A Project Report on

IMPROVED SECURITY USING STEGANOGRAPHY AND CRYPTOGRAPHY BASED ON SMARTPHONE USERS LOCATION

Submitted in partial fulfilment for the award of

Bachelor of Technology

in

Computer Science and Engineering

By

CH. Mohith Agasthyeswar(Y19ACS428) B. Siva Pavani(Y19ACS414)
CH. Praveen Kumar(Y19ACS431) K. Prudhvi Raj(Y19ACS484)

Under the guidance of
Mr. T. Nagarjuna, Assistant Professor

Department of Computer Science and Engineering
Bapatla Engineering College
(Autonomous)
(Affiliated to Acharya Nagarjuna University)
BAPATLA – 522102, Andhra Pradesh, INDIA
2022-2023

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the project report entitled **Improved Security Using Steganography and Cryptography Based on Smartphone Users Location** that is being submitted by **CH. Mohith Agasthyeswar (Y19ACS428), B. Siva Pavani (Y19ACS414), CH. Praveen Kumar (Y19ACS431), K. Prudhvi Raj (Y19ACS484)** in partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Science and Engineering to the Acharya Nagarjuna University is a record of bonafide work carried out by them under my guidance and supervision.

Date:

T. Nagarjuna
Assistant Professor

Dr. P. Pardhasaradhi
Professor and HoD

Declaration

We declare that this project work is composed by ourselves, and that the work contained herein is our own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

CH. Mohith Agasthyeswar (Y19ACS428)

B. Siva Pavani (Y19ACS414)

CH. Praveen Kumar (Y19ACS431)

K. Prudhvi Raj (Y19ACS484)

Acknowledgement

We sincerely thank the following distinguished personalities who have given their advice and support for successful completion of the work.

We are deeply indebted to our most respected guide **Mr. T. Nagarjuna**, Assistant Professor, Department of CSE, for his valuable and inspiring guidance, comments, suggestions and encouragement.

We extend our sincere thanks to **Dr. P. Pardhasaradhi**, Prof. and Head of the Dept. for extending his cooperation and providing the required resources.

We would like to thank our beloved Principal, **Dr. Sk. Nazeer** for providing the online resources and other facilities to carry out this work.

We would like to express our sincere thanks to the project coordination committee and our project coordinator **Dr. N. Sudhakar,** Prof., Dept. of CSE for their helpful suggestions throughout the project work and in presenting this document.

We extend our sincere thanks to all other teaching faculty and non-teaching staff of the department, who helped directly or indirectly for their cooperation and encouragement.

CH. Mohith Agasthyeswar (Y19ACS428)
B. Siva Pavani (Y19ACS414)
CH. Praveen Kumar (Y19ACS431)
K. Prudhvi Raj (Y19ACS484)

Abstract

Smart phones have become one of the most important used devices worldwide. This is, in fact, due to the significant advances in: communication technology, mobility, and the development of various types of these devices. They have been used for communication, entertainment, navigation, etc. Smart phones are usually integrated with navigation systems such as The Global Positioning System (GPS). As a result, the location of Smartphone users can be determined via such systems.

The security of transmitting information over mobile phone networks is an issue by itself. In this proposed a system that integrates the location of the Smartphone obtained via GPS with steganography for transmitting confidential information, and an encryption method to improve the security of the transmitted information. The proposed system has implemented the Twofish Algorithm to encrypt confidential information, and the RSA algorithm to encrypt the GPS coordinates that will serve as keys.

A number of measures of image quality were used to evaluate the proposed system such as MSE, PSNR, SSIM and NCC. As a result of this integration, the proposed system showed more flexibility and a high level of security.

Table of Contents

List of Figures

List of Tables

List of Equations

1 Introduction

Mobile phones have recently become much more powerful than before. Increased memory capacity, higher processor performance, and larger features like accelerometers, light sensors, larger camera pixels, and other uses have pushed the limits of the modern mobile phone even further. Previously, mobile phones were used only to make phone calls; however, with the coming of the smartphone, the mobile phone has evolved into a low-power handheld processing system.

Currently, smartphone usage can be attributed to social enterprises, networks, and some cases; it is a faster way ever to share videos, photos, texts and emails and much more. The computer-like functionality of cell phones providing an all-in-one portable device in terms of interconnection has made smart phones as an integral part of individuals living in this century. Users of mobile phones want more secure and private communication in their daily lives. This is especially critical in classified communications, such as military and government communications.

Services such as GPS are used to determine the location of the mobile phone's users and also to help them in finding places as a navigation application. many security problems still exist because the mechanisms are in place to ensure the protection of data and information are insufficient. Many researchers have tried using either encryption or information hiding.

Figure 1.1 shows how data accessed by intruder before applying security and after applying security.To secure the secret information exchange, cryptography and steganography are utilized. Steganography is the art and science of hiding information by embedding a message within a cover file to produce a secret stego file.

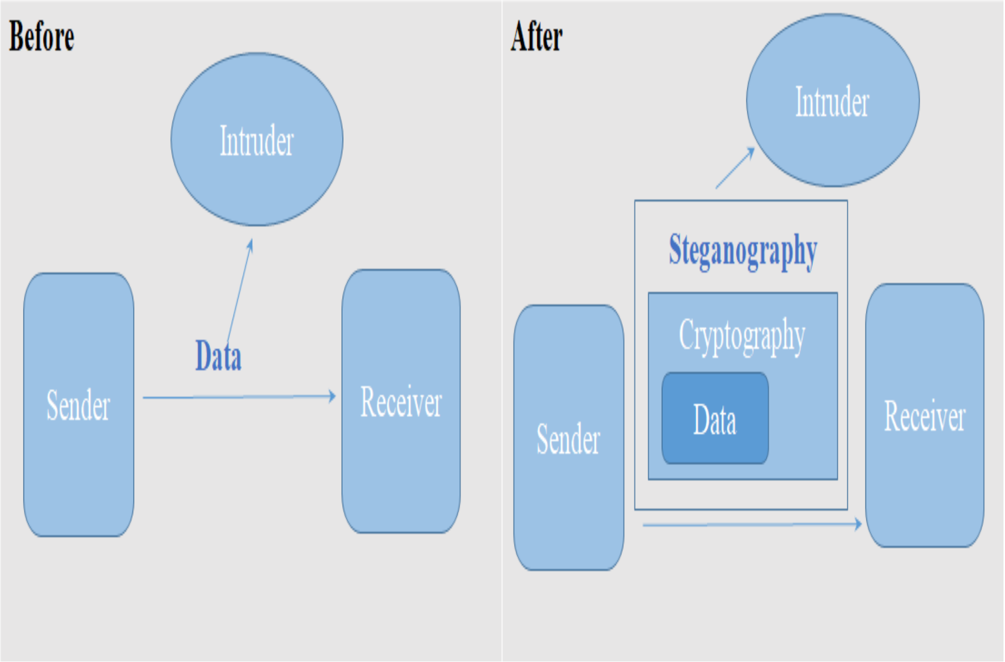


Figure 1.1 Providing Security to Data

In this a solution is proposed that can enhance the security feature of the encryption and steganography method using location for the mobile user that is calculated using the GPS.

1.1 Cryptography

The art of cryptography is considered to be born along with the art of writing. As civilizations evolved, human beings got organized in tribes, groups, and kingdoms. This led to the emergence of ideas such as power, battles, supremacy, and politics. These ideas further fueled the natural need of people to communicate secretly with selective recipient which in turn ensured the continuous evolution of cryptography as well. The roots of cryptography are found in Roman and Egyptian civilizations.

The word ‘cryptography’ was coined by combining two Greek words, ‘krypto’ meaning hidden and ‘graphene’ meaning writing.

The importance of information and communication systems for society and the global economy is intensifying with the increasing value and quantity of data that is transmitted and stored on those systems. At the same time those systems and data are also increasingly vulnerable to a variety of threats, such as unauthorized access and use, misappropriation, alteration, and destruction. Now a days the society needs more security for their communication.

The hiding of information is called encryption, and when the information is unhidden, it is called decryption. A cipher is used to accomplish the encryption and decryption. Cipher is defined as a method of transforming a text in order to conceal its meaning. The information that is being hidden is called plaintext; once it has been encrypted, it is called cipher text.

To hide any data, we use Cryptography. Cryptography is the science of protecting data, which provides methods of converting data into unreadable form, so that Valid User can access Information at the Destination. Cryptography is the science of using mathematics to encrypt and decrypt data.

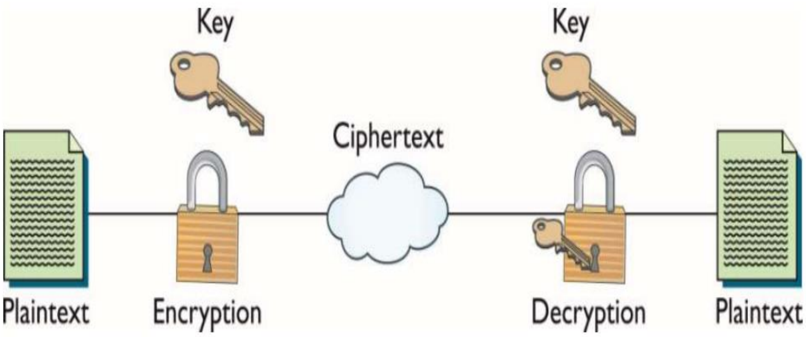


Figure 1.2 Block Diagram of Cryptography

1.1.1 Basic Terminology of Cryptography

Computers are used by millions of people for many purposes such as banking, shopping, military, student records, etc.. Privacy is a critical issue in many of these applications, how ever we need to make sure that an unauthorized party cannot read or modify messages. Many algorithms are developed for transforming plaintext into cipher text.

Figure 1.2 Shows the block diagram of Cyptography. Cipher is the algorithm that is used to transform plaintext to cipher text, this method is called encryption, in other words, it's a mechanism of converting readable and understandable data into "meaningless" data. The Key is an input to the encryption algorithm, and this value must be independent of the plaintext, this input is used to transform the plaintext into cipher text, so different keys will yield different cipher text, In the decipher side, the inverse of the key will be used inside the algorithm instead of the key.

Computer security is a generic term for a collection of tools designed to protect any data from hackers, theft, corruption, or natural disaster while allowing these data to be available to the users at the same time. The example of these tools is the antivirus program.

Network security refers to any activity designed to protect the usability, integrity, reliability, and safety of data during their transmission on a network, Network security deals with hardware and software. The activity can be one of the following anti-virus and anti- spyware, firewall, Intrusion prevention systems, and Virtual Private Networks.

Internet Security is measures and procedures used to protect data during their transmission over a collection of interconnected networks, while information security is about how to prevent attacks, and to detect attacks on information-based systems.

1.1.2 Cryptography Goals

By using cryptography many goals can be achieved, these goals can be either all achieved at the same time in one application, or only one of them. These goals are:

Confidentiality: It is the most important goal, that ensures that nobody can understand the received message except the one who has the decipher key.

Authentication: It is the process of proving the identity, that assures the communicating entity is the one that it claimed to be. This means that the user or the system can prove their own identities to other parties who don’t have personal knowledge of their identities.

Data Integrity: It ensures that the received message has not been changed in any way from its original form. The data may get modified by an unauthorized entity intentionally or accidently. Integrity service confirms that whether data is intact or not since it was last created, transmitted, or stored by an authorized user. This can be achieved by using hashing at both sides the sender and the recipient in order to create a unique message digest and compare it with the one that received.

Non-Repudiation: It is mechanism used to prove that the sender really sent this message, and the message was received by the specified party, so the recipient cannot claim that the message was not sent. For example, once an order is placed electronically, a purchaser cannot deny the purchase order, if non-repudiation service was enabled in this transaction. Access Control: it is the process of preventing an unauthorized use of resources. This goal controls who can have access to the resources, If one can access, under which restrictions and conditions the access can be occurred, and what is the permission level of a given access.

1.1.3 Types of Cryptography

Types of Cryptography are:

1. Symmetric key Cryptography

2. Asymmetric key Cryptography

Symmetric key: In symmetric key cryptography is also known as private-key cryptography, a secret key may be held by one person or exchanged between the sender and the receiver of a message. If private key cryptography is used to send secret messages between two parties, both the sender and receiver must have a copy of the secret key.

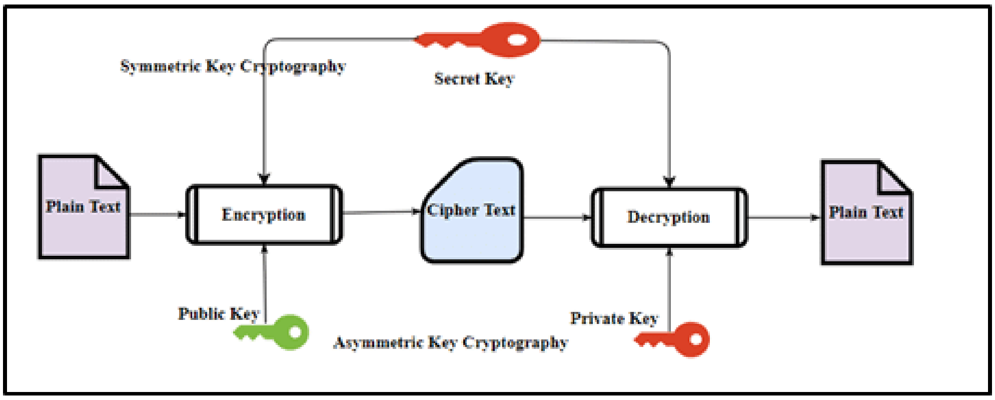


Figure 1.3 Types of Cryptography

Asymmetric key: In the two-key system is also known as the public key system, one key encrypts the information and another, mathematically related key decrypts it. The computer sending an encrypted message uses a chosen private key that is never shared and so is known only to the sender. If a sending computer first encrypts the message with the intended receiver’s public key and again with the sender’s secret, private key, then the receiving computer may decrypt the message, first using its secret key and then the sender’s public key. Using this public-key cryptographic method, the sender and receiver are able to authenticate one another as well as protect the secrecy of the message.

Figure 1.3 shows the difference between Symmetric key Cryptography and Asymmetric key Cryptography in detail.

1.1.4 Applications of Cryptography

Cryptography can be applied in many areas to provide better security from intruders, Some of the applications are:

Time stamping: Time stamping is a technique that can certify that a certain electronic document or communication existed or was delivered at a certain time. Time stamping uses an encryption model called a blind signature scheme. Blind signature schemes allow the sender to get a message receipted by another party without revealing any information about the message to the other party.

Electronic Money: The definition of electronic money (also called electronic cash or digital cash) is a term that is still evolving. It includes transactions carried out electronically with a net transfer of funds from one party to another, which may be either debit or credit and can be either anonymous or identified. There are both hardware and software implementations.

Disk Encryption: Disk encryption programs encrypt your entire hard disk so that you don't have to worry about leaving any traces of the unencrypted data on your disk.

PGP can also be used to encrypt files. In this case, PGP uses the user's private key along with a user- supplied password to encrypt the file using IDEA. The same password and key are used to unlock the file.

1.2 Steganography

Steganography word came from the Greek words Steganós means Covered and Graptos means Writing. The origin of steganography is the biological and physiological. The term “steganography” came into use in 1500’s after the emergence of Trithemius’ book on the subject “Steganographia”. The overview of steganography field can be divided into three parts:

Past It’s very older origins can be traced back to 440 BC. In early times, messages were hidden on back of the wax writing tables, written on the stomachs of the rabbits, or the tattooed on the scalp of slaves. Invisible ink has been in use for centuries—for fun by children and students and for serious espionage by spies and terrorists. Steganography became very common place in the middle periods.

Present The majority of today’s steganographic systems uses the multimedia objects like image; audio; video etc as cover media because people often broadcast digital pictures over email and other Internet communication. So, in present world of steganography various steganographic techniques have been proposed. There are certain cases in which a combination of Cryptography and Steganography is used to achieve data privacy over secrecy.

Future, “Hacking” is very famous term. It is nothing but an unauthorized access of data which can be collected at the time of the data transmission. With respect to the steganography this problem is called as Steganalysis. Steganalysis is a process in which a steganalyzers cracks the cover object to get the hidden data. It is hoped that Steganography along with Cryptography may improve the privacy as well as secrecy.

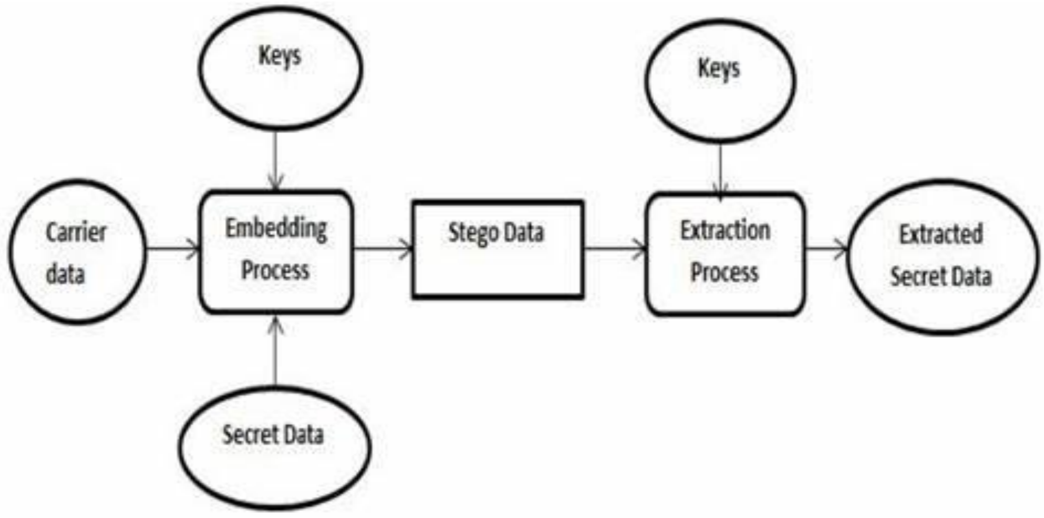


Figure 1.4 Block Diagram of Steganography

Figure 1.4 shows the block diagram of Steganography. Steganography is a method of hiding secret data, by embedding it into an audio, video, image or text file. It is one of the methods employed to protect secret data or sensitive data from malicious attacks. We can conceal a secret message in any media using this tool.

1.2.1 Types of Steganography

Types of Steganography are:

1. Text Steganography
2. Image Steganography
3. Audio Steganography
4. Video Steganography
5. Network or Protocol Steganography

Text Steganography: It consists of hiding information inside the text files. In this method, the secret data is hidden behind every nth letter of every words of text message. Numbers of methods are available for hiding data in text file. These methods are:

1. Format Based Method
2. Random and Statistical Method
3. Linguistics Method.

Image Steganography: Hiding the data by taking the cover object as image is referred as image steganography. In image steganography pixel intensities are used to hide the data. In digital steganography, images are widely used cover source because there are number of bits presents in digital representation of an image.

Audio Steganography: It involves hiding data in audio files. This method hides the data in WAV, AU and MP3 sound files. There are different methods of audio steganography. These methods are:

1. Low Bit Encoding
2. Phase Coding
3. Spread Spectrum.

Video Steganography: It is a technique of hiding any kind of files or data into digital video format. In this case video (combination of pictures) is used as carrier for hiding the data. Generally discrete cosine transforms (DCT) alter the values which is used to hide the data in each of the images in the video, which is unnoticeable by the human eye. H.264, Mp4, MPEG, AVI are the formats used by video steganography.

Network or Protocol Steganography: It involves hiding the information by taking the network protocol such as TCP, UDP, ICMP, IP etc, as cover object. In the OSI layer network model there exist covert channels where steganography can be used.

1.2.2 Factors affecting a Steganographic Method

The effectiveness of steganographic method may be calculated by comparing the concealed image with the cover Image. There are the factors that determine the efficiency of a technique.These factors are following:

Robustness: It refers to the ability of embedded data to remain undamaged if the stego image undergoes through various transformations, such as linear and non-linear filtering and compression.

Imperceptibility: It means invisibility of a steganographic algorithm. It is the basic requirement, because the strong point of steganography lies in its capability to be unnoticed by the human eye .

Payload Capacity: It is defined as the amount of secret information which is hidden in the cover image.

PSNR (Peak Signal to Noise Ratio): It may be defined as the ratio between the maximum power of a signal and the power of humiliating noise that affects the faithfulness of its representation. The higher value of PSNR represents the better will be quality of the image.

Mean Square Error (MSE): It is defined as the average squared difference between a reference image and a distorted image. The lesser the MSE, the more efficient will be the image steganography technique. MSE is computed pixel-by-pixel by adding up the squared differences of all the pixels and dividing by the total pixel count.

SNR (Signal To Noise Ratio): It is defined as the ratio between the signal power and the noise power. It compares the level of the desired signal to the level of background noise.

1.2.3 Applications of Steganography

Steganography can be used anytime you want to hide data. There are many reasons to hide data but they all boil down to the desire to prevent unauthorized persons from becoming aware of the existence of a message. In the business world steganography can be used to hide a secret chemical equation or plans for a new development. It can also be used for corporate espionage by sending out trade secrets without anyone at the company being any the wiser. Steganography can also be used in the non-commercial sector to hide information that someone wants to keep private. Spies have used it since the time of the Greeks to pass messages undetected. Terrorists can also use steganography to keep their communications secret and to coordinate attacks.

Copyright Protection: A secret copyright notice can be embedded inside an image to identify it as intellectual property. In addition, when an image is sold or distributed an identification of the recipient and time stamp can be embedded to recognize potential pirates. A watermark can also serve to detect whether the image has been subsequently modified.

Secret Communications: In various situations, transmitting a secret message draws unwanted attention. The use of cryptographic technology may be forbidden by law. A trade secret, blueprint, or other sensitive information can be transmitted without alerting potential attackers or eavesdroppers.

Digital Watermark: It is used to recognize ownership of the copyright of such signal. A digital watermark is a type of marker covertly embedded in a noise tolerant signal such as audio or image data.

Used by Terrorists: Steganography on a large scale is used by terrorists, who hide their secret messages in innocent, sover sources to increase terrorism across the country.

Printers: It is also used in the printers. In printers, very small yellow dots are inserted into all pages. Information is hidden inside these yellow dots like serial number of the page or message, date and time stangs. This property is available in laser.

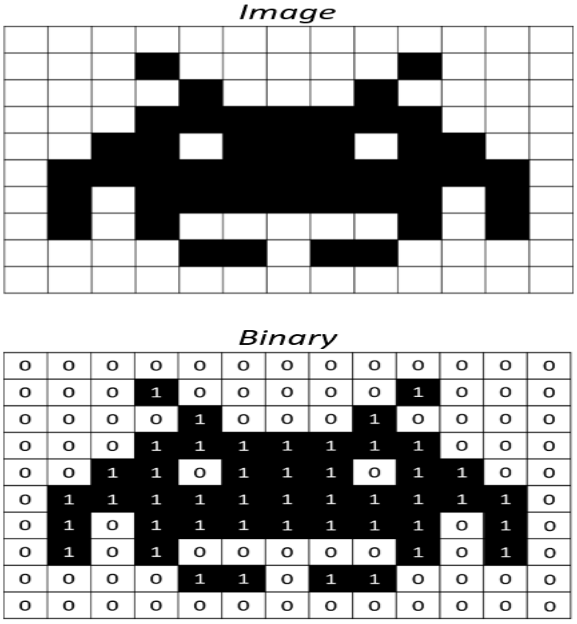
Other applications include the following

1. TV broadcasting
2. Video-audio synchronization
3. Protection of data alteration
4. Companies safe circulation of secret data, Access control system for digital content distribution, TCP/IP packets

1.2.4 Image Steganography

Image Steganography refers to the process of hiding data within an image file.The image selected for this purpose is called the cover-image and the image obtained after steganography is called the stego-image. Types of Images:

Binary Image: It is the simplest type of image. It takes only two values i.e., Black and White or 0 and 1. The binary image consists of a 1-bit image and it takes only 1 binary digit to represent a pixel. Binary images are mostly used for general shape or outline.



0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	1	0	0	0
0	0	0	0	1	0	0	0	1	0	0	0	0
0	0	0	1	1	1	1	1	1	1	0	0	0
0	0	1	1	0	1	1	1	0	1	1	0	0
0	1	1	1	1	1	1	1	1	1	1	1	0
0	1	0	1	1	1	1	1	1	1	0	1	0
0	1	0	1	0	0	0	0	0	1	0	1	0
0	0	0	0	1	1	0	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 1.5 Binary Image

Figure 1.5 shows the difference between normal image and binary image. Binary images are generated using threshold operation. When a pixel is above the threshold value, then it is turned white ('1') and which are below the threshold value then they are turned black ('0')

Example: Optical Character Recognition (OCR).

Grayscale Image: Grayscale images are monochrome images, means they have only one colour. Grayscale images do not contain any information about colour. Each pixel determines available different grey levels.A normal grayscale image contains 8 bits/pixel data, which has 256 different grey levels. In medical images and astronomy, 12 or 16 bits/pixel images are used.

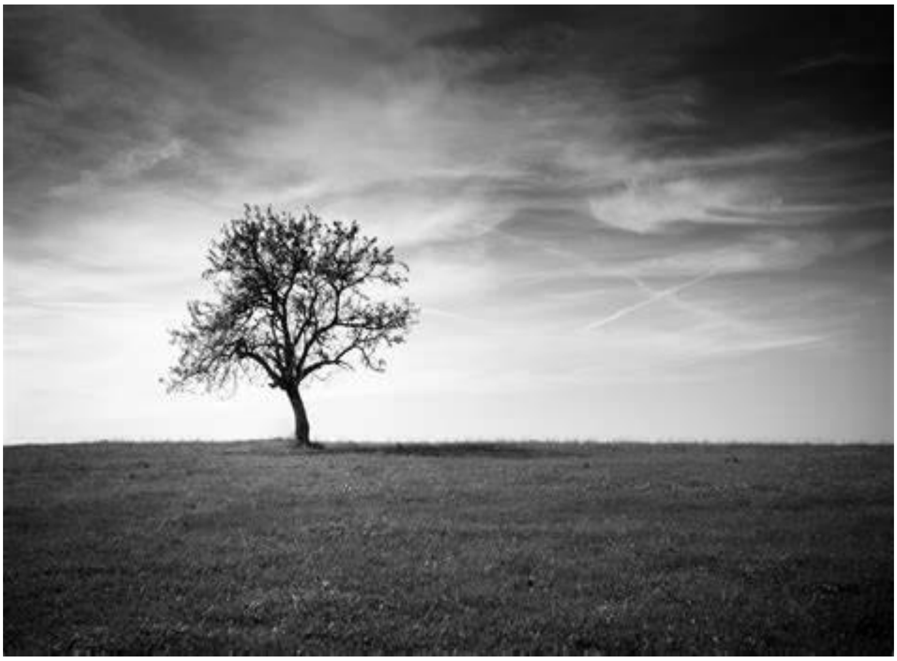


Figure 1.6 Grayscale Image

Colour Image: Colour images are three band monochrome images in which, each band contains a different colour and the actual information is stored in the digital image. The colour images contain Gray level information in each spectral band.

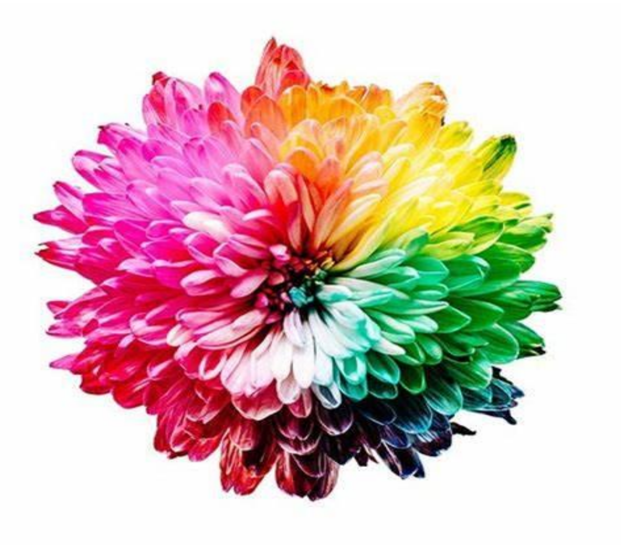


Figure 1.7 Colour Image

The images are represented as red, green and blue (RGB images). And each colour image has 24 bits/pixel means 8 bits for each of the three-colour band (RGB).

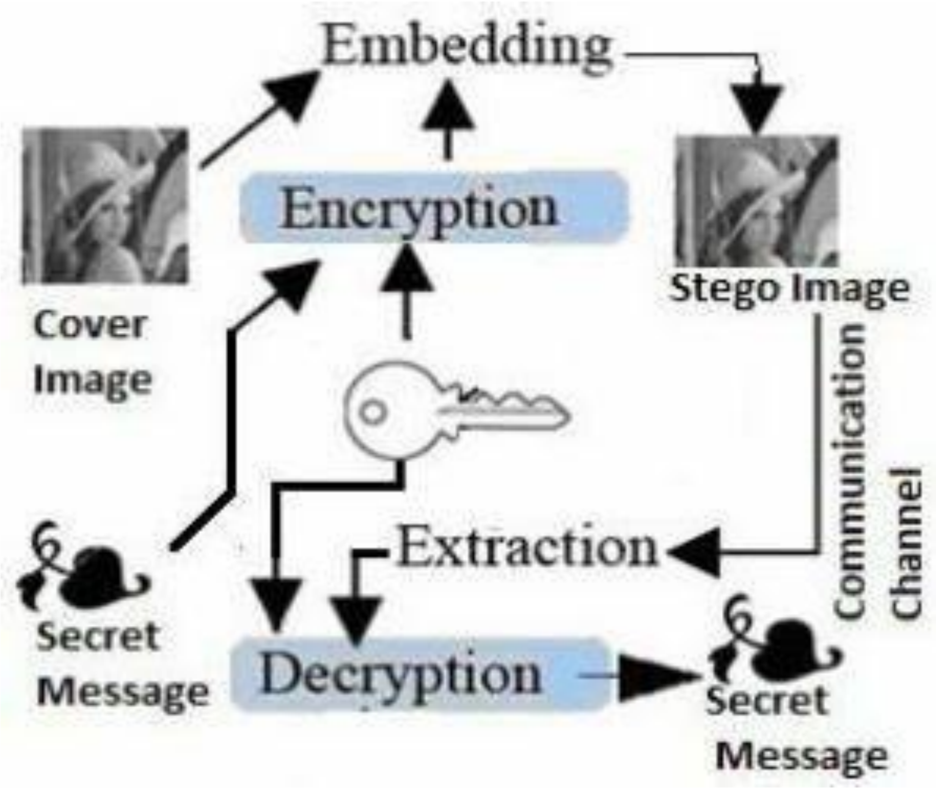


Figure 1.8 Image Steganography

Figure 1.8 shows the block diagram of Basic Image Steganography and its Terminologies. Those terminologies are as follows: -

1. **Cover Image**: Original image which is to be used as a carrier for secreted information.

2. **Message**: Real information which is to be used to hide into images is called a message. It can be a plain text or images.

3. **Stego Image**: Embed message into the cover image is called as stego-image.

4. **Stego Key**: It is used for embedding or extracting messages from the respective cover images and the stego-images.

1.3 Combination of Steganography and Cryptography

Both the techniques are very significant but are better when used together. When cryptography is used alone, the intruder knows the cipher text and if proper cryptanalysis is made the security can be compromised. Steganography when used alone, hides the data but once found by the intruder then decoding the data wouldn’t be a big task. Hence the combined approach is encouraged as it provides confidentiality, integrity and authenticity as well as secrecy to the data. In combined approach the data is hidden from the intruder and even if found, decoding would be difficult, thus providing a double layer of security.

2 Literature Survey

Cryptography and steganography are techniques used to secure data and information in digital communication. Cryptography deals with encryption, decryption, and authentication of data, while steganography is the art of hiding information in plain sight. Smartphones are now ubiquitous and play a crucial role in our daily lives. This literature survey aims to explore the use of cryptography and steganography in improving the security of smartphones user location.

In 2017, Lin You et al., proposed a new location-based encryption model based on a fuzzy vault scheme in their paper. After deciding on an encryption algorithm, they safeguarded the secret key by employing a fuzzy vault scheme based on a location-based digital fingerprint. They used location data captured by the users' mobile devices and linked the digital fingerprint and secret key to create a fuzzy vault to securely store both of them **[**1**]**.

In 2017, Nur et al., proposed the use of symmetric encryption as a solution to encrypt data sent within the client and LBS to solve the problem that the user information was not private because the service provider was aware of the user's location and could leak this information to any unauthorized entities. A symmetric encryption experiment was conducted by using a TCP/IP client-server to protect the user's privacy in the connection to prove that the user's information was not leaked to any unauthorized entities **[**2**]**.

In 2017, Gaikwad et al., developed a method for securing data transmission by encrypting the data to be transmitted and employing the idea of geo-encryption, or location-based encryption, which limits the location of the data to be transmitted. At the time when the data will be decrypted, and the encryption and decryption process is done by using the AES algorithm, the sender supplies a file location and the receiver's time, and the encrypted message is sent. To decode the message, the recipient must be present at the stated location and at the specified time. Message decoding occurs only if the user is present at the stated location and time; otherwise, the message is not decoded [3].

In 2017, Mainak et al., devised an encryption algorithm based on LDEA. The mobile device that provides the latitude and longitude in this case is a smartphone. Latitude and longitude are used to encrypt and decrypt data [4].

In 2018, Sriram et al., focused in their research on the idea that a location-based data encryption algorithm and decryption is in a specific location, where they used the location as an additional security feature due to the improvement of mobile phone networks and GPS technology [5].

In 2020, Nur and Sakinah proposed an improved technology in mobile application design that uses location-based encryption to encrypt data before sending them to cloud storage, as well as a secret keyword to handle the upload and download process during the hashing function to protect the keys stored in cloud storage. The AES method was used to encrypt and decrypt data with location coordinates as an additional encryption key termed geo-lock key due to its great performance. The purpose of using location information to generate the key was to ensure that the decryption process would only occur at the specified location before the encryption process began [6].

3 Design

In Existing Systems, only Cryptography is used to transmit the confidential information, depending on the location of the mobile phone user's coordinates but where as in our System, we use steganography in addition with cryptography, we embed the cipher text in cover image and we transmit this image to the Receiver. If we use only Cryptography the attacker knows that there is some data, but in combination of both Cryptography and Steganography it is to maintain others from thinking that the data even exists.

3.1 Data-Flow Diagrams

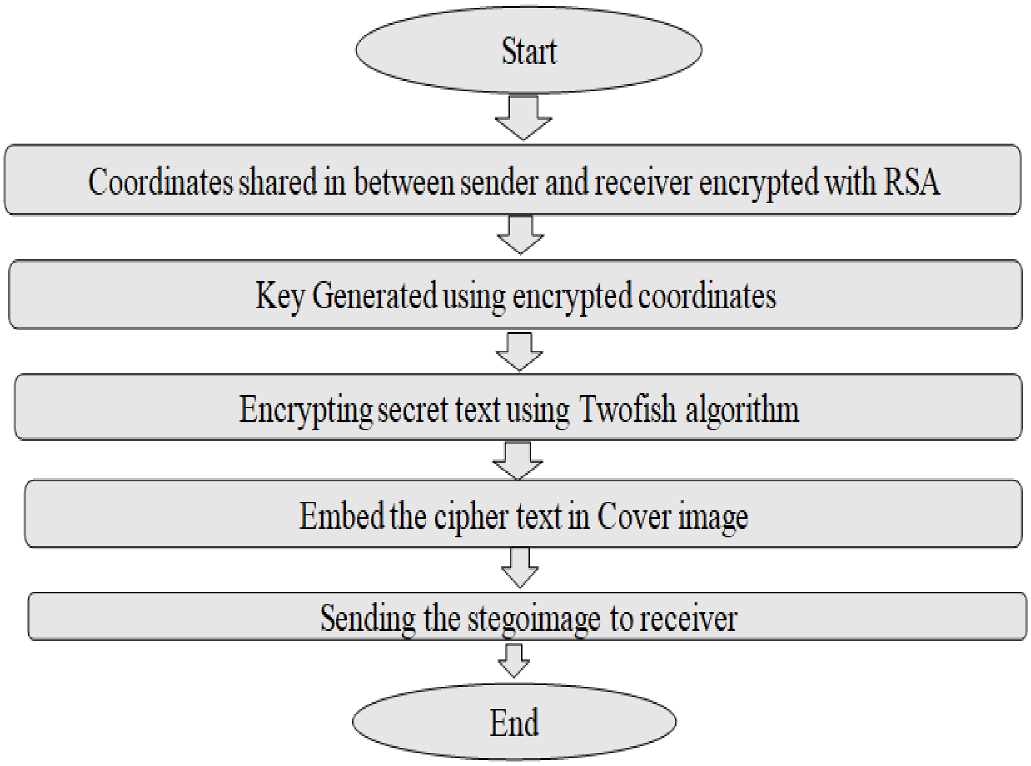

Start
Coordinates shared in between sender and receiver encrypted with RSA
Key Generated using encrypted coordinates
Encrypting secret text using Twofish algorithm
Embed the cipher text in Cover image
Sending the stegoimage to receiver
End

Figure 3.1 Encryption process of Proposed System

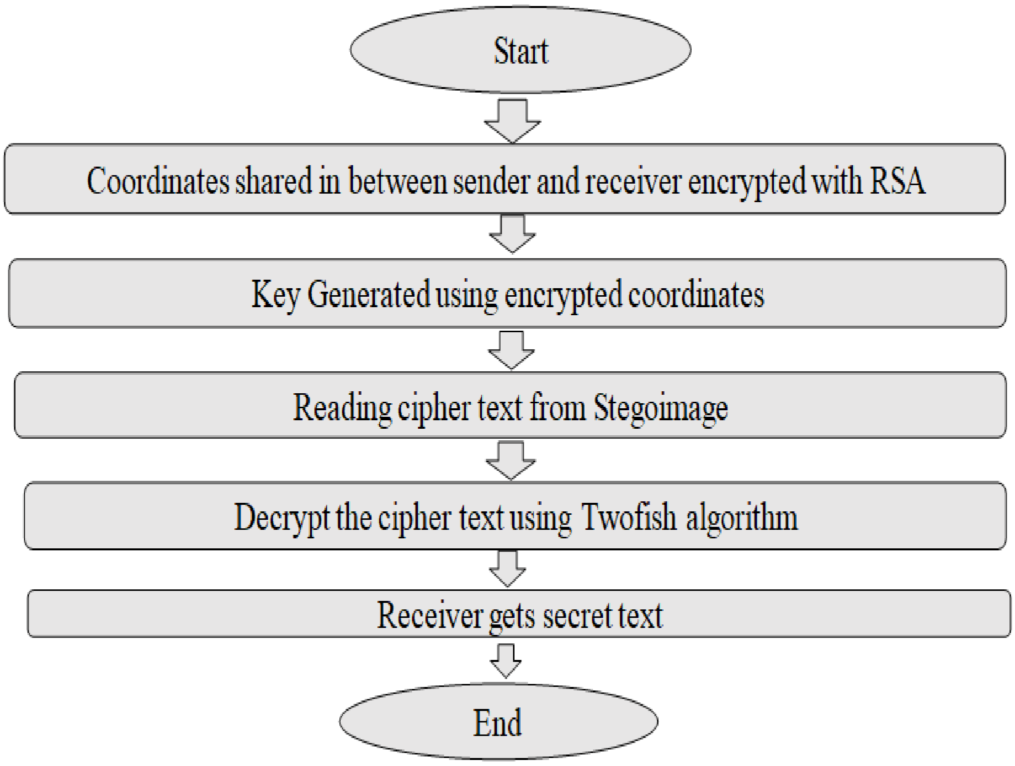

Start
Coordinates shared in between sender and receiver encrypted with RSA
Key Generated using encrypted coordinates
Reading cipher text from Stegoimage
Decrypt the cipher text using Twofish algorithm
Receiver gets secret text
End

Figure 3.2 Decryption process of Proposed System

3.2 Proposed System

As Shown in Figure 3.1and Figure 3.2 there are some common steps followed by both Sender and Receiver like Encrypting Coordinates, Sharing Coordinates, Generating Key, etc. Let us look into it in detail.

3.2.1 Encrypting Coordinates

At First both sender and receiver will collect their GPS coordinates using Smartphone and encrypt them using RSA algorithm.

Both Sender and Receiver after they encrypt their coordinates they share their Encrypted coordinates between them using any social media platform.

3.2.2 Key Generation Process

After sharing the coordinates of both the sender and the receiver, A XOR is performed between the coordinates of the sender and receiver and the result will be the key to the Twofish Algorithm. The Key obtained by both Sender and Receiver are same.

The proposed system is represented in two parts: the first part is the process of embedding text in the cover Authorized licensed image, and the second one is the process of extracting text from the cover image. At first, we will need both the embed and retrieve algorithms to obtain the coordinates of the sender and the receiver by GPS mobile phone and share the coordinates by both parties, which will be used as a key to the algorithm.

3.2.3 Text Embedding Algorithm

Input: Secret message, cover photo.

Output: An image containing the secret message.

The embedding process includes the following steps:

1. Secret text is encrypted by the Twofish Algorithm.
2. Reading cover image data.
3. The pixels in which the text will be embedded are determined by the Mersenne Twister random number generator.
4. After extracting the blue byte from the indicated location, the cipher-text will be embedded in the cover image based on the proposed method for embedding text in the cover image as in Figure 1.1 The embedding process will be done depending on the number of ones and the number of zeros, for example: If the

bit of the cipher text is 0 and the number of ones is greater than the number of zeros in the first three bits of a byte, the bit will be switched from one to 0 and from the right side and then the number of ones and the number of zeros are compared. If the number of zeros does not become more, go to the second bit of the byte and change it to 0 and so on until the number of zeros becomes greater than the number of ones because the secret message bit was 0 and so on. It will continue until the cipher text is completed.

5. Produce a stego image that we send through a social networking program.

6. The End.

3.2.4 Retrieving Text Algorithm

Input: Stego image

Output: Secret message

The retrieving process includes the following steps:

1. Reading the data of the stego image.

2. The pixels in which the text will be extracted are determined by the Mersenne Twister random number generator.

3. After extracting the blue byte from the indicated location, the cipher text will be extracted from the Stego image based on the proposed retrieving method.

4. It will continue until the cipher text is retrieved completed.

5. Decrypting the cipher text and getting the secret message.

6. The end.

3.3 Algorithms used in Proposed System

Algorithms used in this system are:

1. RSA Algorithm
2. Twofish Algorithm

3.3.1 RSA Algorithm

The RSA (Rivest-Shamir-Adleman) cryptography algorithm is asymmetric, both the public and private keys can encrypt a message in RSA cryptography; the opposite key used to encrypt the message which is used to decrypt it. This is one of the reasons why RSA is the most often used asymmetric algorithm [7]. The production of public and private keys is the most difficult aspect of RSA cryptography. The difficulty of factoring in huge prime numbers is what gives RSA its security.

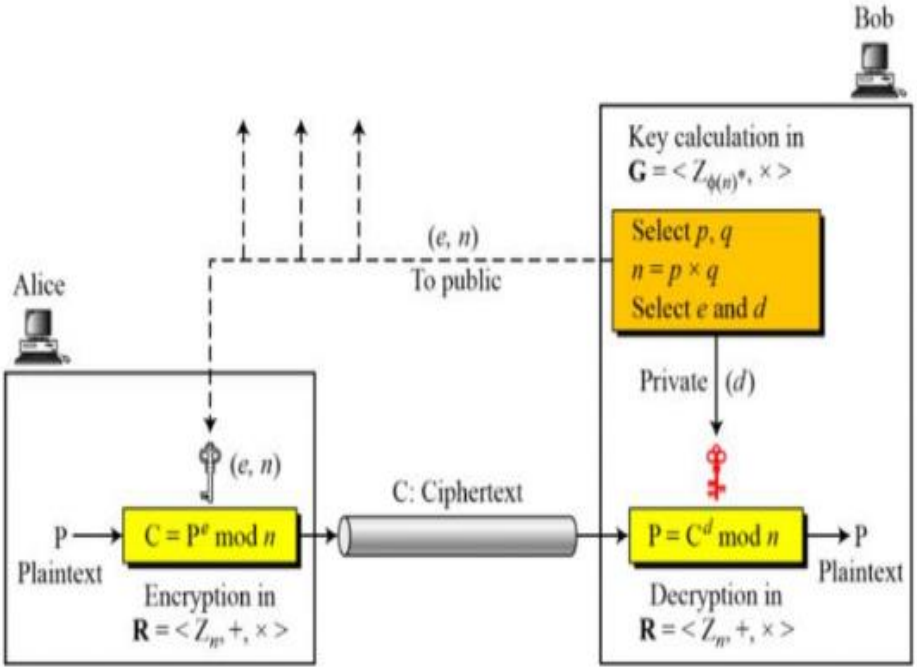


Figure 3.3 RSA Algorithm

3.3.1.1 Encryption

Now, after generating the private and public key we will now encrypt the message. In RSA the plain text is always encrypted in blocks. The binary value of each plain text block should be < n. Encryption is done with the intended receiver's public key. The expression to calculate cipher text as follow:

$$\boldsymbol{c = (m^e) mod\, n} \tag{3-1}$$

3.3.1.2 Decryption

For decryption in RSA, we require a cipher text and the private key of the corresponding public key used in encryption. The expression to calculate plain text is as follow:

$$\boldsymbol{m = (c^d) mod\, n} \tag{3-2}$$

So, Equations 3-1 and 3-2 are used to encrypt and decrypt the message in RSA. It is very important to remember that in RSA we have to encrypt the message using the intended receiver's public key. So, the message can only be decrypted by the intended receiver private key. This provides confidentiality to our message.

3.3.1.3 Advantages of RSA

1. RSA is stronger than any other symmetric key algorithm.
2. RSA has overcome the weakness of symmetric algorithm i.e. authenticity and confidentiality.

3.3.2 Twofish Algorithm

Bruce Schneier, an American cryptographer, initially released The Two Fish in 1998. The TwoFish Algorithm is a form of a block cipher that uses a plaintext of 128 bits and a key size of 128, 192, or 256 bits **[**8**]**. Twofish Algorithm is a symmetric block cipher that handles the 128-bit input message as blocks with a key. It is distinguished by its robust keys and adaptable design. It is efficient, both in terms of hardware and software, and it can be used on a variety of systems. It is also suitable for stream ciphering. Twofish's main work is built on the Feistel network, which has 16 iterations. There is currently no profitable cryptanalysis of Twofish **[**9**] [**10**].**

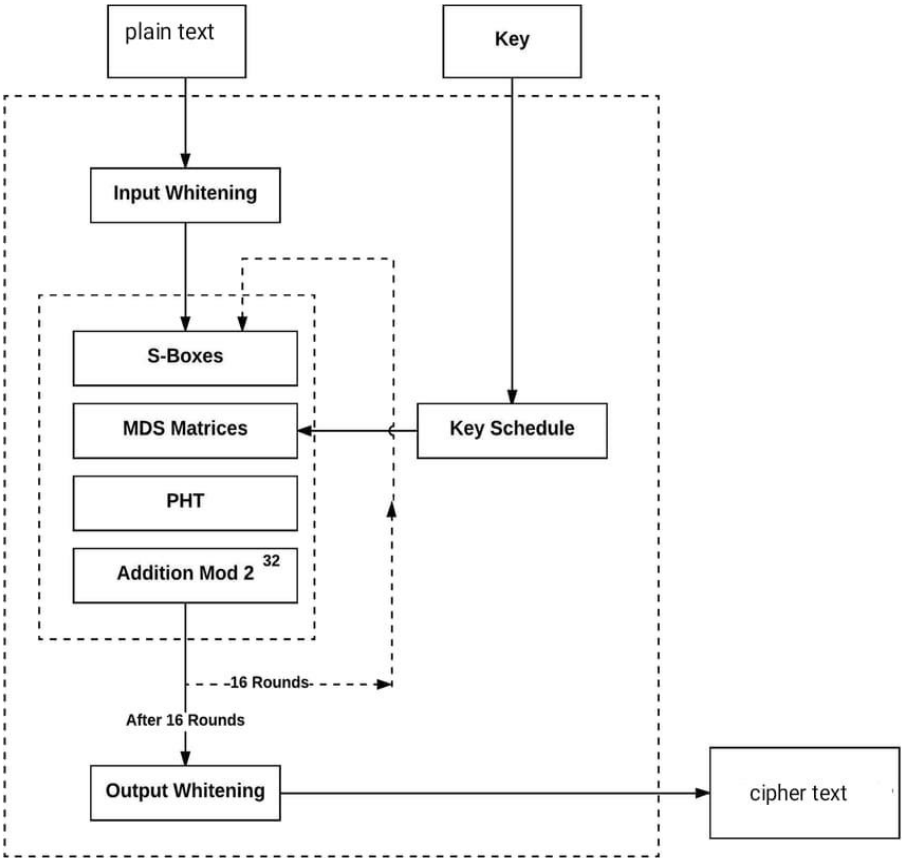


Figure 3.4 Twofish Algorithm

3.3.2.1 Components of Twofish

Twofish consists of six main components, they are:

Feistel Network: The Feistel network is a standard structure used in many block ciphers. It consists of a series of rounds, each of which performs a series of operations on the plaintext. In Twofish, each round consists of four operations: S-boxes, MDS matrices, Pseudo-Hadamard Transforms, Whitening. Twofish is a 16-round Feistel network with a bijective F function.

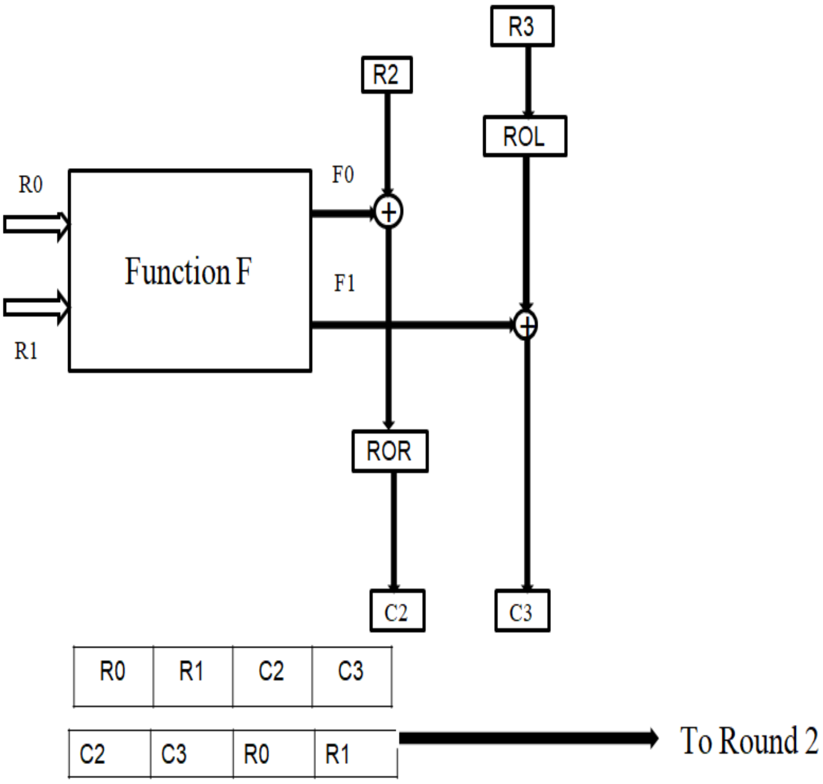


Figure 3.5 F function of the Twofish algorithm

In Figure 3.5 the result words R0, R1, C2 and C3 are the final results of round1. S-box before sending the four words to Round 2 we need to swap them. Twofish has 16 rounds this is only the result of the first round.

S Boxes: Substitution boxes (S-boxes) are used to replace the input bits with new values. Twofish uses a combination of fixed and generated S-boxes, which are derived from the user-supplied key. This makes it difficult for attackers to predict the output of the S-boxes.

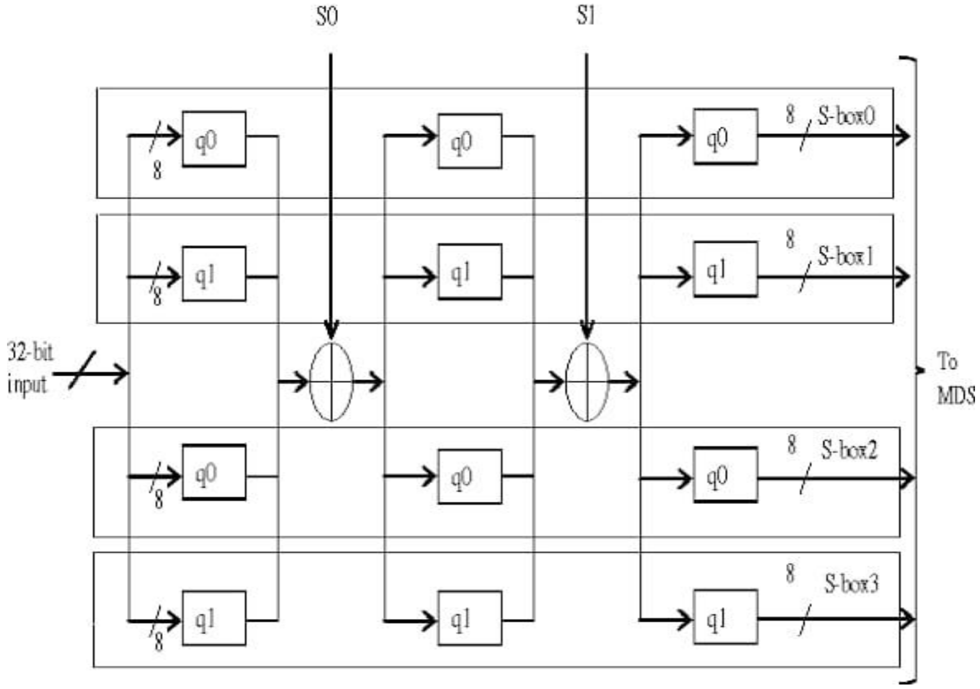


Figure 3.6 S boxes of the Twofish algorithm

MDS Matrices: A maximum distance separable (MDS) code over a field is a linear mapping from a field elements to b field elements, producing a composite vector of a + b elements, with the property that the minimum number of non-zero elements in any non-zero vector is at least b + 1. It is a matrix of bytes that multiplies a vector of four bytes. Multiplications are carried out in the Galois Field GF (2^8) with the primitive polynomial:

$$\boldsymbol{x^8 + x^6 + x^5 + x^3 + 1} \qquad 3\text{-}3$$

Pseudo-Hadamard Transforms: A pseudo-Hadamard transform (PHT) is a sim- ple mixing operation that runs quickly in software. Given two inputs, a and b, the 32-bit PHT is defined as:

$$\boldsymbol{a' = a + b \; mod \; 2^{32}} \qquad 3\text{-}4$$

$$\boldsymbol{b' = a + 2b \; mod \; 2^{32}} \qquad 3\text{-}5$$

Twofish uses a 32-bit PHT to mix the out- puts from its two parallel 32-bit g functions. This PHT can be executed in two opcodes on most mod- ern microprocessors, including the Pentium family.

Whitening: Whitening is a technique used to add an extra layer of security to the encryption process. In Twofish, the whitening step involves XORing the plaintext with a fixed value before the first round of encryption. The same value is XORed with the ciphertext after the last round of encryption, to get the final encrypted output.

Key Schedule: The key schedule is responsible for generating a series of round keys from the user-supplied key. Twofish uses a unique key scheduling algorithm that incorporates both the key material and the input block into the round key generation process. This helps to prevent attacks that exploit weaknesses in the key schedule.

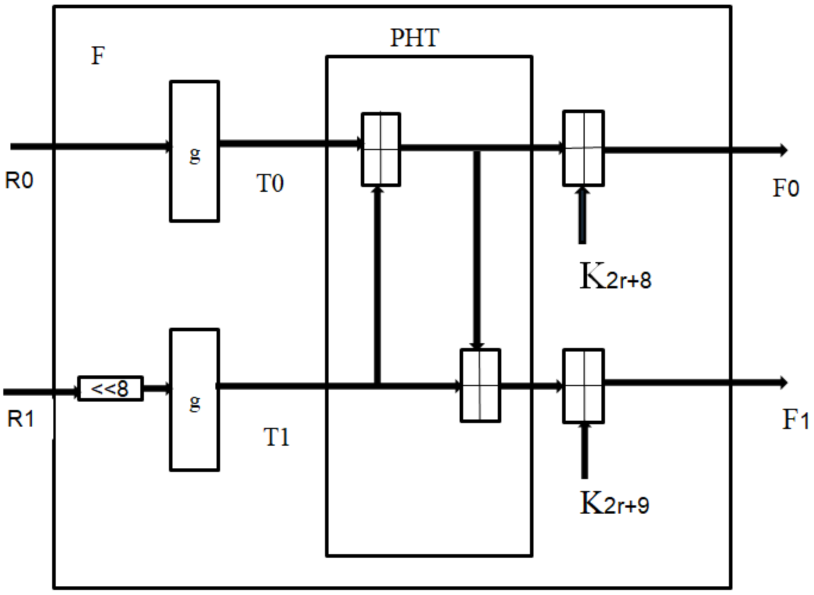


Figure 3.7 Structure of F function

Figure 3.7 shows the Inner structure of the F function, where g is the another function and T0,T1 will be the output of the g function and PHT is performed on the T0, T1 and key is added at this point and the result will be F0, F1.

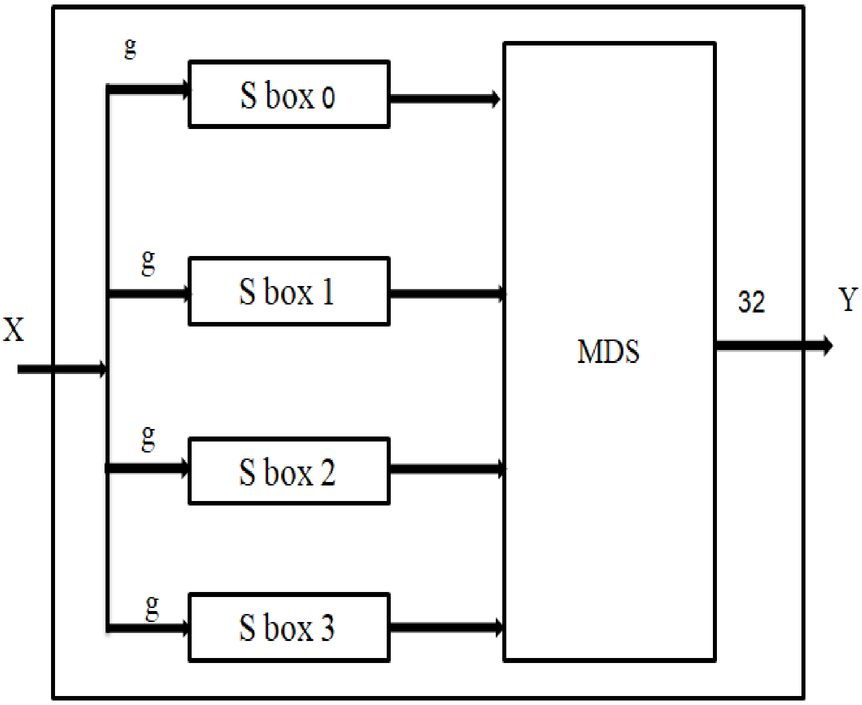


Figure 3.8 G function of the Twofish algorithm

Figure 3.8 shows the inner structure of the G function, here X is taken as input and divided into 4 parts of same size and substitute using S boxes individually and combine the output from four S boxes and taken as input to MDS matrices, let the result be Y

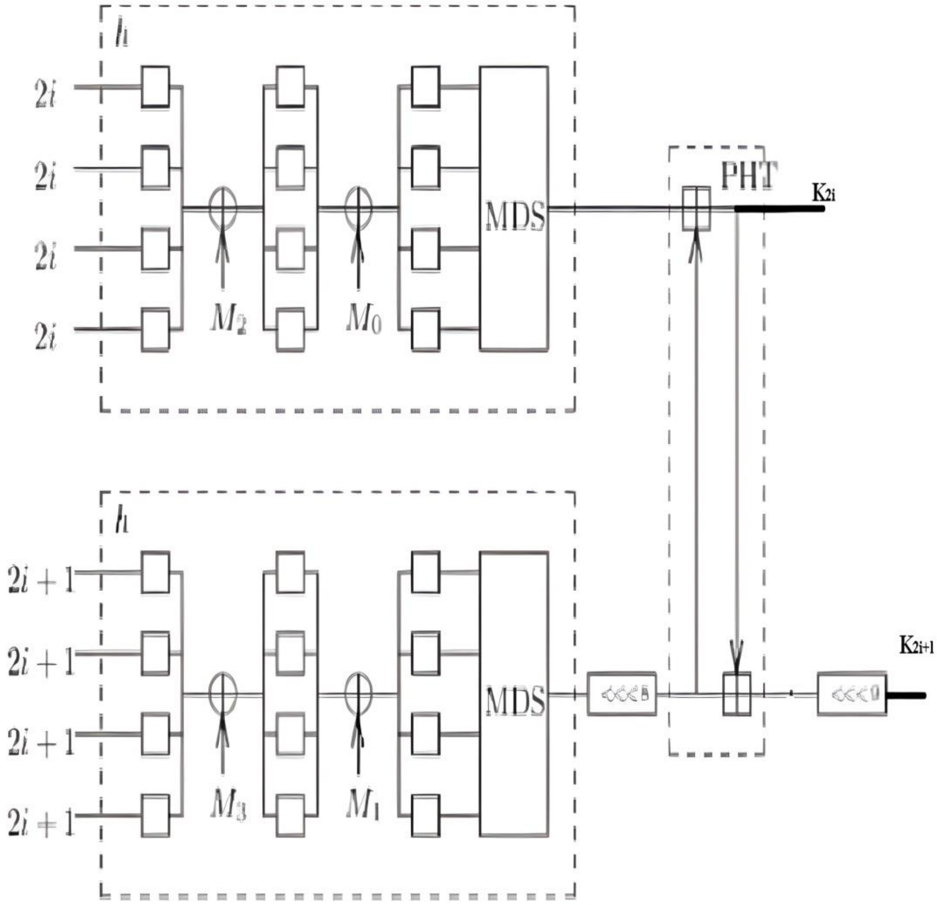


Figure 3.9 H function of the Twofish algorithm

Keys used in both F function and Whitening are obtained by using H function as shown in Figure 3.9

3.3.2.2 Features of Twofish Algorithm

The Twofish algorithm has several features that make it a highly secure and efficient encryption algorithm:

Security: Twofish is a highly secure encryption algorithm that is resistant to many types of attacks, including differential and linear cryptanalysis. It has undergone extensive testing and analysis, and its security has been validated by the cryptographic community.

Flexibility: Twofish supports key sizes of 128, 192, or 256 bits, making it highly flexible and adaptable to different security requirements.

Efficiency: Twofish is an efficient encryption algorithm that can be implemented in software or hardware. It is designed to be fast and efficient, without compromising its security.

Symmetric-Key: Twofish is a symmetric-key encryption algorithm, which means that the same key is used for both encryption and decryption. This makes it highly suitable for applications that require fast and efficient data encryption.

Block Cipher: Twofish is a block cipher algorithm that operates on fixed-length blocks of data. This makes it highly suitable for encrypting large amounts of data, as the data can be broken up into smaller blocks and encrypted separately.

Whitening: Twofish uses a whitening step that XORs the plaintext with two 32-bit values. This helps to reduce the correlation between the plaintext and the key, making it more difficult for an attacker to break the encryption.

Key Scheduling: Twofish uses a key scheduling algorithm that generates a set of round keys from the user-supplied key. This makes it more difficult for an attacker to recover the key from the encrypted data.

3.4 Software and Hardware Requirements

To run our application we need the following requirements

Software Requirements

1. Python 3.6 or above versions
2. MATLAB simulation tool.
3. Any Python IDE like Jupyter Notebook
4. Any Operating Sysytem

Hardware Requirements

1. A minimum of 4 GB RAM
2. Memory of minimum 512GB
3. Any dual core processor

4 Working Model of the Proposed System

This model is divided into two modules which run simultaneously, they are

1. Sender side
2. Receiver side

At first GPS coordinates of sender and receiver were taken as input in both sender side and receiver, which are useful in key generation process.

Let sender's latitude and longitude values be *15.83077*, *80.361505*

Let receiver's latitude and longitude values be *16.582804*, *80.525189*

Key used to encrypt the coordinates will be automatically generated at both sender and receiver side separately with the help of prime numbers which are saved in a list.

For this module we need to import random library. Here is the pseudo code for encrypting Coordinates which is same for both sender and receiver.

```
p = random.choice(primes)
q = random.choice(primes)
#primes is a list of prime numbers
while p == q:
    q = random.choice(primes)
print("Generating public/private key-pairs now . . .")
public, private = generate_key_pair(p, q)
lat = input(" - Enter Senders latitude : "))
lon = input(" - Enter Senders longitude : "))
message = lat+lon
encrypted_msg = rsa_encrypt(public, message)
print(" - Your encrypted location is: ",encrypted_msg)
```

4.1 Key Generation

RSA algorithm is used to encrypt the coordinates. Keys will be automatically generated by taking random prime numbers from the list of prime numbers.

Sender’s encrypted location for the above input is: “*654683136538715137353573136654686546857728*”.

Receiver’s encrypted location for the above input is: “*74903337921972177177490904990493050417986*”.

An XOR operation is performed between sender’s encrypted location and receiver’s encrypted location, the result will be the key to our model.

The Resultant key after performing XOR operation is: “*363432343331363531373532323832353632383334393738373330303330303036383433353232313134*”.

Here is the pseudo code for generating key to the Twofish algorithm using above encrypted location.

```
#At Sender Side
r_gps = input("Enter Receivers encrypted location: ")
key = (encrypted_msg)^(r_gps)
# Receiver side
s_gps = input(" - Enter Senders encrypted location: ")
key = (encrypted_msg)^(s_gps)
```

4.2 Encryption

A plain text is taken as input and it is converted into cipher text using Twofish algorithm by taking above key.

Let the plain text be “*Improve Security Using Steganography and Cryptography Based on Smartphone Users Locations*”

The Resultant cipher text after encrypting the secret text is: ‶*fb74b3b7ab29938099d7a86ec40b97acfcb805f3056af4579df35c273e8aeff30fe0661da4325bb38621a97543a31486a95063881c53d255b115e9c941f2ff2c3006fbd314d89ce09835e86b322d17550400bee8d3e918a9b675406c118c244a*″

Here is the pseudo code to encrypt the plain text using Twofish algorithm.

```
def encrypt(plaintext,key):
    # Making the required keys
    round_keys = key_schedule(key)
    white_keys = round_keys[:4]
    output_keys = round_keys[4:8]
    # Whitening the Input
    r1_array = whitening(plaintext,white_keys)
    r_array = []
    “““Convert the array to a 16 8-bit numbers from 4 32-bit
number”””
    # next looping 16 time for each round
    for r in range(16):
        # Calling F function
        f0,f1 = f_function(r_array,round_keys[2*r+8],
round_keys[2*r+9])
        c2 = f0^r_array[2]
        c2 = ROR(c2,1,32)
        r3 = r_array[3]
        c3 = ROL(r3,1,32)
        c3 = f1^c3
        r_array = [c2,c3,r_array[0],r_array[1]]
    # undo the steps
    r_array = [r_array[2], r_array[3], r_array[0], r_array[1]]
    # printing the output
    return(output)
s = input("Enter the plaintext : ")
cipher = ""
cipher += encrypt(plaintext,key)
print("The Ciphertext is : ",end=" ")
print(cipher)
```

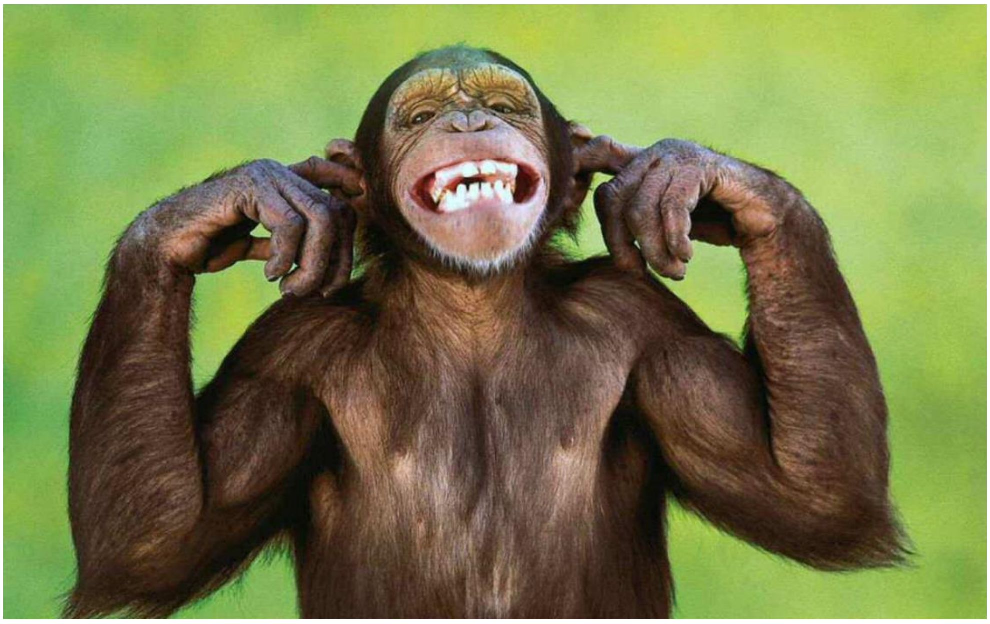


Figure 4.1 Cover Image

Let us take a cover image as input to embed above cipher text into that input image which is as shown in Figure 4.1, here is the pseudo code to embed cipher text into cover image for this we need to import stepic library and PIL library.

```
img = Image.open("Image.png")
cipher = chunkcipher.encode()
encoded_img = stepic.encode(img, cipher)
encoded_img.save("StegoImg.png")
#Embedding Cipher Text into Cover Image is Completed!
simg = Image.open("StegoImg.png")
plt.imshow(simg)
plt.show()
```

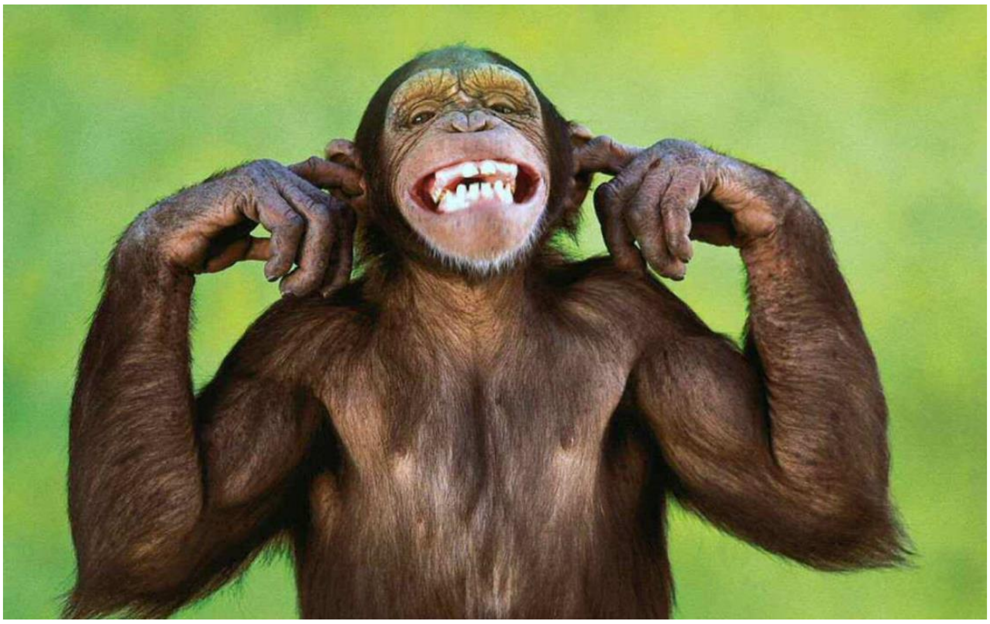


Figure 4.2 Stego Image

After Embedding the Cipher text the result will be the stego image which is as shown in Figure 4.2

4.3 Decryption

Let us take an image as input to extract cipher text from the stego image as shown in Figure 4.2 and the result will be the cipher text, here is the pseudo code to extract the cipher text from stego image for this we need the same libraries used embedding the secret text.

```
image=Image.open("StegoImg.png")
c = stepic.decode(image)
#Extracting Cipher Text from Stego Image
print("Decoded Message:",c)
```

The cipher text extracted from the above stego image is: “*fb74b3b7ab29938099d7a86ec40b97acfcb805f3056af4579df35c273e8aeff30fe0661d* *a4325bb38621a97543a31486a95063881c53d255b115e9c941f2ff2c3006fbd314d89ce 09835e86b322d17550400bee8d3e918a9b675406c118c244a*”

A cipher text is taken as input and it is converted into plain text using Twofish algorithm by taking above key.

The resultant plain text after converting from cipher text using Twofish algorithm is: “*Improve Security Using Steganography and Cryptography Based on Smartphone Users Locations*”.

Here is the pseudo code to decrypt the above cipher text using Twofish algorithm we use same libraries and functions used in Encrypting process.

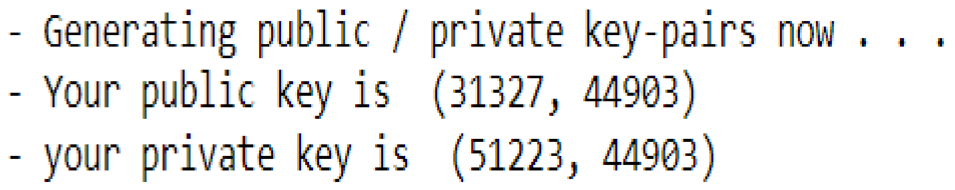
```
def decrypt(ciphertext,key):
    # Make the required keys with scheduling
    round_keys = key_schedule(key)
    white_keys = round_keys[:4]
    output_keys = round_keys[4:8]
    # Convert ciphertext to array of 16
    ciphertext = [ciphertext[i:i+8] for i in
                 range(0,len(ciphertext),8)]
    r_array = []
    # Adjust the little endian format
    # Ciphertext whitening with output whiten keys
    for j in range(len(output_keys)):
        r_array[j] = r_array[j]^output_keys[j]
    # Do the criss cross swapping in Fiestal cipher
    r_array=[r_array[2],r_array[3],r_array[0],r_array[1]]
    # Call the loop for 16 rounds
    for r in range(15,-1,-1):
        """ Reversing the states,the 3rd and 4th element will
be 1st and 2nd element of previous round state array """
        a = r_array[2]
        b = r_array[3]
        c2 = r_array[0]
        c3 = r_array[1]
        """Call the F function with the 3rd and 4th element"""
        f0,f1 = f_function([a,b], round_keys[2*r+8],
round_keys[2*r+9])
        """Reversing to get the r2 and r3 of previous round in
ecryption"""
        r2=ROL(c2,1,32)
        r2=r2^f0
        r3=f1^c3
        r3=ROR(r3,1,32)
        r_array=[a,b,r2,r3]
    """After 16 rounds ,whitening the array 3999933with input
whiten keys this """
    ans=""
    # Print the output in Big Endian format
    for i in r_array:
        tmp=[]
        for j in range(0,len(i),2):
            tmp.append(i[j:j+2])
        tmp=tmp[::-1]
        ans+=''.join(tmp)
    ans=ans.lstrip('0')
    return(ans)
plain = decrypt(Ciphertext,key)
print("The Decoded plaintext is : ",end=" ")
print(plain)
```

5 Experimental Results

Python was used to carry out the project on the Jupyter Notebook Platform. The sender's location is initially determined by the GPS of his or her mobile phone, and then the location is shared with the recipient through any social media application while it is being transmitted. The coordinates are encrypted using the RSA algorithm so that they will only be received by the recipient. The used key for the encryption algorithm is then produced by performing an XOR operation on the encrypted coordinates. The secret message will be encrypted using the Twofish algorithm and the key generated by the coordinates after being entered. Once the cover image has been read and the text has been embedded, the Stego Image is the outcome. The recipient will receive this Stego image.

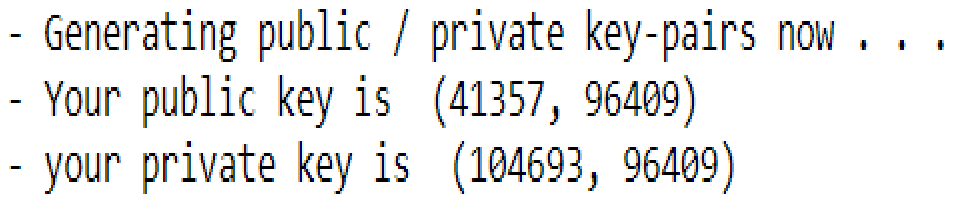
The secret text is produced when the receiver decrypts the stego image using the same key after reading the image and extracting the cipher text from it.

5.1 RSA Key Generation



```
- Generating public / private key-pairs now . . .
- Your public key is  (31327, 44903)
- your private key is  (51223, 44903)
```

Figure 5.1 Sender side RSA Key Generation

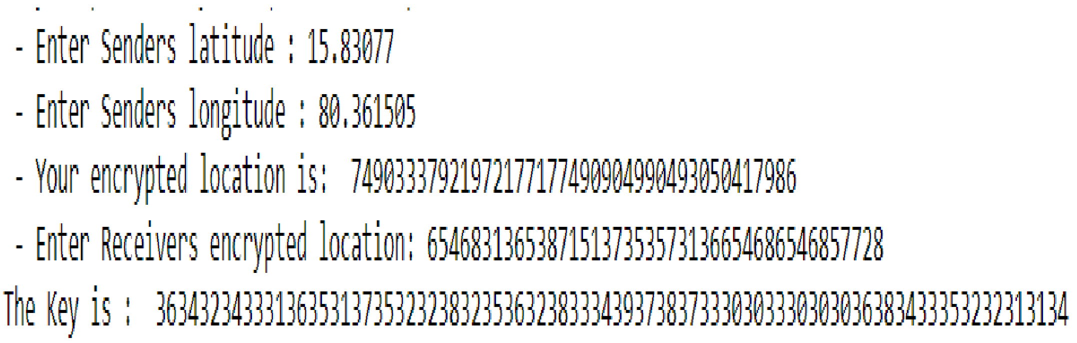


```
- Generating public / private key-pairs now . . .
- Your public key is  (41357, 96409)
- your private key is  (104693, 96409)
```

Figure 5.2 Receiver side RSA Key Generation

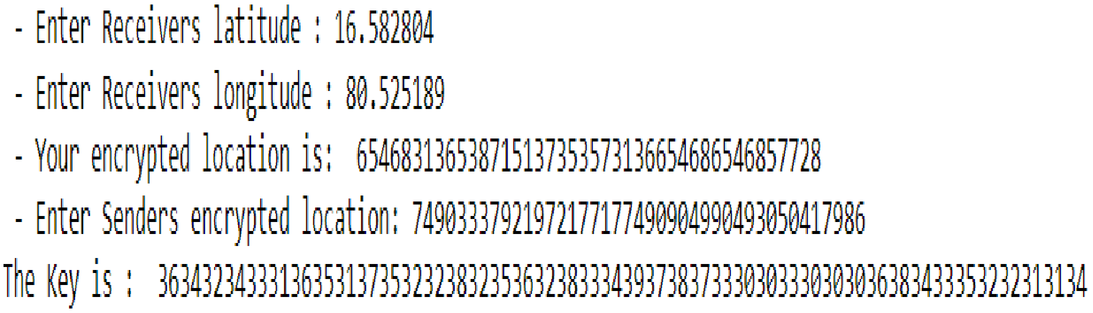
Figure 5.1 and Figure 5.2 shows the RSA key generation process, the public and private keys in above images are automatically generated in our application with the help of prime numbers which are saved in a list.

5.2 Twofish Key Generation



```
 - Enter Senders latitude : 15.83077
 - Enter Senders longitude : 80.361505
 - Your encrypted location is:  7490333792197217717749090499049305041 7986
 - Enter Receivers encrypted location: 6546831365387151373535731366546865468 57728
The Key is :  363432343331363531373532323832353632383334393738373330303330303036383433353232313134
```

Figure 5.3 Sender side Key Generation for Twofish



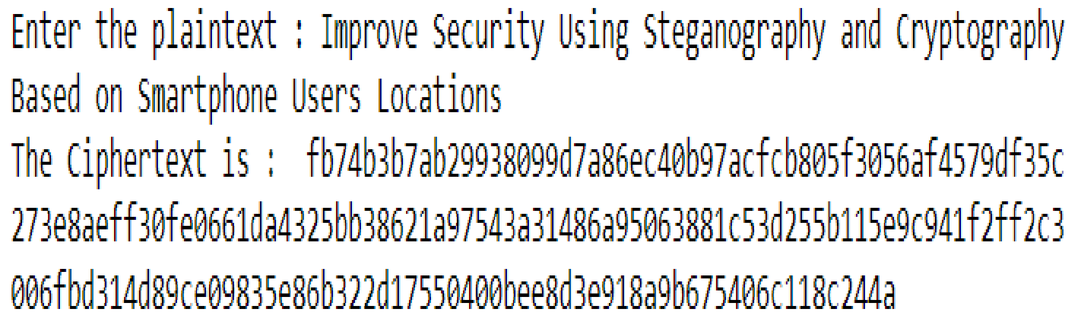
```
 - Enter Receivers latitude : 16.582804
 - Enter Receivers longitude : 80.525189
 - Your encrypted location is:  654683136538715137353573136654686546857728
 - Enter Senders encrypted location: 74903337921972177177490904990493050417986
The Key is :  363432343331363531373532323832353632383334393738373330303330303036383433353232313134
```

Figure 5.4 Receiver side Key Generation for Twofish

Coordinates are taken as input on both sender side and Receiver side and they are encrypted using RSA algorithm with their individual keys as shown in Figure 5.1 and Figure 5.2.

The Encrypted coordinates are shared in between sender and receiver and XOR operation is performed on following encrypted coordinates and the result will be the key, where the key is same on both sides as shown in Figure 5.3 and Figure 5.4.

5.3 Encrypting Secret Message



```
Enter the plaintext : Improve Security Using Steganography and Cryptography
Based on Smartphone Users Locations
The Ciphertext is :  fb74b3b7ab29938099d7a86ec40b97acfcb805f3056af4579df35c
273e8aeff30fe0661da4325bb38621a97543a31486a95063881c53d255b115e9c941f2ff2c3
006fbd314d89ce09835e86b322d17550400bee8d3e918a9b675406c118c244a
```

Figure 5.5 Encrypting Plain Text

At sender side a Plain text is taken as input and it is encrypted into cipher text using Twofish algorithm as shown in Figure 5.5

5.4 Embedding Cipher Text into Cover Image

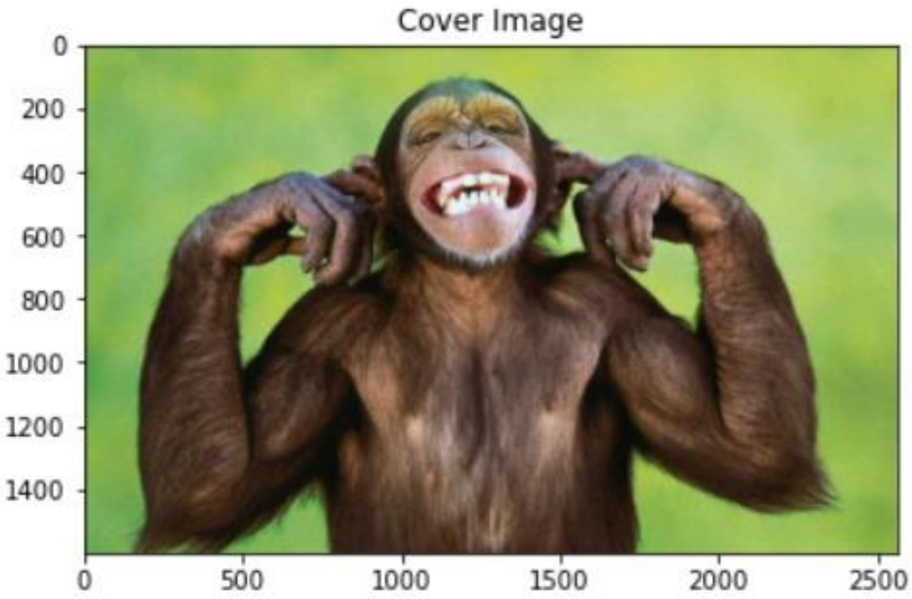
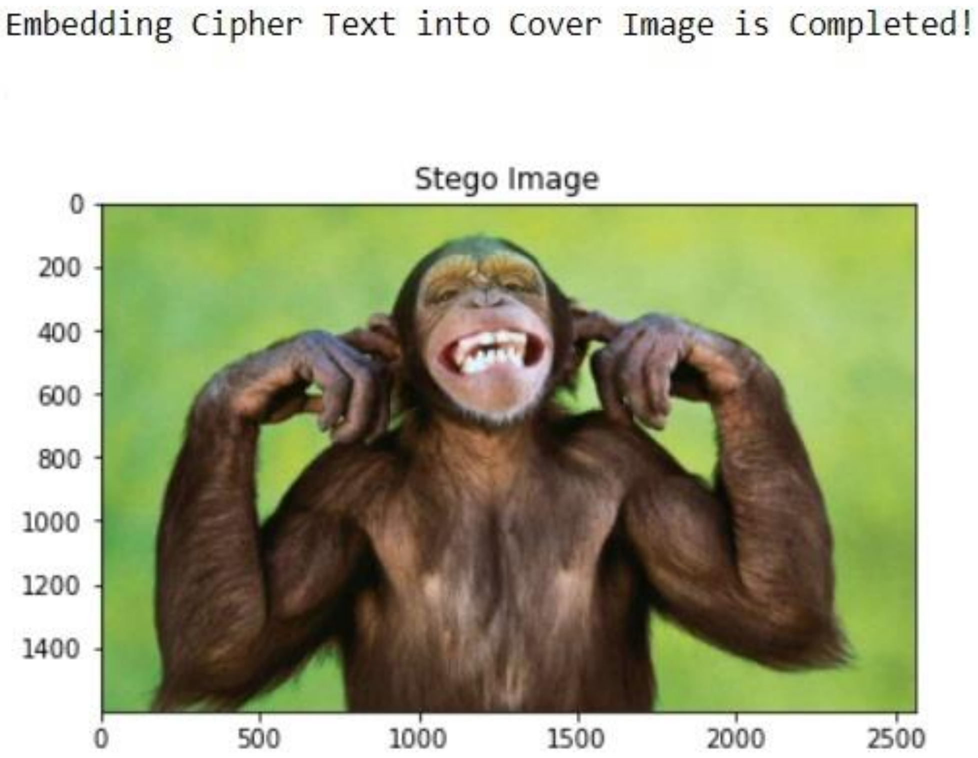


Figure 5.6 Cover Image

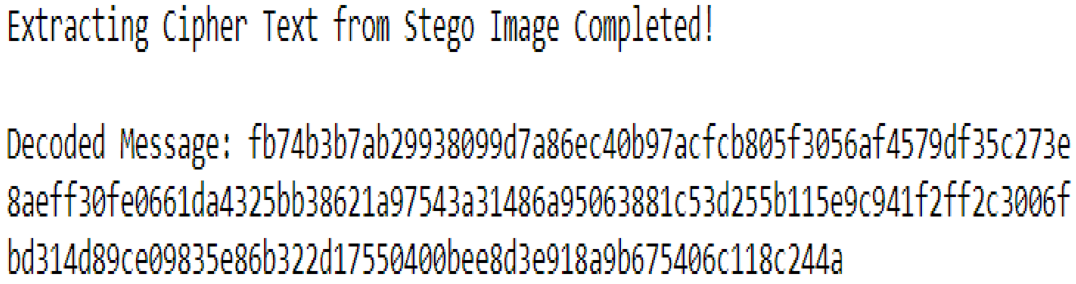


```
Embedding Cipher Text into Cover Image is Completed!
```

Figure 5.7 Stego Image

Figure 5.6 shows the Cover Image which is used to embed the cipher text into it and Figure 5.7 shows the Stego Image which means the image after embedding the cipher text.

5.5 Extracting Cipher Text from Stego Image



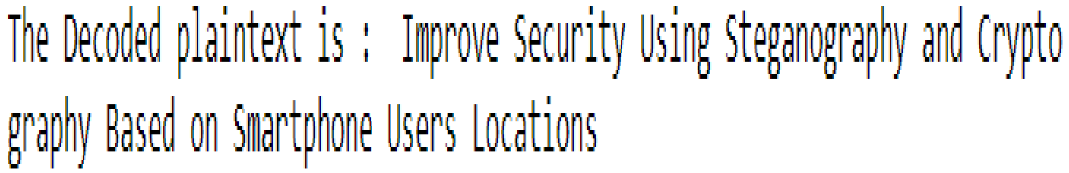
```
Extracting Cipher Text from Stego Image Completed!

Decoded Message: fb74b3b7ab29938099d7a86ec40b97acfcb805f3056af4579df35c273e
8aeff30fe0661da4325bb38621a97543a31486a95063881c53d255b115e9c941f2ff2c3006f
bd314d89ce09835e86b322d17550400bee8d3e918a9b675406c118c244a
```

Figure 5.8 Extracting Cipher text from Stego Image

At Receiver side the Stego Image which is sent by the Sender will be taken as input and Receiver will extract the cipher text from it as shown in Figure 5.8

5.6 Decrypting Cipher Text



```
The Decoded plaintext is :  Improve Security Using Steganography and Crypto
graphy Based on Smartphone Users Locations
```

Figure 5.9 Decrypting Cipher Text

At Receiver side the extracted cipher text as shown in Figure 5.9 is decrypted into plain text using Twofish algorithm.

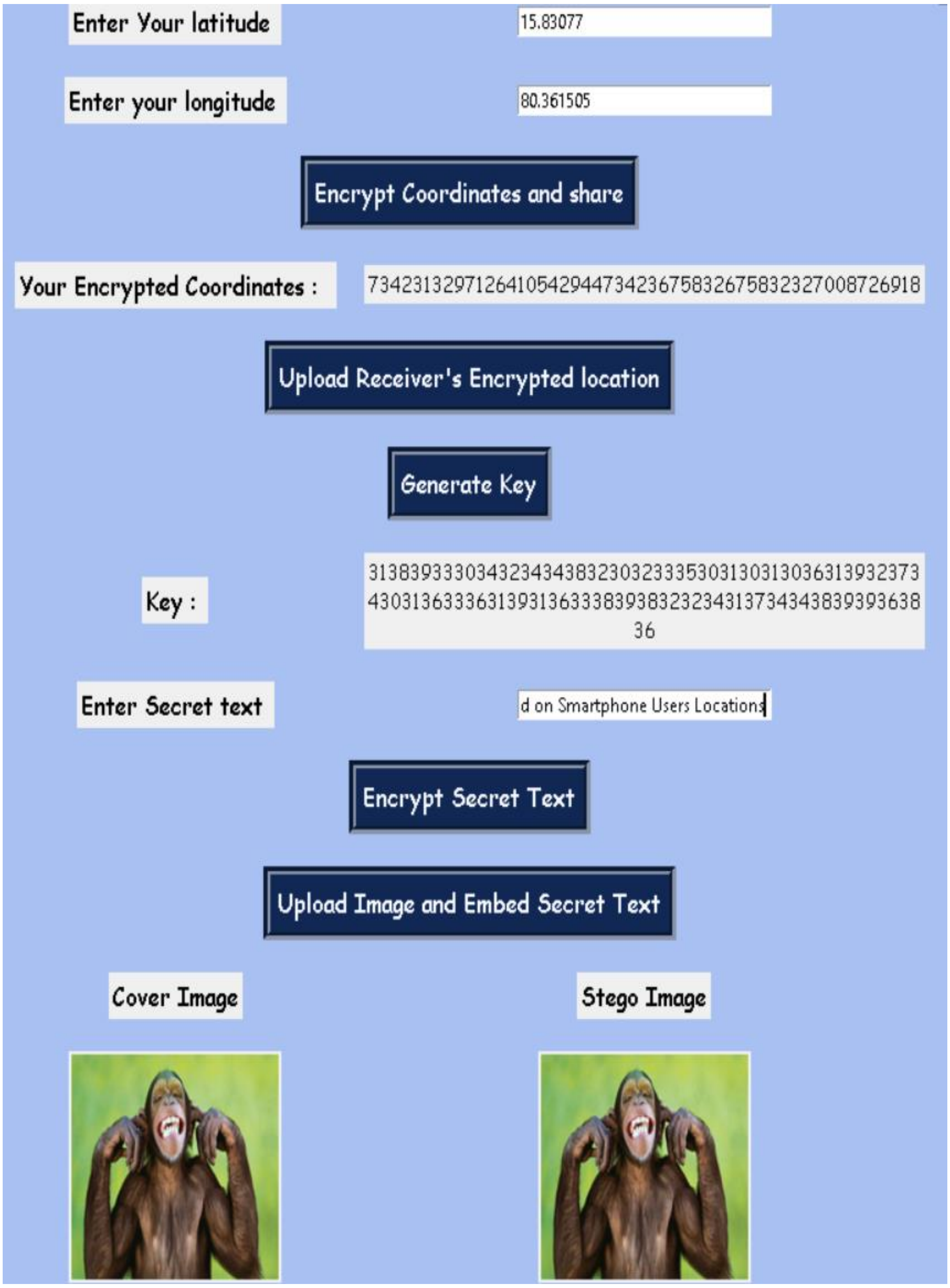
Up to now we have seen the results of different modules individually, Let us look into the total result with user interface. So that we will get an idea about the whole application. Process in user interface is same as above. Only difference is user interface provides a more user-friendly, efficient, and accessible way for users to interact with application than a console.

To provide interface we import tkinter, Tkinter is a standard Python library used for creating graphical user interfaces (GUIs) that allows developers to create windows, dialogs, buttons, menus, and other GUI components in Python code.

Here are some common uses of Tkinter:

1. **Creating standalone desktop applications**: Tkinter can be used to create standalone desktop applications with a graphical user interface. This is useful for creating applications that users can interact with directly on their computer.

2. **Developing scientific and engineering applications**: Tkinter is widely used in scientific and engineering applications because it provides an easy way to create graphical user interfaces for simulations and data visualization.



Enter Your latitude 15.83077

Enter your longitude 80.361505

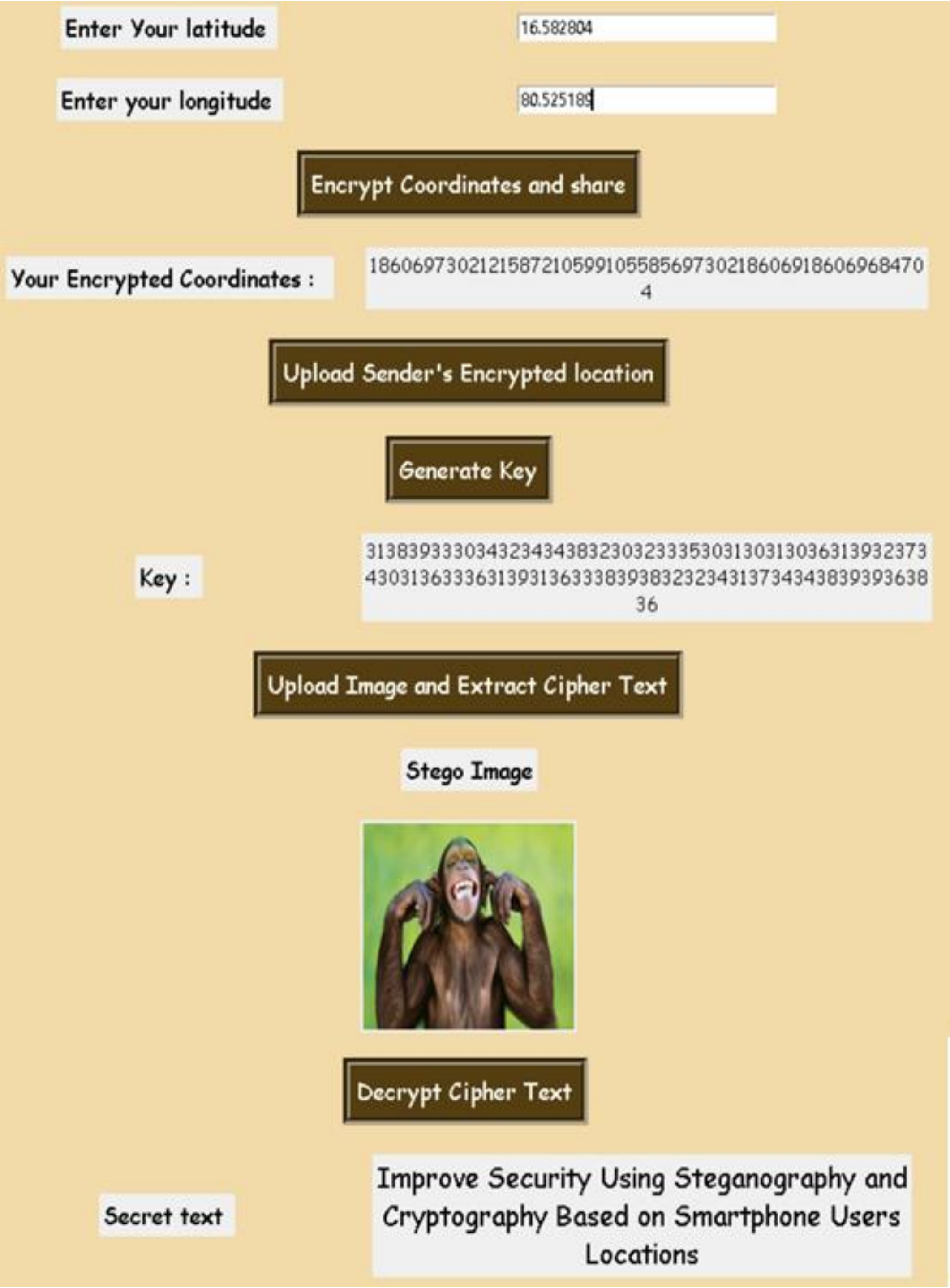
Your Encrypted Coordinates : 734231329712641054294473423675832675832327008726918

Key : 313839333034323434383230323335303130313036313932373430313633363139313633383938323234313734343839393638
36

Enter Secret text d on Smartphone Users Locations

Encrypt Coordinates and share
Upload Receiver's Encrypted location
Generate Key
Encrypt Secret Text
Upload Image and Embed Secret Text
Cover Image
Stego Image

Figure 5.10 Sender side Interface



Enter Your latitude 16.582804

Enter your longitude 80.525189

Your Encrypted Coordinates : 18606973021215872105991055856973021860691860696847
04

Key : 31383933303432343438323032333530313031303631393237
34303136333631393136333839383232343137343438393936
3836

Stego Image

Secret text Improve Security Using Steganography and Cryptography Based on Smartphone Users Locations

Encrypt Coordinates and share
Upload Sender's Encrypted location
Generate Key
Upload Image and Extract Cipher Text
Decrypt Cipher Text

Figure 5.11 Receiver side Interface

Using the MATLAB simulation tool, the results of the proposed system were obtained. Mean square error (MSE), peak signal to noise ratio (PSNR), and structural similarity index measure (SSIM) are some of the performance metrics used to quantify the proposed work. The following is performance metrics:

Mean-squared error (MSE): The differences between the pixels of two images (cover image (CI) and stego- image (SI)). The MSE equation is:

$$\boldsymbol{MSE = \frac{1}{mn}\sum\nolimits_{I=1}^{n}[CI(k,I) - SI(k,I)]^2} \tag{5-1}$$

The image size is represented by m and n.

Peak signal-to-noise ratio (PSNR): It is an equation for the ratio of a signal's greatest possible value (power) to the power of distorting noise that influences the quality of its representation, which is used to compare two images' ratios. The PSNR equation is:

$$\boldsymbol{PSNR = 10\ log10\ \frac{RI^2}{MSE}} \tag{5-2}$$

The maximum value of the image's pixels is represented by RI.

Structural information change (SSIM): SSIM is a method for determining the likeness between two images that improve on the traditional PSNR and MSE methods. The SSIM index's range value is [0,1]. If the index is high, it suggests the two images (cover image (CI) and stego-image (SI) which are more similar, and it is computed as follows:

$$\boldsymbol{SSIM(x1,y1) = \frac{(2\mu_{x1}\mu_{y1} + C_1)(2\sigma_{x1y1} + C_2)}{(\mu_{x1}^2 + \mu_{y1}^2 + C_1)(\sigma_{x1}^2 + \sigma_{y1}^2 + C_2)}} \tag{5-3}$$

Table 5.1 Results Of Embedding 1500-Bit Text In The Cover Image

Images	Pixels	MSE	PSNR	SSIM
	9,025 KB	0.0002	85.5839	1.0000
	7,089 KB	0.0002	86.3670	1.0000
	10,804 KB	0.0002	85.2803	1.0000
	8,138 KB	0.0002	85.6653	1.0000
	6,567 KB	0.0002	85.5879	1.0000
	7,536 KB	0.0002	85.5218	1.0000

Table 5.2 Results Of Embedding 3300-Bit Text In The Cover Image

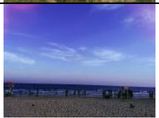
Images	Pixels	MSE	PSNR	SSIM
	9,025 KB	0.0003	83.8354	1.0000
	7,089 KB	0.0002	84.6181	1.0000
	10,804 KB	0.0003	83.5140	1.0000
	8,138 KB	0.0003	83.8713	1.0000
	6,567 KB	0.0003	83.8341	1.0000
	7,536 KB	0.0003	83.8132	1.0000

Table 5.1 represents the results of performing the process of encrypting the secret text with a size of 1500 bits and then it was embedded in several images of different sizes. the best value of the MSE amounted to 0.0002 and the PSNR the best value reached 86.3670, while the SSIM had the result equal to 1.0000 in all images.

While Table 5.2 represents the results of performing the process of encrypting the secret text but with a size of 3300 bits and then it was embedded in several images of different sizes. The best value of the MSE amounted to 0.0002 and the PSNR the best value reached 84.6181, while the SSIM had the result equal to 1.0000 in all images. From the following two tables, notice that when you increase the size of the confidential data that will be included in the cover image, the values of MSE will be decreased. In general, in both tables, the results were good.

6 Conclusion and Future Work

This model accomplished excellent security by utilizing GPS technology to collect the sender and recipient's locations, as well as an equation between the two positions to obtain the algorithm key. The system has achieved enhanced security for the embedded confidential data by merging both encryption and steganography techniques through using the proposed approach for embedding the encrypted text in the cover image. The proposed approach for embedding text is quite efficient, as see when the text with a size of 1500 bits is encrypting and then it is embedded in multiple images of various sizes, as the MSE was improved. In all photos, the best value of the MSE was 0.0002, and the best value of the PSNR was 86.3670, while the SSIM had the result of 1.0000. Found that the suggested system addressed the major concerns about safeguarding information received over mobile networks, resulting in a high level of security.

The following suggestions may be considered for future studies: applying the proposed method to embed confidential data in other multimedia such as audio, and video. Applying the proposed method to embed confidential data in the GIS file. Making the application embed other types of data such as images, video, and audio.

7 Bibliography

[1] L You, Y Chen, B Yan, and M Zhan, "A novel location-based encryption model using fuzzy vault scheme," *Soft Comput.*, vol. 22, no. 10, pp. 3383-3393, 2018.

[2] N N Mohamad, H Othman, M A M Isa, N A M Noor, and H Hashim, "A secure communication in location based services using aes256 encryption scheme," *in 2017 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, pp. 163-167, 2017.

[3] P S Gaikwad, P Dalvi, M Patel, C Dhalpe, and A Chaudhari, "MOBILE APPLICATION FOR PROVIDING SECURITY TO DATA TRANSMISSION".

[4] M Sen, S Tibrewal, S Majumder, N Ahmad, and K K Singh, "Location dependent cryptographic algorithm based on open source application," *Int. J. Comput. Appl.*, vol. 975, p. 8887, 2017.

[5] G Sriram, B Srikanthreddy, K V Seshadri, K Hemantth Kumar, and N Suresh, "Location based encryption-decryption system for android," *Proceedings of the International Conference on Smart Systems and Inventive Technology.*, pp. 590-593, 2018.

[6] N S M Shamsuddin and S A Pitchay, "Implementing location- based cryptography on mobile application design to secure data in cloud storage," , vol. 1551, 2020, p. 12008.

[7] R F S Lizy, "Improvement of RSA Algorithm Using Euclidean Technique," *Turkish J. Comput. Math. Educ.*, vol. 12, no. 3, pp. 4694-4700, 2021.

[8] A Devi and B S Ramya, "Two fish Algorithm Implementation for lab to provide data security with predictive analysis," *Int. Res. J. Eng. Technol.*, vol. 4, no. 5, pp. 3033-3036, 2017.

[9] S M Kareem and A M S Rahma, "A novel approach for the development of the Twofish algorithm based on multi-level key space," *J. Inf. Secur. Appl.*, vol. 50,

p. 102410, 2020.

[10] G Dhamodharan, S Thaddeus, L C Flores, J L Hilario-Rivas, and F Sandoya, "Embedding Elliptic Curve Cryptography and Twofish Algorithm to Improve Data Security in Internet of Things," *Adv. Mech.*, vol. 9, no. 3, pp. 971-978, 2021.