readable digital data representation. QR codes can be used in various fields because of their reliability, useability and speed. In digitalised world QR codes have become more popular specifically in some applications they are also used in stores and chains for the money transactions, tracking items and much more. Another technique which is implemented to enhance the security is cryptography. Many cryptography algorithms ensures that the data integrity and security is not lost while transferring any type of data. Key is considered as one of the important parts for the encryption and decryption process in cryptography. Cryptography is mainly of 2 types symmetric and asymmetric cryptography. In this research we design a new method to make the secure data transmission with the help of combining these three techniques which plays a crucial role hiding the sensitive information.

1.2 Importance of Data security

Security for data plays an important role in modern digital era because of various reasons. Firstly, it is crucial for protecting personal and confidential information, preventing unauthorized users from accessing others sensitive information and misusing it. Moreover, to avoid legal consequences it is important to adhere strict data protection regulations and rules. Data breaches can affect and harm organization reputation which may lead to financial losses or break the trust of stakeholders. Overall data security is one of the important aspects of modern business.

1.3 Research Problem

With the increasing reliance on online communication and the expansion of various network technologies, guaranteeing safe data transfer has become an urgent challenge. Traditional ways of safeguarding data transmission, such as encryption and authentication, may be ineffective in a heterogeneous network environment with a mix of wired and wireless connections, varied

security protocols, and dynamic network topologies. These traditional ways such as encryption and authentication could be exploited by an attacker using any attacking technique. This is considered as a challenge and a scope to develop new secure way to transmit confidential information from one end to another.

1.4 Research Questions

How secure data transmission methods can be enhanced?

How user can securely transfer secret message?

How cryptography, Image processing and Steganography can be combined together to increase the security levels?

1.5 Research Objective

In order to address these challenges and security concerns this applied research proposes an enhanced security technique combining cryptography, Image processing with QR code and Steganography. Making use of these technologies together under one platform would increase the data security level as well as more difficult for attacker to violate and damage the sensitive information. The primary objectives are: -

1. Use Cryptography to encrypt the user's data.
2. Generate the Quick Response Code of encrypted text.
3. Hide QR code in an image using Steganography.
4. Combine these techniques together to enhance the security majors.
5. Only valid receiver should be able to decode and read the information.

1.6 Research Roadmap

Below attached image of Research roadmap shows all the present chapters and gives overview of each of these chapters.

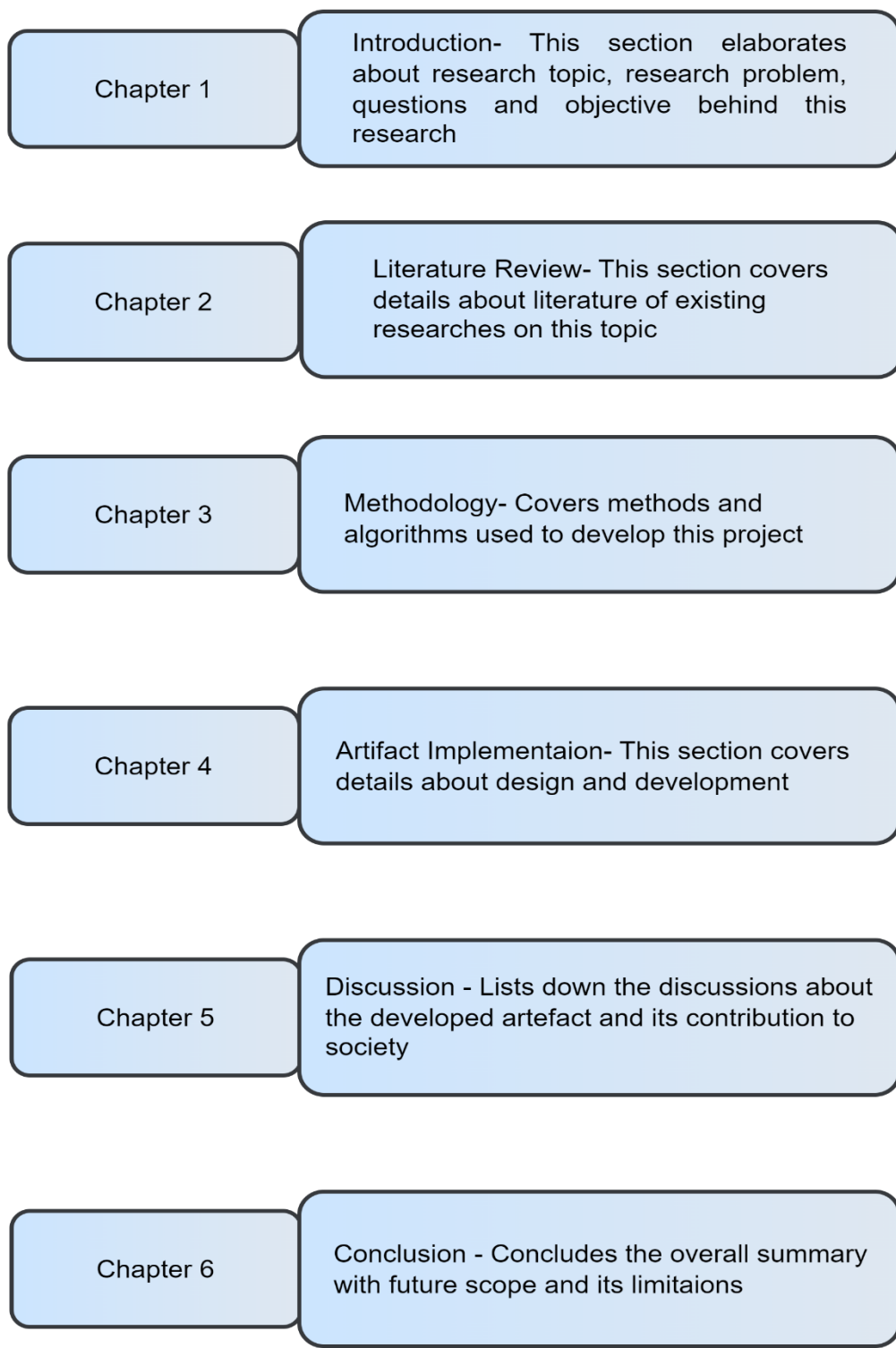


Figure 1 Research Roadmap

Chapter 2

Literature Review:

As technology advances, so does digital communication which eventually leads to increase in information circulation from one end to other. Moreover, most of the time confidential information is transferred using public or open communication platforms which increase the probability of data getting leaked or attacked by unauthorised user.

In past few years, the number of cyber-attacks has increased by high margin and this has always been a major concern for data security experts all over the world. Cryptography and Steganography plays a very important role in this challenge. Combining these two techniques along with QR code would offer high security rather than using all these techniques separately. Recently the range of research papers has increased, which elaborate various cryptographic techniques for encryption and hiding the information in images. Each paper explores various aspects of cryptography and steganography also proposes some innovations to improve the data security.

2.1 Different ways of Cryptography

Cryptography is one of the methods which is considered to be secured and safe to pass on the data from one end to other. Main mechanism of cryptography is that it encrypts the data into unknown format and makes it difficult to understand to any other user, while uses the same mechanism to decrypt the text and turn it into readable format [2]. In [9] author proposed evaluated the performance of various encryption algorithms based on symmetric functionalities to find the effectiveness of algorithms. Cryptography employs a variety of methods and algorithms to provide secure data transfer and the protection of sensitive information. There are various cryptographic techniques such as symmetric and asymmetric cryptography, hash function, digital signature, steganography etc. Every form has its own set of advantages and disadvantages.

In [1] author Musa. M and Aminat introduced a method using Advanced Encryption Standard (AES) with Steganography combined together which improves the security for data transfer. Plain text is first encrypted using AES algorithm. This algorithm basically divided into four phases and each phase plays avital role for the encryption. Encrypted text is then passed to steganography process where least significant bit algorithm is applied to hide the text into cover medium. Author A. Gambhir and A.R. Mishra in [10] proposed a new way to use RSA algorithm. They introduced a new way where firstly the encrypted text is created by changing

the hidden text, and the encrypted text is then concealed in audio cover with the help of LSB for audio steganography. Sound material is first received by the receiver, which then decrypts it into text. Overall, this methodology uses encryption and steganography for security purposes. In [11] author suggested approach a new visual cryptographic technique. Bitmap and grayscale images can both benefit from this method. The idea behind this methodology is to create shares and utilize an algorithm for share stacking based on the residual number system (RNS). basis of a Chinese Remainder Theorem. The Share Creation algorithm separates the secret message into a multiple share. The shares produced by this algorithm are unintelligible format which makes difficult to view the hidden picture. Single piece of share can not the entire hidden message. The Share stacking algorithm helps to reveal the hidden picture by choosing the number of shares as a data. Decoding is done by joining the shares. Whereas the keys are generated using the Mixed Key Generation. Researchers in [12] introduced a new way of encryption using 128 key RSA algorithm to encrypt the secret information prior to integration of message into a cover medium and then gradually inserting the encrypted information into a cover picture using F5 steganography algorithm. Authors used DCT method to integrate the hidden message using F5 algorithm.

2.2 Problems with Different Approaches in cryptography.

Three major disadvantages could be identified of private key cryptography. First of all, the secure channel is required to for both the end users to select the key and then transport the key. Secondly, users communicating using private key cryptography. Users select their secret unique key, which basically add up to multiple keys if many users are using the same cryptography. Lastly, the private key cryptography does not ensure the authentication on an open channel. Public key cryptography is considered time consuming and slow as compared to private key cryptography.

2.3 QR code generation and protection.

Image Processing is a field that studies, manipulates and improve digital images. It entails transforming images using various algorithms and approaches to increase their quality, extract relevant information, and make them fit for specific purposes. Image processing is widely used in a variety of industries such as photography, medicine, computer vision, robots, and others. QR code generation is also closely related to image processing because QR code are mainly 2D barcodes which are visualised in a square image format. QR codes are designed with the help of various image processing techniques which consist of encoding data, generating the pattern and then rendering the QR code as an image.

In [13] authors are using a QR code encoder which uses an algorithm to encrypt the data into an QR code format. Using this encryption algorithm ensures the data is encoded into QR format and cannot be easily accessed or understood by any human. Next author in [15] proposed a way for secretly sharing data in form of QR image. They designed a system which can be used to securely transfer data which also generates a QR code in a system. Moreover, the essential idea of the proposed methodology was that it splits the information into N number of shadows. It is not possible to decode the information by itself. Their methodology uses Shamir's secret scheme. In this scheme the data is divided into shares of shadows with the secret sharing method. Anyone cannot get access to the information until the defined threshold of shares is achieved.

Another author in [15] suggested the new approach known as SQR which basically contains the information in an encrypted format with the AES algorithm using a 128-bit key. They made sure that the provided approach delivers the anticipated result following testing by looking up the QR additionally, if the code is flawed or includes any sensitive information, it will now proceed with analysis process to avoid any malicious attack.

2.4 Problems with Different approaches in QR code encryption.

The keys added to the QR code throughout the encoding and decoding processes, as well as the dependability of the key, are some of the issues that have been found. One more issue with key is that it consumes the part of data space which gives a smaller space to store information. One of problem identified in [15] methodology is that their method consumes more time while creation process and scanning process of the QR code.

2.5 Problem's resolution

Using cipher block chaining is one way to reduce time consumption. To ensure that the message is not altered or modified by any attacker, AES can be used. Scanner the most widely used tool however it cannot detect if phishing attacks. A software needs to be built which would help to represent users decision-making process on URL's confidentiality. QR code generator and QR code scanner technology should be built in way that it adds the encrypted data into QR code and identify possible threat. Digital signature and hashing make difficult for the attacker to damage or access the data in QR code. Overall aim is to enhance the security of the QR and ensure that it can only be accessed by authenticated user.

2.6 Steganography approaches.

Authors Musa and Aminat in [1] proposed a method for image steganography using Least significant bit algorithm. All the information first encrypted using AES algorithm. Then this encrypted text is compressed and converted into binary code. This binary conversion is performed with ASCII equivalent values of the character. Finally, the least significant bit from each pixel of cover medium is identified to embed the bits of secret message into it. Same process is repeated until all the message is embedded into cover. In [3] authors introduced a new method using DNA sequence with Hyper-elliptical cryptography along with steganography. This method offers a good security for communication. Method hides a hidden image in some other cover image by transforming nucleotide into a binary interpreter. The complete encoding process is divided into 3 steps firstly, the pixel value of cover image and hidden image is transformed into DNA 3-fold value. Later the 3-fold values are converted into a binary code and then finally using the XOR method new stego-image is produced with binary values of both cover and hidden image.

Author Krishna and Ramakalavathi in [6] proposed a method to securely transfer the information using cryptography and steganography. In this methodology random sequence of pixels is generated by passing a secret key as a seed. These pixels are then used to store an information. Authors mentioned that embedding the message bits will make noise to avoid they used an image whose pixels already has noise. These will be helpful as all the pixels have noise hence it will difficult for an attacker to identify the pixels containing information. The method is designed in such a way that a recipient with correct login credentials would get access to image.

This work [16] represents an approach of text steganography which uses null spaces. In this approach the blank spaces of cover medium are set in text format if the binary bit of coded data is 1 and blank space remains as it is when the binary code of coded data is 0. This approach is mainly divided into 2 phases. First phase is concealing where information is concealed in wording and second phase is extraction where data is extracted from the stego-text. Authors in [17] has explored 3 different methodologies for the steganography. First method is least significant bit method which is implemented after an encryption of a text. Header is added to the encrypted text prior concealing it. The modified text consists of a key and extension file. Authors encoded the information before hiding it into LSB. Another approach implemented was Pseudo-Random Number Generator (PSNR), it creates PSNR using random strings 1 and 0. To randomly generate a sequence authors used Blum Blum Shub (BBS). The last approach

which they implemented is scattered LSB which basically breaks BMP file into multiple blocks. These blocks are used to store the information using LSB based on PRNS.

In [18], to provide security services that confidentially authors implemented Secure information hiding system (SIHS). Main goal of the system is to hide the message using SHIS and apply the LSB algorithm. In this approach they use an image of 800 x 600 pixels size which can hold information of 60Kb. Later they embedded this picture of containing the message into cover picture. The Stego-image and cover image appeared to be same and makes a minimum change in a cover image which results into an unnoticeable change for human eyes. As a result, they discovered that the message's size is less than the cover images. A large capacity decreases the bandwidth required to transmit the steganography image by allowing the use of a smaller cover-image for a communication message with a fixed size.

In order to provide safe, secure, and very difficult to decode interactions of sensitive information, this work [19] proposes a technique that utilizes extensively encoded algorithms and steganographic approaches. Before being encoded in a QR code, the concealed letter is first encrypted using a cryptography technique. They have converted the message into base 64 layouts using AES-128 encoding in order to facilitate further research. The encrypted image is scrambled to achieve a higher level of security. Eventually, a suitable LSB of the cover image include a scrambled QR code. To create steganography for digital images, they adopted an LSB strategy. The recipient side decoding process gathers the sensitive data. For the exchange of a confidential message, a four-level security framework is offered. It is designed using 4 different security levels. First, the message is encrypted using AES algorithm and then this encrypted message is converted in base64. Second, this base 64 encoded message is merged into a QR code. Third, random scramble order is generated in this level and this order is stored with obtained RGB values. After this concatenation picture is generated from it. Lastly the Least significant bit insertion is used to generate a digital image steganography.

2.7 Problems with Various steganography approaches

If some of the LSB bits are modified in stego-image the message embedded in cover image could be lost and cannot be recovered. If any alterations are performed on stego-image like cropping, scaling, rotating the image then message might get affected and could not be retrieved successfully. Any stego-image can be easily violated by filtering or manipulation. Attacker can alter the text message by discarding the LSB plane. In [17] the entire methodology works on the basis of Bitmap file. If any image processing method is applied on stego image

then retrieving an information or message would be difficult task also if stego key is known to attacked it would be very easy to crack the information. The biggest flaw [16] is that it takes up a lot of storage space for encryption of a small number of bits. For example, a single character is equivalent to 8 bits and needs around 8 inter-spaces to be encrypted. Additionally, some difficulties are brought on by the cover message's limited storage. The component size of the stego-cover image increases as a result of the addition of more gaps to show data. Only a text file can be used to store a secret message, and a modification in a single bit can change the ASCII code.

2.8 Solution to the problems

A solution to the blank pixel was put out by one of the writers. He followed the steps so that the final pixel in a block of k (Stego-key) and m (Message) pixels equals $k + m - 1$. An array for identifying the empty pixel in the most recent block. This approach employs a block to cover the information in itself.

Chapter 3

Methodology

Due to the proliferation of contemporary communication methods, the preservation of data during transmission has become a pressing concern. It is crucial that data exchanged between parties be protected against unauthorized access or alteration. The primary goal of secure data transmission is to protect data from unauthorized access, tampering or eavesdropping during its transit from one user to another. Secure data transmission can be achieved by keeping the confidentiality, integrity and authenticity of the data. To protect, data confidentiality, integrity and authenticity techniques like cryptography, steganography etc. can be applied [4].

To protect the information from unauthorized access a certain method is designed and applied on a data. This section gives detailed understanding of the algorithm designed and in-depth knowledge of the mechanisms used to implement the same. The proposed methodology works on 3 important concepts which altogether helps to improve the data security. The concepts include Cryptography, Image processing in which QR code is generated and Steganography. This method is mainly divided in 3 different layers where each layer plays its important role and provides extra security. To support the same methodology and to perform encoding and decoding operations using this methodology one platform is designed using Java with Spring and Hibernate framework along with JavaScript, CSS, Bootstrap for frontend.

First Layer is Symmetric cryptography algorithm. This layer uses Advanced Encryption Standard algorithm also known as AES to encrypt the message and convert the plain text message into cipher text which cannot be understood without decrypting. Output of this layer is given as an input to the next layer. Second layer is QR Code generation which is considered as a part of image processing. In second layer Quick Response code also known as QR code is generated of the encrypted text. QR code can hold the information in an image format and cannot be understood by humans unless images are scanned by scanner or a device. This QR code holds encrypted message which is then passed to the next phase. The third and final layer of security is steganography which is very important in this methodology. Using steganography QR code is embedded into a cover image of a sender's choice. Least Significant Bit algorithm of steganography is used to embed the QR into cover image. To make sure only authorised user can read and decode the image the OTP along with stego image is sent to a receiver's mail id which can be accessed by only intended receiver and the same OTP can be used as key to decrypt the cipher text.

Below attached figures 1 and 2 represent the flow of the encoding and decoding algorithm respectively.

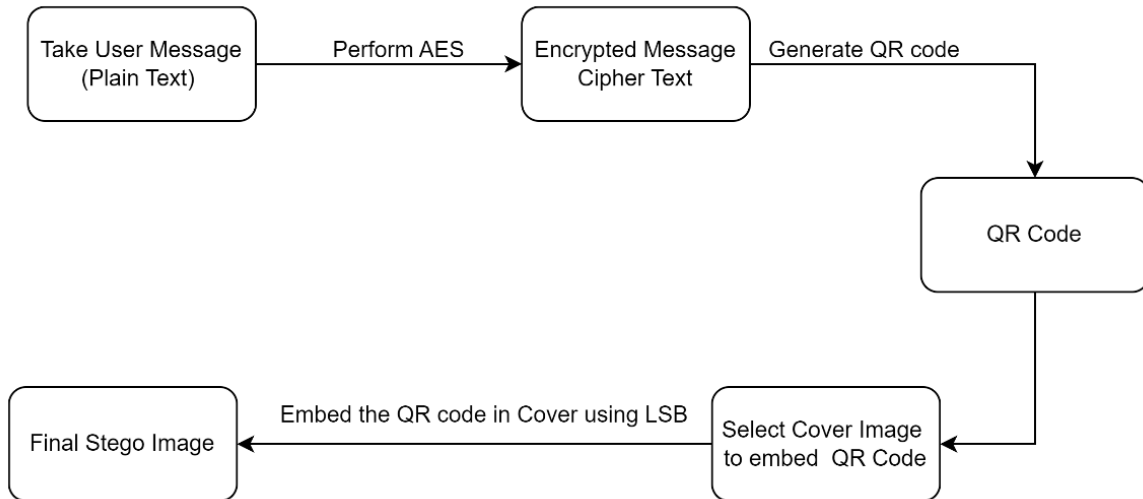


Figure 2 Encryption process

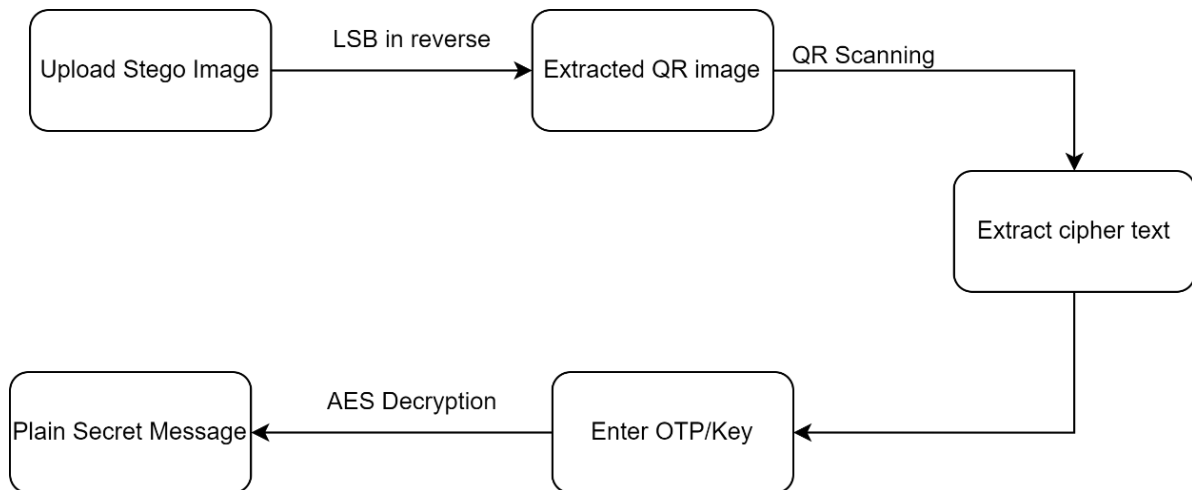


Figure 3 Decryption Process

Let us understand each Phase in detail

3.1 AES Encryption and Decryption Process:

This sub-section elaborates about the complete working of the Advanced Encryption Standard algorithm. As proposed to use the SHA256 algorithm for the cryptography process however, AES is being implemented in the methodology the reason for this change is, Hashing is mainly used for creating a fixed size hash values of the given text and then same hashing method is used to verify the data integrity. Main purpose of Hashing is to store user credentials in database by changing the user entered password into hash value and same algorithm is used for validation and comparison of user entered and stored data when user perform login operation.. In this project as a part of security measures no such data or information related to message is being stored in database to validate hence AES is being used for Encryption and Decryption. while properly implemented and using strong keys, AES offers a high level of security. It is frequently used to protect sensitive data, such as in file encryption and SSL/TLS for secure web communication. AES is also known as substitution -permutation network cipher (SPN) which generally perform multiple rounds and each round consists of multiple stages of operation. Let's see each of these rounds in details

3.1.1 Encryption Rounds:

1. Key Expansions:

AES supports 128,192 or 256 bits of key size and each round of encryption and decryption uses a set of round keys that are enlarged from the initial key. A key schedule method is used in the process of key expansion to generate subkeys for each round.

2. Initialization Round:

In this step the XOR operation is performed on the plain text combining with first round key. This step serves as initializer for each round and ensures that process of encryption always begins with combining plain text and Key.

3. Main Encryption Loop:

The Encryption Loop generally depends on the size of the key, If the key size is 128 bits, then it performs 10 rounds, for key size 192 performs 12 rounds and 14 rounds for key size 256 bits. These rounds are divided into 4 operations which are following:

- Sub Bytes:

Byte-level substitution with the aid of an S-box. The relevant value from the S-box is substituted for every byte in the state matrix.

- **Shift Rows:**
Diffusion is created by shifting rows of the current state matrix to the left. The first row is left unaltered, followed by a one-byte shift in the second row, a two-byte shift in the third row, and a three-byte shift in the fourth row.
- **Mix Columns:**
The state matrix's individual columns are blended via a matrix multiplication process. Diffusion is created, and the cryptographic strength is increased.
- **Add Round Key:**
In this stage state matrix is XORed with the round key which is assigned for the current round.

4. Final Round

In the final round of encryption, the mixed columns are removed which helps to simplify the decryption process.

3.1.2 Decryption Process

AES decryption process is somewhat similar to the encryption however the operations are performed in a reversed manner.

Decryption process mainly involves following stages:

- **Adding Round Key:**
In this stage the XOR operation is performed on the cipher text with the last round key.
- **Inverse Shift Rows:**
In this stage the rows which were shifted to the left side during encryption process are shifted back in opposite direction to reverse the encryption effect.
- **Inverse Sub Bytes:**
Each byte of state matrix was replaced by a corresponding byte from the s-box during encryption process same process is reversed in this stage the bytes of s-box are replaced with corresponding value.
- **Final Round of Decryption:**
The first "Add Round Key" operation, which was applied to the plaintext during encryption, is reversed in this stage. The result of the "Add Round Key" operation is the original plaintext, which has been successfully decrypted, represented in the state matrix.

Key always play important role whether it is a symmetric or asymmetric cryptography. Before implementing the AES, cryptography key is being generated using a method called generateOTP shown in figure 3. This method uses java math class to generate a 4-digit random number for each round of encryption. This 4-digit number acts as a key and it is then given to encryption and decryption process of AES.

```
public static String generateOTP()
{
    Random random = new Random();
    int ranNum = random.nextInt(9000)+1000;
    String num = Integer.toString(ranNum);
    return num;
}
```

Figure 4 Generate OTP

Now that key is generated, before passing the 4-digit number as key to encryption and decryption method it is given to SetKey method shown in figure 4 where configuration is performed on the key. Firstly, it is converted to bytes using the UTF-8 encoding process which is commonly used to ensure the consistency across various platforms. After the conversion md.digest method is called to calculate the hash value of the key and the result of the hash value is truncated to use only six bytes which ensures that the key is of the appropriate length for use. Below image shows the same.

```
public static void setkey(final String myKey)
{
    MessageDigest md = null;
    try {
        messageInBytes = myKey.getBytes("UTF-8");
        md = MessageDigest.getInstance("SHA-1");
        messageInBytes = md.digest(messageInBytes);
        messageInBytes = Arrays.copyOf(messageInBytes, 16);
        secrets = new SecretKeySpec(messageInBytes, "AES");
    } catch (NoSuchAlgorithmException | UnsupportedEncodingException e) {
        e.printStackTrace();
    }
}
```

Figure 5 Setting Key for AES

After the configuration same key is ready to use for encrypting and decrypting operation. To perform the encryption or decryption process Configured key is passed along with the text which can be plain text for encryption and cipher text for decryption method. Then AES performs various rounds explained in 3.1.1 and 3.12. Below images show the encryption and decryption method respectively.

```

public static String encryptMessage(final String message, final String otp) {
    try {
        setOtp(otp);
        Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
        cipher.init(Cipher.ENCRYPT_MODE, secrets);
        return Base64.getEncoder()
            .encodeToString(cipher.doFinal(message.getBytes("UTF-8")));
    } catch (Exception e) {
        System.out.println("Error while encrypting: " + e.toString());
    }
    return null;
}

```

Figure 6 Message Encryption using AES

```

public static String decryptMessage(final String encryptedmsg, final String otp) {
    try {
        setOtp(otp);
        Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5PADDING");
        cipher.init(Cipher.DECRYPT_MODE, secrets);
        return new String(cipher.doFinal(Base64.getDecoder()
            .decode(encryptedmsg)));
    } catch (Exception e) {
        System.out.println("Error while decrypting: " + e.toString());
    }
    return null;
}

```

Figure 7 Message Decryption using AES

3.2 QR Code Generation and Scanning:

Image Processing is a field that studies, manipulates and improve digital images. It entails transforming images using various algorithms and approaches to increase their quality, extract relevant information, and make them fit for specific purposes. Image processing is widely used in a variety of industries such as photography, medicine, computer vision, robots, and others. QR code generation is also closely related to image processing because QR code are mainly 2D barcodes which are visualised in a square image format. QR codes are designed with the help of various image processing techniques which consist of encoding data, generating the pattern and then rendering the QR code as an image.

QR code generally consists of black and white pattern on a two-dimensional geometric planer surface. Black pattern represents the binary number 1 while binary number 0 is used to represent white pattern. This code can be rotated 360 degrees. It also has 3 finding patterns

which are at corner. QR Code also has error correction capability which allows to restore data even when substantial parts of the code are damaged.

Error correcting in QR Code [7].

QR Code has a function of an error correcting for miss reading that white is black. Error correcting is defined in 4 levels as below.

1. level L: about 7% or less errors can be corrected.
2. level M: about 15% or less errors can be corrected.
3. level Q: about 25% or less errors can be corrected.
4. level H: about 30% or less errors can be corrected.

One of the reasons of using Java at the backend is that it supports multiple libraries which helps to implement and optimize the code. QR code generation and scanning is being implemented with the help of Google's ZXing library. ZXing library of Google is a widely used and open-source library for working with barcodes and QR codes. Below figure shows the various classes used from the ZXing library to support the functionality.

```
import com.google.zxing.BarcodeFormat;
import com.google.zxing.EncodeHintType;
import com.google.zxing.WriterException;
import com.google.zxing.common.BitMatrix;
import com.google.zxing.qrcode.QRCodeWriter;
import com.google.zxing.qrcode.decoder.ErrorCorrectionLevel;
```

Figure 8 Google ZXing library

3.2.1 QR Generation Process

Once the secret message is converted to cipher text methodology enters into second phase of security. To generate a QR code a method name generateQRcode is developed. This method takes cipher text as an input. Once the cipher text is received it starts with setting the dimensions of QR code that will be generated and error correction level is configured. Instance of a Multiformatwriter class from ZXing library is created. This class simplifies the task of generating QR code. Cipher text along with configurations are passed to this instance which

return the data of QR image in bit matrix format. This QR image bit matrix data is used to hide it in the cover image.

```
//to generate QR code of the encrypted text
@SuppressWarnings("unchecked")
public BitMatrix generateQRCode(String text) throws WriterException {

    int width = 200;
    int height = 200;

    @SuppressWarnings("rawtypes")
    Hashtable hintMap = new Hashtable();
    hintMap.put(EncodeHintType.ERROR_CORRECTION, ErrorCorrectionLevel.L);

    MultiFormatWriter qrCodeWrite = new MultiFormatWriter();
    BitMatrix qrCodeMatrix = qrCodeWrite.encode(text, BarcodeFormat.QR_CODE, width, height, hintMap);

    return qrCodeMatrix;
}
```

Figure 9 Generate QR from cipher

3.2.2 QR Code Scanning:

QR code scanning is performed once the QR code is extracted from the Cover image. To scan the extracted QR image and get the cipher text from it, method named decodeQRcode is developed. QR code image is pass as an input to this method and with the help of zxing library task is being performed. In a process Luminance Source is class from Zxing library which is used to abstract the images. This luminance sources are then converted to a binary bit map format. Finally, this bitmap is passed to a Multi format reader class where the text is extracted and stored in a Result. Below figure 9 shows the same implementation.

```
private static String decodeQRCode(File qrCodeimage) throws IOException {
    BufferedImage bufferedImage = ImageIO.read(qrCodeimage);
    LuminanceSource source = new BufferedImageLuminanceSource(bufferedImage);
    BinaryBitmap bitmap = new BinaryBitmap(new HybridBinarizer(source));

    try {
        Result result = new MultiFormatReader().decode(bitmap);
        return result.getText();
    } catch (NotFoundException e) {
        System.out.println("There is no QR code in the image");
        return null;
    }
}
```

Figure 10 QR Scanning and Reading Data.

3.3 Steganography Process Using LSB

Steganography is an additional and important method for enhancing the security of the data transmission procedure [8]. The basic idea of steganography is to accomplish covert communication, in which the existence of the confidential information is concealed from anyone who is not supposed to receive it. The purpose of steganography is to make it difficult for unauthorized individuals to access the concealed message by concealing its existence.

This method mainly takes benefit of the fact that micro or small changes in the pixels least significant bit are rarely visible to the human vision. In Digital images pixels are mainly represented or consists set of colour channels like Red, Green and Blue in RGB image. Each colours intensity is presented using certain number of bits. LSB is the rightmost bit of the binary representation of colour values.

The fundamental principle of LSB steganography is to swap out the LSB of the cover image's pixel values with the bits of the hidden message. This alteration typically goes undetected since the LSB has the least impact on the image's aesthetic appearance. Let's see how the encoding and decoding process works with LSB in steganography along with its code implementation.

3.3.1 Embedding of the data into cover Image using LSB

Hiding the data into cover image consists of multiple steps and each step is very important as per security concerns. These steps are as following:

1. Preparing the Message:

Generally, the message which is to be embedded into Cover image is in text or binary format. So, each bit of the message needs to be represented as equivalent to binary.

2. Pixel Selection:

In this step the sequence of pixels is chosen from the cover image where the data will be hidden. The selected pixel should be big enough to accommodate the complete message.

3. Embedding:

The embedding procedure substitutes bits from the secret message for the least significant bits of each pixel's colour channel values (Red, Green, and Blue) for each frame in the chosen sequence.

For example, let's suppose if the least significant bit of pixel's colour values is (130,199,50) for RGB respectively and the message bit is 10000101 so it changes the value as follows

130 in binary is 10000010 so adding the message bit to its LSB will give 10000011 which makes small change from 130 to 131.

199 in binary is 11000111 so adding the message bit to its LSB will give 11000110 which is 198 in decimal.

Same with 50 which is 00110010 in binary and adding message bit will produce 00110011 that is 51 in decimal.

Same process of modifying the least bit of the pixel is repeated for each bit in the secret message sequence and entire message is accommodated in a cover image.

4. End marking:

To ensure decoding process fully smooth and quick end market could be added to the encoded data which helps to identify the message is complete.

In the implementation of the steganography with LSB algorithm the method named `hideQRCodeInImage` is developed which takes the cover image and QR code generated in previous phase as an input and performs the encoding process on the same. Firstly, the cover image is converted into a buffered image format. Converting the Image file into a Buffered Image helps to perform the operation on specific pixel values of an image data. QR code data in bit matrix format. First loop in this method first iterates through each pixel of a cover image and other loop iterates through height and width of a cover image. After that right shift operation is done on 32-bit cover pixel to move the alpha component to least significant byte. Bitwise AND operation is performed to with "0xFF" to extract the least significant byte. Corresponding bit from QR code bit matrix is retrieved and checked if the bit value is 1 then LSB of alpha is set to 1 if bit is 0 then alpha is set to 0. This process hides the QR data into cover image.

```

//hide QR code in image
public boolean hideQRCodeInImage(BitMatrix qrCodeMatrix,String filepath, String file, String newname ) throws IOException {
    String file_name = filepath+"\\ "+file;
    File file2 = new File(file_name);
    BufferedImage coverImage = ImageIO.read(file2);

    int width = coverImage.getWidth();
    int height = coverImage.getHeight();

    BufferedImage stegoImage = new BufferedImage(width, height, BufferedImage.TYPE_INT_ARGB);

    for (int y = 0; y < height; y++) {
        for (int x = 0; x < width; x++) {
            int coverPixel = coverImage.getRGB(x, y);
            int alpha = (coverPixel >> 24) & 0xFF;

            if (y < qrCodeMatrix.getHeight() && x < qrCodeMatrix.getWidth()) {
                int bit = qrCodeMatrix.get(x, y) ? 1 : 0;
                int stegoPixel = hideBitInPixel(coverPixel, bit);
                stegoImage.setRGB(x, y, stegoPixel);
            } else {
                stegoImage.setRGB(x, y, coverPixel);
            }
        }
    }

    return setImage(stegoImage,new File(filepath+"/"+newname+".png"),"png");
}

```

Figure 11 LSB method

3.3.2 Extracting Data from cover image using LSB

Once the data is embedded into a Cover image, to extract the hidden data same methodology is applied but in a reverse manner and this methodology consists of different steps which are as follows.

Choose Pixels: In this step the pixels of image which was modified during the embedding process are identified. Which helps to locate the part of an image where data is stored.

Retrieve LSB: Once the pixels are selected in a sequence then Least significant bit of each colour from each pixel is extracted and then the last bit of each colour is isolated. Once all the last bits are extracted then they are rearranged in such a manner to reconstruct the hidden binary data.

Termination Marker: During the encoding process termination marker or end marker was added to signal the end of the secret message. This step involves detecting this end marker once the marker is found it indicates that entire data is extracted successfully.

Reconstruction: Once all the bits are collected and end marker is detected this step starts processing. In this the sequence of bits is grouped into bytes and then these bytes are reconstructed into original data format

```
public boolean extractQRCodeFromImage(String filepath, String filename) throws IOException {
    String QRfilepath="D:\\secureIT\\QRImage\\";
    String file_name = filepath+"\\"+filename+".png";
    System.out.println(file_name);
    File file = new File(file_name);
    BufferedImage stegoImage = ImageIO.read(file);

    int width = stegoImage.getWidth();
    int height = stegoImage.getHeight();

    BufferedImage qrImage = new BufferedImage(width, height, BufferedImage.TYPE_INT_ARGB);

    for (int y = 0; y < height; y++) {
        for (int x = 0; x < width; x++) {
            int stegoPixel = stegoImage.getRGB(x, y);
            int lsb = (stegoPixel >> 24) & 1;

            if (x < qrImage.getWidth() && y < qrImage.getHeight()) {
                int qrPixel = (lsb == 1) ? 0xFF000000 : 0xFFFFFFFF;
                qrImage.setRGB(x, y, qrPixel);
            }
        }
    }
    return setImage(qrImage,new File(QRfilepath+"/"+filename+"_QR.png"),"png");
}
```

Figure 12 Decode Stego Image

Extraction of the QR code process happens at the receiver's end. To extract the QR code image from Steganographic encoded image method named `extractQRCodeFromImage` shown in figure 11 is developed. This method takes stego image as an input. Extracting the hidden data is complete reverse implementation of the LSB on cover image.

In this process outer loop iterates through the image's height (y), and the inner loop iterates through the width (x). Least significant bit of current pixel is extracted using bit manipulation. Variable `lsb` holds either 1 or 0 value which represents the hidden QR bit. If the `lsb` is 1, then pixel is set to black (0xFF000000); if the `LSB` is 0, pixel is set to white (0xFFFFFFFF). Once all pixels are processed the resulting QR image is saved.

Chapter 4

Implementation and Design of a System:

Web applications now a days are considered to be a necessity in every organization to implement the solution for proposed cause or algorithm. To design and develop such web applications there are various such programming languages like java, python, JavaScript are available in a market. To support the working of the proposed methodology and implement the secure algorithm for the sender and receiver, one platform is designed named as SecureIT. SecureIT is a platform which offers the all the functionalities like encoding of the data in sender's selected cover image, decoding of the same cover image at the receiver's end with scanning of the QR code which extracts the final secret message for receiver. However, to access all these functionalities user needs to be registered first in SecureIT portal if user is not registered, they can always sign up by providing the required details.

Tech stack for designing and development of this secure IT portal includes Java with Spring and Hibernate at the backend where all the important methods and operations are performed. Java script, CSS and bootstrap are being used at the frontend where users can actually interact with the system and use all the functionalities. Lastly to store the user related details MySQL database is used.

To run the final code locally on server Apache tomcat server is configured. Java is a high-level programming language which is widely used for software developments because of its characteristic of "write once run anywhere". This characteristic makes java platform independent. Moreover, using java with frameworks like spring and hibernate offers a strong combination that makes use of each technology's advantage. This combination enhances the development process. Spring provides a comprehensive framework that simplifies application development by offering various modules for dependency injection, aspect-oriented programming, and transaction management. This results in cleaner and more modular code, reducing the complexity of application logic. Hibernate is a powerful object-oriented relational mapping framework which easily integrates with spring.

Use of hibernate with spring eliminates the need of creating manual SQL queries to interact with database. Hibernate offers high level mapping of java objects and relational database tables. MySQL is one of the open-source relational database management systems (RDBMS) which is widely used for storing, managing and retrieving stored data. MySQL uses client

server architecture where client side interacts with server to perform various operations on database tables.

Frameworks and technologies used for the development are as follows

Frontend	Bootstrap, CSS, HTML, JavaScript
Backend	Java 1.8 v, Hibernate, Spring
Database	MySQL
Server	Tomcat Apache v-7
IDE	eclipse

4.1 Proposed System

The proposed system is a web-based application which will allow all the users to send secret messages in encoded format and decode the encoded message. It depends on the user, whether they are interested in encoding and sending the message to their friend or decoding the encoded message received from their friend. The application implements the proposed algorithm for encryption and decryption which makes it secure to transmit confidential information.

The sequence flow of the application would be as follows.

The below attached image shows the Sequence flow for the sender. The sequence starts from the login action. If the sender completes the login process with correct credentials, then they can access the home page and choose the encryption action. Where they can type their secret message, receiver's mail ID and provide the cover image for encoding of the secret message. After providing all details, just one click on the encode button does encoding using AES, QR, and LSB steganography. This encoded image is then passed to the receiver's mail ID with an OTP key.

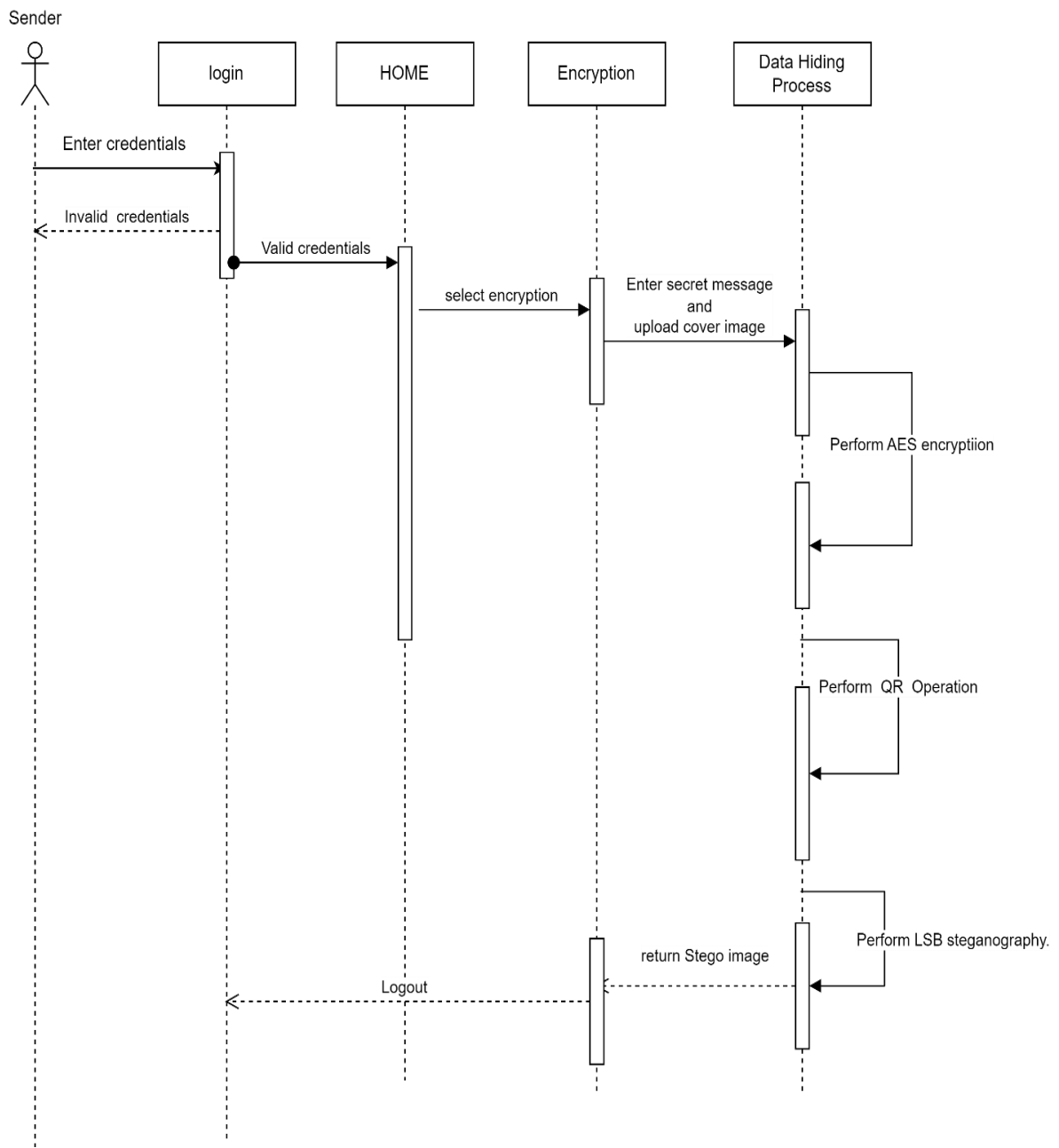


Figure 13 Sender role and flow using SecureIT platform

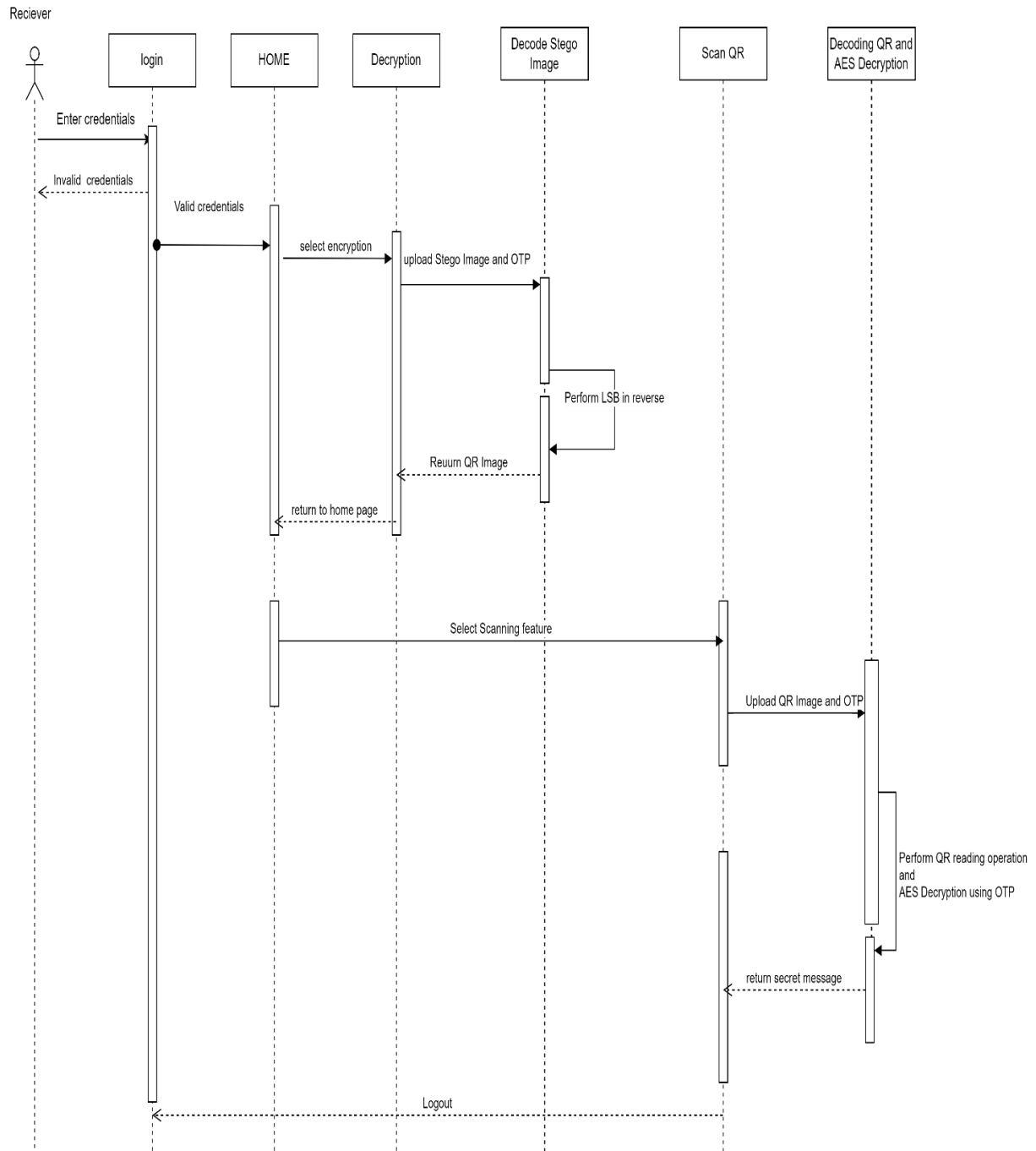


Figure 14 Receiver role and flow using SecureIT platform

The above attached image showcases the sequence flow for the receiver. Login action is mandatory for all the users. Now receiver has to click on decrypt message which asks for an encoded image and OTP received on mail id. Once receiver provide required details and click on decode button, it will extract the QR image from the encoded cover image. Now to extract the secret message receiver has to use the scan functionality and provide the QR image and enter the OTP again. This action will fetch the and decrypt the cipher text from QR and show the secret message to the receiver.

4.2 Application User Interface

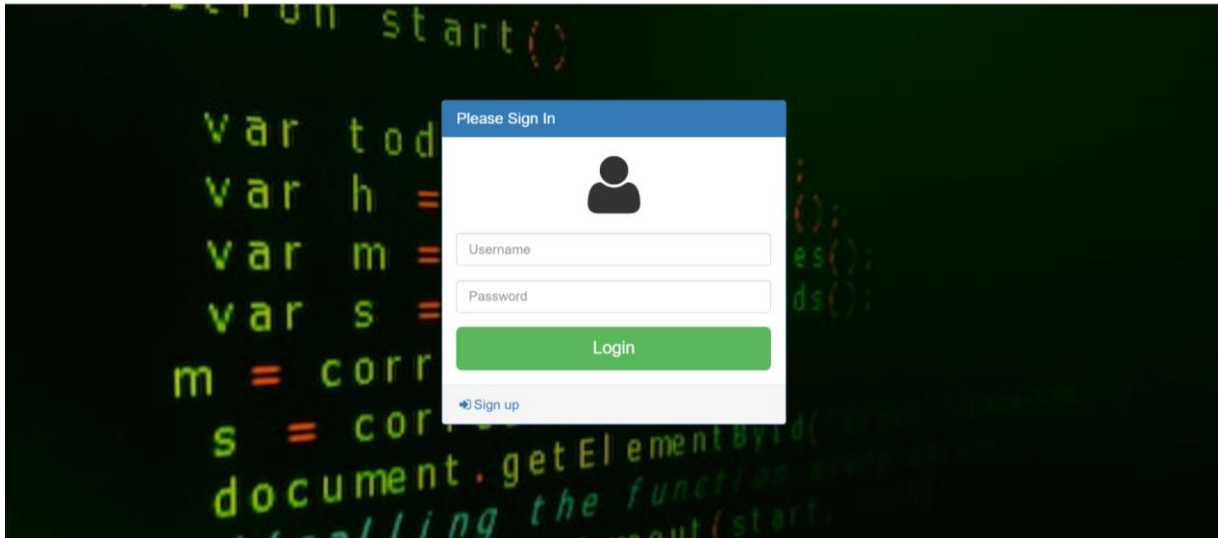


Figure 15 Login Page

Figure (14) shows the Login user interface which asks user for the valid credentials and then only allows to use the encoding and decoding functionalities. If the user does not have an account then they can sign up using the sign up link below login button, which will take them to the sign up page which looks like below attached image

Welcome [New User](#)

[Home](#) / [User Signup](#)

Add User

First Name:

Last Name:

UserName:

Password:

Email Address:

Mobile Number:

Figure 16 Sign up Page

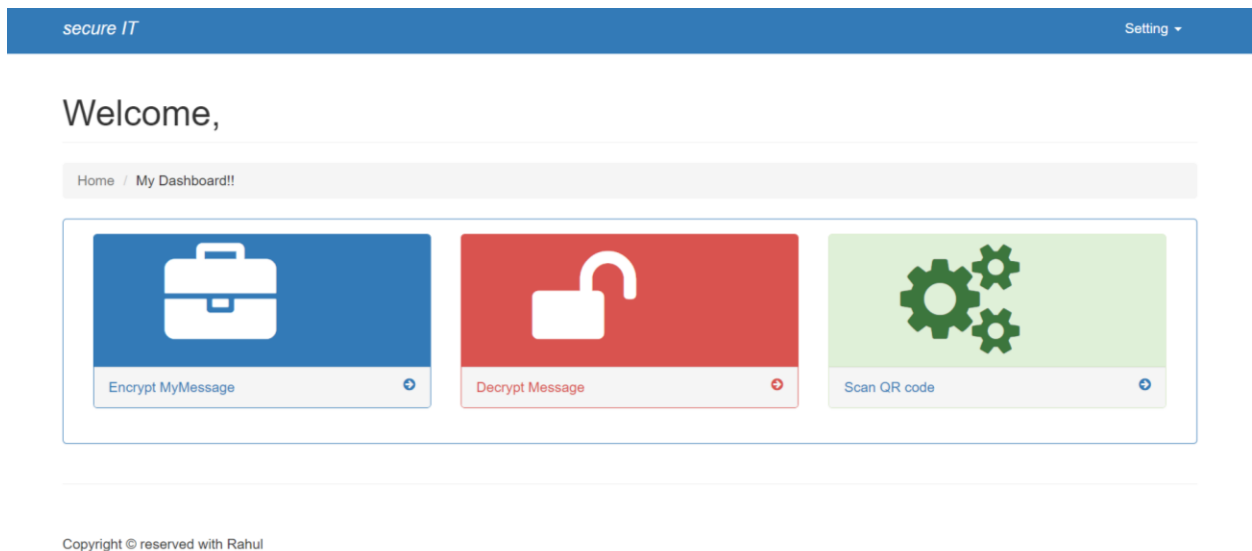


Figure 17 Home Page

Once User enters the correct credentials, they get access to the home page which is shown in figure 16. Home page allows user to choose the task they want to perform of their choice. Let's suppose user wants to send a secret message to his friend then user has to click on encrypt my message tab which leads to the Encryption page and it looks like below image.

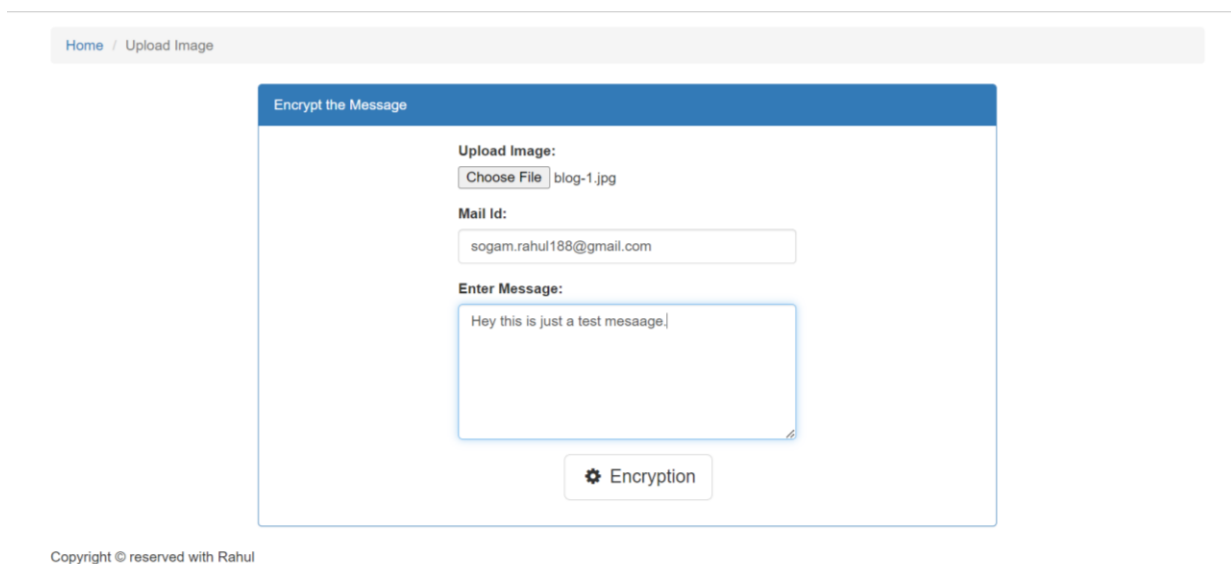


Figure 18 Encryption page

Here sender will write their secret message and select the cover image of their choice and provide the receiver's email id. Once everything is filled sender has to click on Encryption button which does the task for them and gives an encoded image. Firstly, the message is encrypted into a cipher text using AES algorithm which looks like shown in below image.

```
Secret Message Before ENcryption :Hey this is a test messgae
Secret Message after ENcryption :x3FACTxEjHlnf3Ak8F9/rZ6PmotFchc5XJ3+pbzCOMc=
Hibernate: insert into stegnographic_container (isActive, modifiedImageName, orignalImageNam
```

Figure 19 Output of AES encryption

Once the sender clicks the button, they get an option to download an encoded image. If Sender wants the encoded image, then they can download it from the below shown link. Receiver will receive the same encoded image and OTP on their mail id used by sender. Original Image and encoded image are shown in below pictures

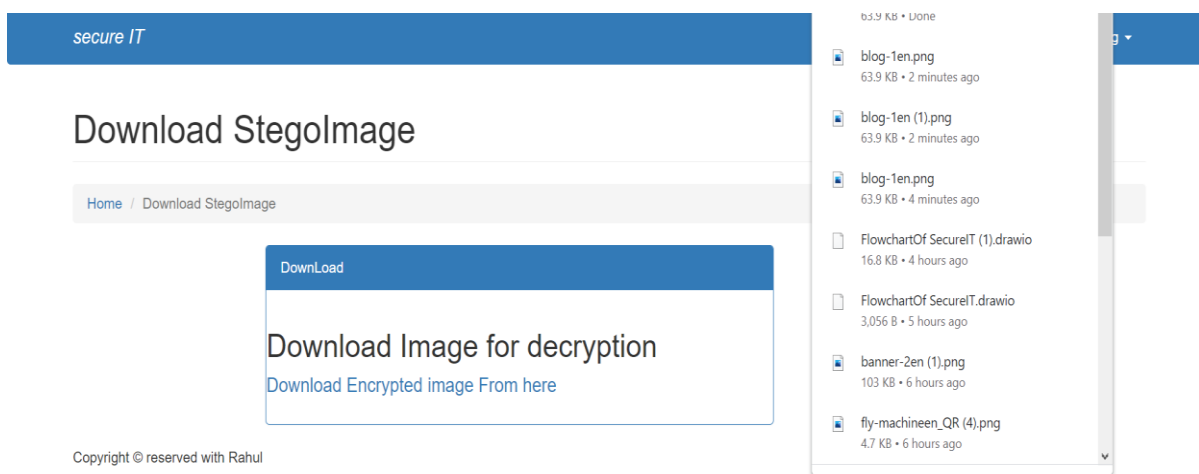


Figure 20 Download encoded image.



Figure 21 Original Cover Image



22 Stego Encoded Image

Once sender sends an encoded message to the receiver, receiver has to login and go to the decrypt message tab, upload the encoded image and OTP received on an email and click on decode button.

Download StegoImage

Home / Upload Image

Decode Image

File To Decode:

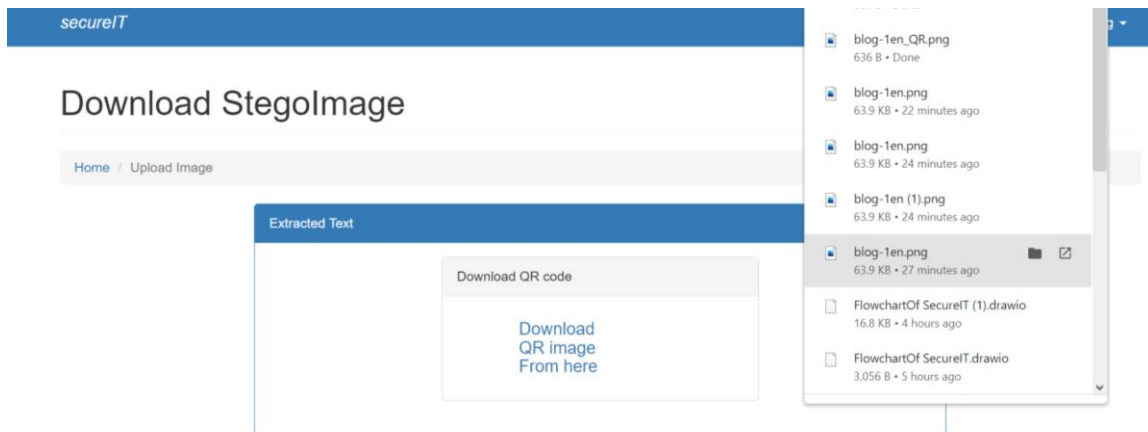
Choose File

Enter OTP:

Copyright © reserved with Rahul

Figure 23 Decoding image

Decode action will perform the extraction task using LSB in reverse manner. It first extracts the hidden QR code out of encoded image which contains the secret message in an encrypted format and allows user to download the same QR code.



Copyright © reserved with Rahul

Figure 24 Download extracted QR

Extracted QR Image from cover image looks like shown in below figure 24 which contains the secret message but in its encrypted format so even if any unauthorized user gets access, they cannot read the actual secret message.



Figure 25 Extracted QR

Usually, Users can perform the scanning of the QR code but as it contains the encrypted data nobody could be able to understand the message just by scanning the QR hence receiver will have to pass this QR image to the scanning functionality of the portal where receiver needs to provide OTP and QR image file like shown in below figure 25.

Download StegoImage

[Home](#) / [Upload Image](#)

Decode Image

File To Decode:

blog-1en_QR.png

Enter OTP:

Copyright © reserved with Rahul

Figure 26 Scanning QR Image

Once receiver provide QR image, OTP and clicks on scan button, scanning of QR method gets called and secret message gets extracted from the QR in encrypted format then this encrypted message is passed on to decryption method, where AES decryption process does its work with the provided OTP using it as key and finally, they get a decrypted secret message which looks like following:

secret message extracted from QR : 9SJkqv4J9prcstdQt/+vkw==
secret message after decrypting : Hey how are you

Figure 27 secret message extracted from QR

Download StegoImage

Home / Upload Image

Extracted Text

Your Secret Message

Hey this is just a test message.

Copyright © reserved with Rahul

Figure 28 Decrypted Secret Message.

Chapter 5

Discussion:

The plan for employing steganography, Image processing and cryptography to safely transfer the data is put into practice. Nevertheless, there were a lot of obstacles to overcome when carrying out the strategy. The offered idea could be altered in a few specific ways. The design completes the idea for information transmission as well. The findings are identified in some cases.

- It becomes difficult to extract information from a QR code if an attacker decides to disrupt the stego image using different steganographic methods since the QR code's dimensions and pixels are altered. But still the designed system is secure because the attacker can attack only stego image and cannot get any kind of data out of that. No attacker can decode the QR without knowing its size.
- Once the encoding is done OTP and encoded image is sent to receivers mentioned mail id. This ensures that the only authorised receiver gets a key for AES decryption and encoded image to extract the QR code.
- No message related data is being store in a database except the OTP which makes sure that no user who has access to the database can read the image or message in any format.
- If any unauthenticated user tries to decode the image, they need a valid OTP which is only known to the authenticated receiver. No unauthorized user can decode the cover image as at the backend OTP validation is performed.
- In previous paper, author [13] talked about hiding the QR code in cover image. The issue with this is if any attacker extracts the QR code from cover image then it won't be a big task for an attacker to scan and read the secret message. This issue is being solved in this proposed methodology by implementing the AES cryptography algorithm on secret message before generating QR and storing it into a cover image.
- The encryption and decryption times of the suggested approach are quicker than [20].

Comparison	Field	Robust	Encoding time	Decoding time
[20]	Secret Sharing	High	8.01 sec	8.01 sec
Proposed	Secret Message Sharing	High	2.79 sec	2.10 sec

Chapter 6

Conclusion:

6.1 Summary

The goal of this proposed methodology is to securely transfer the data from one user to another by implementing the AES cryptography scheme for securing the secret message in QR code and steganography. Our design achieved the idea of protecting the text message from being attacked or altered by any attacker while transferring from to the receiver. AES algorithm does the work of converting text to cipher text using the key and transferring same key to the receiver to decrypt the cipher text. The goal also ensures that no message related information is being stored in database so no person having access to the database can read the data or change the data. Also, with the help of Java, Spring, hibernate and MYSQL one platform to perform all these operations is developed which allows users to practically use these functionalities and share the message. However after all these research and implementations there is always a scope for a future research and work which is as follows.

6.2 Limitation:

Encryption of messages and images sent through the device is done at the sender's end. However, the decryption key and image are passed through a third-party email application where receiver needs to have an account to receive OTP and encoded image.

Steganographic image is vulnerable to the steganographic attacks however attacker cannot extract or read the message because of encryption and misplaced pixels of data.

QR code can hold up to 2000 characters of data at a time.

In the proposed methodology the cover medium used for hiding data is limited to the image format which could be extended to other mediums like audio or video format which could help to contain more data in a cover medium and more difficult for attacker to attack the encoded cover medium.

6.3 Future work:

Some watermarking technique can be used to get the image back in all the steganographic attacks to increase the data storage capacity in QR instead of using black and white QR codes colour QR code can be used which allows to transmit more information in a small size of QR image. Some more cover medium can be used. More options for cover medium like audio and video can be used to hide the data

References

- [1] Musa. M. Yahaya, Aminat Ajibola. "Cryptosystem for Secure Data Transmission using Advance Encryption Standard (AES) and Steganography". International Journal of Scientific Research in Computer Science Engineering and Information Technology (IJSRCSRI). Dec 2019.
- [2] M. Mary Shanthi Rani, K. Rosemary Euphrasia. "Data Security Through QR code Encryption and Steganography." Advanced Computing: An International Journal (ACIJ). Mar 2016.
- [3] p. Vijayakumar and V. Vijayalakshmi, "An Improved Level of Security for DNA Steganography Using Hyperelliptic Curve Cryptography," Wireless Personal Communications, vol. 89, no. 4.
- [4] R. Pranesh, M. Vigneshwaram, V. Harish, G. Manikandan. "A new approach for secure data transmission". IEEE. Aug 2016.
- [5] Varghese F, Sasikala P. "A Detailed Review Based on Secure Data Transmission Using Cryptography and Steganography. Wireless Personal Communications". Mar 2023
- [6] Nunna KC, Marapareddy R. "Secure data transfer through internet using cryptography and image steganography.". In 2020 SoutheastCon 2020 Mar (Vol. 2, pp. 1-5). IEEE
- [7] Phaisarn Sutheebanjard; Wichian Premchaiswadi. "QR-code generator. 20 Jan 2011
- [8] Pandey HM. "Secure medical data transmission using a fusion of bit mask oriented genetic algorithm, encryption and steganography. Future Generation Computer Systems," 2020 Oct 1.
- [9] Osuolale, A. Festus. "Secure Data Transfer Over the Internet Using Image CryptoSteganography". International Journal of Scientific and Engineering Research . December 2017
- [10] A. Gambhir and A. R. Mishra, "A New Data Hiding Technique with Multilayer Security System," International Journal of Innovations & Advancement in Computer Science IJIACS, vol. 4, May 2015.

- [11] R. K. H S, P. K. H R, S. K B and G. Aithal, "ENHANCED SECURITY SYSTEM USING SYMMETRIC ENCRYPTION AND VISUAL CRYPTOGRAPHY," in International Journal of Advances in Engineering & Technology, 2013.
- [12] M. Mishra, G. Tiwari and A. K. Yadav, "Secret communication using Public Key steganography," in International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2014), Jaipur, India, 2014.
- [13] Prof. Shrikant Dhamdhere, Sarthak Thorat , Shweta Patil, Sejal Dolas, Omkar Panchal, "Data encryption through QR code and steganography," in International Journal of Advances in Science and Engineering, 2023.
- [14] C. J. Chou, Y. C. Hu and K. H. Ju, "A Novel Secret Sharing Technique Using QR Code," International Journal of Image Processing, vol. 4, 2010.
- [15] N. Goel, A. Sharma and S. Goswami, "A way to secure a QR code: SQR," in 2017 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, India, 2017.
- [16] P. Singh, R. Chaudhary and A. Agarwal, "A Novel Approach of Text Steganography based on null spaces," IOSR Journal of Computer Engineering (IOSRJCE), vol. 3, no. 4, pp. 11-17, 2012.
- [17] W. W. Zin and T. N. Soe, "Implementation and analysis of three steganographic approaches," in 2011 3rd International Conference on Computer Research and Development, Shanghai, China, 2011.
- [18] M. M. Amin, M. Salleh, S. Ibrahim, M. R. Katmin and M. Z. Shamsuddin, "Information hiding using steganography," in 4th National Conference of Telecommunication Technology, 2003. NCTT 2003 Proceedings., Shah Alam, Malaysia, Malaysia , 2003.
- [19] B. Karthikeyan, A. C. Kosaraju and S. G. S, "Enhanced security in steganography using encryption and Quick Response code," in 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, India, 2016.
- [20] A. Mendhe, D. K. Gupta and K. P. Sharma, "Secure QR-Code Based Message Sharing System Using Cryptography and Steganography," in 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), India, 2018.