DATA HIDING USING STEGANOGRAPHY AND CRYPTOGRAPHY

A Project Report

Submitted in the partial fulfillment of the requirements for
the award of the degree of

Bachelor of Technology
in
Department of Computer Science and Engineering

By

P DEEKSHITHA 180030028

T NAGA MADHURI 180030290

Under the supervision of

Mr. R.M. BALAJEE

(Asst. Professor)

Department of Computer Science and Engineering

K L E F, Green Fields,

Vaddeswaram- 522502, Guntur(Dist), Andhra Pradesh, India.

2021- 2022

DECLARATION

The Project Report entitled **"DATA HIDING USING STEGANOGRAPHY AND CRYPTOGRAPHY"** is a record of bonafide work of **P.DEEKSHITHA 180030028, T. NAGA MADHURI 180030290** submitted in partial fulfillment for the award of B.Tech in DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING to the K L University. The results embodied in this report have not been copied from any other departments/University/Institute.

Signature of Students

P.DEEKSHITHA 180030028

T. NAGA MADHURI 180030290

CERTIFICATE

This is to certify that the Project Report entitled "**DATA HIDING USING STEGANOGRAPHY AND CRYPTOGRAPHY**" is being submitted by P. DEEKSHITHA 180030028, T. NAGA MADHURI 180030290 in partial fulfilment for the award of B.Tech in Computer Science and Engineering to the KL University is a record of bonafide work carried under our guidance and supervision.

The results embodied in this report have not been copied from any other departments/University/Institute.

Signature of Supervisor

Mr. R.M. BALAJEE

Signature of the HOD

Signature of External Examiner

ACKNOWLEDGEMENTS

Our sincere thanks to **Mr.R.M.BALAJEE** in the research for her outstanding support throughout the project for the successful completion of the work.

We express our gratitude to **Mr.R.M.BALAJEE** Faculty for the project course in Computer Science and Engineering Department for providing us with adequate planning and support and the means by which we can complete this.

We express our gratitude to **Mr. V. HARIKIRAN**, Head of the Department for computer science and Engineering for providing us with adequate facilities, ways, and means by which we can complete this Research work.

We would like to place on record the deep sense of gratitude to the honorable Vice-Chancellor, K L University for providing the necessary facilities to carry the work.

Last but not the least, we thank all the Teaching and Non-Teaching Staff of our department and especially our classmates and our friends for their support in the completion of our work.

NAME	ID.NUMBER
P.DEEKSHITHA	180030028
T. NAGA MADHURI	180030290

ABSTRACT

The increasing Internet connectedness of computers and networks, as well as users' reliance on network-enabled services, has increased the quantity and sophistication of attack techniques, as well as the simplicity with which an assault may be launched. In order to avoid that, two parties communicating need to communicate in such a way that it is more secure and no one can modify it. So big organizations use many encryption techniques to secure communication. In order to secure the information Cryptography and steganography are widely used. By merging these techniques security level increases. This project aims to merge the LSB and AES,RSA algorithms for data encryption. By Hiding data into an image the attacker have a less chance to recover data because before hiding data into an image we first encrypt it using two encryption techniques which is more secure.

TABLE OF CONTENTS

CHAPTER-1 INTRODUCTION

1.1 Why Encryption?

Sending and receiving information via email or the use of web browsers are not secure as sensitive information such as credit card and some personal and important information sent over such medium can be hacked or attacked.We need a system to secure information which is being attacked through online. Encryption is needed to avoid such kind of attacks in which the user probably can secure the most important and personal information such as account numbers and passs where as coming to large organizations they need encryption for hiding the ip address of the website and securing the data. As we can see nowadays just a change in the domain or a small change in the ip address leads us to go to an unethical or a wrong website but the original and wrong website both seems similar so anyone cannot identify which is the real one. Inorder to avoid that whatever the elements related to networking like ip address and our information should be encrypted and kept safe.

1.2 Importance:

Sending and receiving most confidential and personal data through internet is likely to be attacked by unauthorized parties. The number of attacking cases increases which make the need for data security important. There are many encryption techniques. Steganography and cryptography are two alternative approaches for keeping data safe and secure**.** The purpose of Steganography is to hide secret and confidential messages in digital media sources such as text or video or image files. Cryptography is a technique in which the text or inputted data will be changed and no one can understand the meaning of it.

1.2.1 Steganography

- Image Steganography is a technique that finds applications in many fields, for purposes like data hiding or storing confidential data.
- Storing data in an image will always be more secure because the chance of guessing that data may be stored in an image file was low.
- Although this encryption technology was one of the most secure, it was not extensively employed.
- One of the most secure methods for concealing information is steganography. It is more secure since information is buried in an image, making it impossible to distinguish between a regular image and an encrypted image.

- The science and art of hiding information by embedding messages in seemingly benign items. It works by substituting bits of distinct and invisible data with specified pieces of useless or unused data in typical computer files (for example, text, HTML, audio, or photos).
- Both images seems similar and an attacker cannot identify if at all the attacker identifies it is most difficult to crack the data which is hidden.
- The LSB steganography approach is more secure than other steganography techniques.
- A message can be hidden in a variety of ways. Some bytes in a file or picture are unnecessary and can be replaced with a message without damaging the original message when it is created. The secret message is hidden in this way. There are several varieties of steganography, such as steganography in video, audio, and other media.

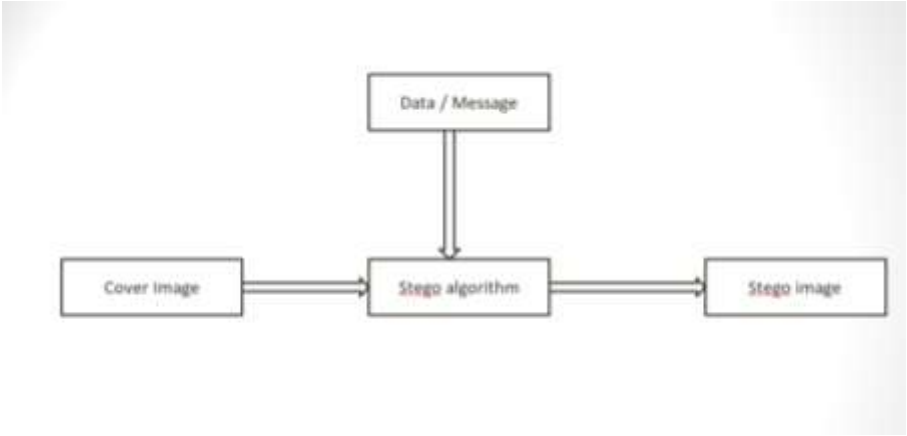


Fig.1.1:Steganography Block diagram

- The steganography techniques utilised aid in the best possible concealment of the message, ensuring that it is exposed only at the destination. The techniques used are:
- **Least Significant Bit(LSB):** The attacker finds the least important bits of data in the carrier file and replaces them with the secret message, which is usually harmful code. When the victim downloads the file, malware is installed on the computer, allowing the hacker or attacker to get access to the device. Sandboxes are used to detect faulty files, but hackers have devised ways to get around them, such as sleep patching. Sandbox does not identify sleep patched malware because it is innocuous and takes time to detect.
- **Palette Technique:** The attackers encrypt the message and then conceal it in a wide palette of the cover image using digital images as malware carriers. Although it can only carry a limited quantity of data, cybersecurity professionals are frustrated since the data is encrypted and takes time to decrypt.
- **Secure Cover Selection**:Cybercriminals must compare blocks of the carrier picture to specific blocks of specific virus, which is a fairly sophisticated approach. It entails finding the ideal candidate to carry the infectionThe identical match is precisely matched to the carrier image.

Because the generated image is identical to the original, software programmes and cybersecurity tools have a harder time detecting it.

1.2.2 Cryptography

- Cryptography is the process of securing and safeguarding data so that only the intended users have access to it.
- It can secure messages that pass via untrustworthy networks. An adversary may attempt to carry out one of two types of assaults on a network.
- An attacker using passive assaults simply listens on a network segment and tries to read sensitive data as it travels. Passive attacks can be carried out online or offline (in which an attacker just collects traffic in real time and examines it later—possibly after decrypting it). An attacker impersonates a client or server, intercepts communications in transit, and views and/or modifies the contents before passing them on to the intended recipient.
- The most essential thing to keep in mind is that you should never attempt to build your own cryptosystem. The world's most intelligent cryptographers (Phil Zimmerman and Ron Rivest, for example) often produce cryptosystems with major security problems.
- To be certified "safe," a cryptosystem must be subjected to rigorous testing by the security community. Never put your security in the hands of obscurity or the possibility that attackers are unaware of your system. Remember that your system will be attacked by malicious insiders and determined attackers.
- Cryptography in Modern Times

  Cyber cryptography techniques are being used to send electronic data over the internet in such a way that no third party can read it. Four factors are used to assess the code's strength:

  1. Non-disclosure

  This refers to how many individuals outside of the two persons involved in the discussion can grasp the information being delivered. If more persons can view the files, it indicates that the communication method is insecure.

  2. Reliability

  This refers to how quickly information may be changed while being transported from one location to another without the sender or receiver being aware of it.

  3. The principle of non-repudiation

  Whether or not the originator of the piece of communication will be able to subsequently refute the motivations for developing the message or its manner of dissemination.

4. Verification

Both the transmitter and recipient should be able to verify each other's identities as well as the source of the communicated data. This is an important initial step in determining the integrity of the sent file.

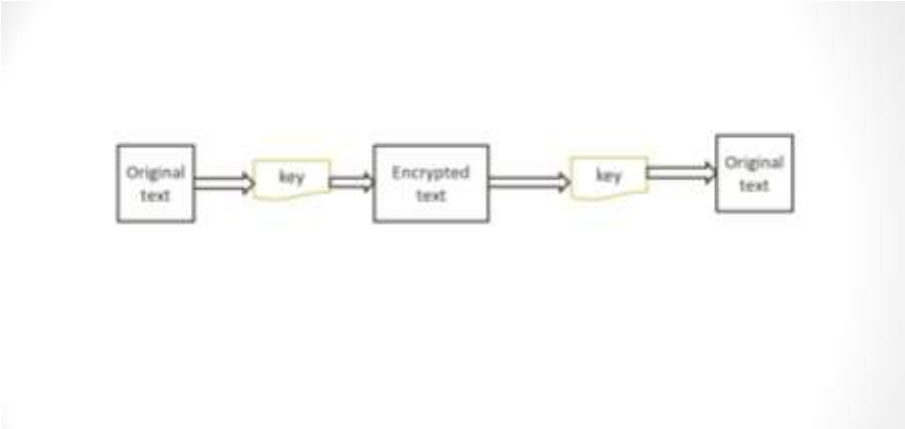


Fig.1.2:Cryptography Block Diagram

- There are 3 types of cryptography:
- **Symmetric Cryptography:** Never trust your security on obscurity or the likelihood that an adversary is ignorant of your system.
- The AES encryption technique, which employs a single key for both encryption and decryption, is an example of symmetric cryptography.

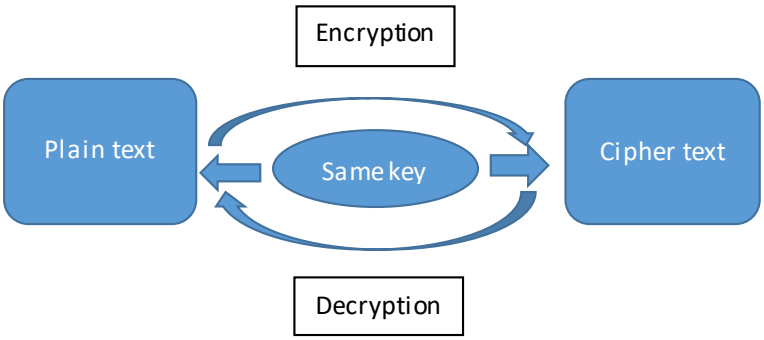


Fig.1.3:Symmetric Cryptography

- AES stands for **Advanced Encryption Standard** which is one best cryptography technique that provides users a safety communication via internet.
- AES is a proved experimentally one of the best and most secure cryptographic techniques.

- AES is more better than 3DES,DES,RSA and all the other cryptograeulc algorithms.The main key factor which is deciding that AES is far better than all the other is the key lengths because AES allows 128,192,256 bit keys than 56 bit key which DES allows which makes users feel more secure to use than any other cryptograeulc algorithm does.
- **Asymmetric Cryptography:** Because it employs two keys, one for encryption and the other for decryption, this is also known as public key cryptography.
- The sender will have the public key, while the receiver will have the private key.

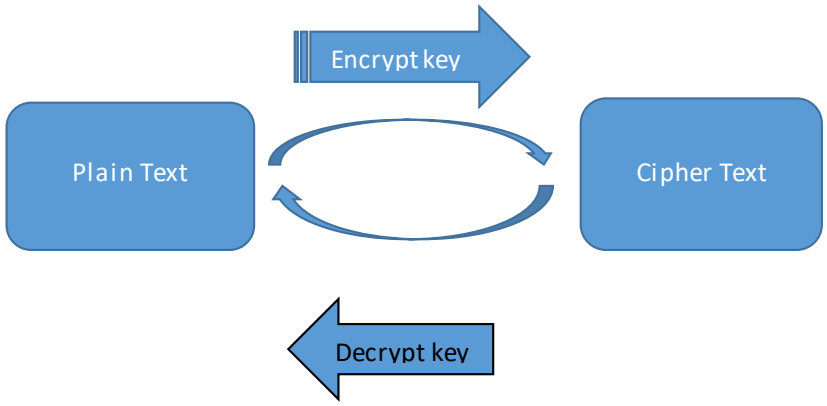


Fig.1.4:Assymetric Cryptography

- The RSA Algorithm is an example of asymmetric cryptography that employs both public and private keys.
- RSA stands for Rivest**, Shamir, Adleman** which is also a cryptograeulc technique which make sures that to establish secure connections between VPN clients and VPN servers.
- As Encryption power depends on size of key as we said before here also the same happens i.e RSA also makes sure that the length of the key is big so that the encryption is more and more secure and no one cannot decrypt it.
- To achieve secrecy, public key algorithms can require substantially larger keys. Whereas brute force is commonly used to break symmetric keys, public key techniques entail obtaining the matching secret key from the public key. Depending on the sort of event, the method varies.
- As we know that RSA is hard to break because we are using 2 different types of keys so only the authenticated person can view the message with the key so we are using this cryptograeulc algorithm inorder to get authentication.
- We have used 2 different types of cryptography algorithms which are faster and provide best and strong security than other algorithms.
- **Hash Functions-Functions having hash value:** A cryptographic hash function's fundamental functioning does not necessitate the usage of keys. This method uses a one-way procedure to

generate a tiny digest or "hash value" from often massive volumes of data. Hash functions are often used to provide the foundation for key management and security services such as encryption.

- We may provide source and integrity authentication services by establishing message authentication codes (MACs).
- Digital signature creation and verification using message compression.
- In key-establishment algorithms, obtaining keys.
- Creating random numbers that are deterministic.
- The result of plaintext or ciphertext is a hash value. Hashing is a cryptographic method for converting any type of data into a unique text string.
- There is a predictable output for each given input. In basic terms, putting plaintext through a hashing algorithm yields the same result. Assume you alter the input or plaintext to the hashing algorithm in any way. In addition, the hashing output changes.
- Hashing transforms a readable text into an unreadable safe data text.
- Hashing is well-executed, but it's exceedingly tough to undo. As I previously indicated, hashing and encryption are frequently confused. Encryption is a two-sided process. The plaintext can be used.
- People can receive data authorisation without understanding the data's substance thanks to hashing. Passwords are stored using hashing techniques and databases.
- Instead than being kept in plaintext, passwords are saved as a hash value or a hash password. The data is more secure because of the hash value.

- Potential attackers are deterred by cryptographic hashing. If a bad individual tries to access the database, the hashes will be visible. The attacker, on the other hand, is unable to convert the hash value back to the original password.
- **The goal of hashing is to:**

  To ensure that the data is accurate.

  Authentication.

  To keep track of sensitive information.

1.3 Benefits

- We have used 2 different types of cryptography algorithms which are more faster and provide best and strong security than other algorithms.
- Highly secure message transfer.

- Even image was decrypted still data is confidential since we induced double encrypted message.
- When compared to the existing scheme proposed system provides more security as it is providing triple layer of security.
- It allows the use of digital certificates and timestamps, which is an extremely safe signature authorization method. We'll look at digital timestamps and digital signatures in a moment.
- In the domain of cryptography, quantum cryptography and DNA cryptography are two promising approaches. Quantum cryptography uses photons, or light packets, to attempt to attain the same level of information security as previous forms of cryptography.
- The method, which is still in the experimental stages, takes advantage of light's polarisation properties and is proving to be a very promising anti-eavesdropping defence.
- DNA cryptography employs a set of carefully chosen DNA strands, the combination of which yields a precise answer to a problem.
- Using steganography for this system has the following advantages:
- One-Way Hashing: Used to verify that a message has not been tampered with by a third party. This is performed by hashing the message and assigning a constant character length to each item in the message, not withstanding the fact that the original components had varying lengths of characters. The message's hash is encrypted and sent with it. When the communication reaches the designated recipient, it is decrypted. If the hash from the decoded message does not match the hash from the encrypted message, both the sender and the recipient of the communication will be aware that it has been tampered with.
- Adding Explanatory Notes to an Image: - Explanatory notes are added to an image. This could be used in the medical field where one medical office.
- If the sending medical office has to provide notes explaining what the receiving medical office should be focusing on, this is the place to do it.
- Steganography may also be used to prevent the theft of people's identities and important data.
- Unauthorized perusal or even sabotage by enclosing the communication in a shady picture.

1.4 Drawbacks:

- The global application of this might result in a significant increase in unemployment.

- The polarization of photons can vary while moving through the channel (i.e. optical fibre or air) owing to a variety of factors.
- Many important elements of quantum cryptography are missing, such as digital signatures and certified mail.
- Between the source and the destination, a dedicated channel is required, which comes at a hefty cost. Multiplexing is against quantum principles, hence sending keys to two or more separate places over a

  quantum channel is impossible. This necessitates the use of distinct channels between the source and the many destinations. This is a significant drawback of quantum communication.

1.5 Project Goal:

The goal of the project was to:

- To establish a secure transformation of message content.
- To avoid sensitive data leakage.
- To make use of encryption algorithms.
- Provide triple encryption for the data.

CHAPTER-2 LITERATURE SURVEY

Data transmitted via network was not secure since there is a chance for attackers to steal, modify or delete data. Hence we need to make sure that the data which we sent must be received to the authorized receiver and to achieve this encryption was the only way[1]. To encrypt the data there are many encryption techniques each technique is unique from the others some techniques are AES, 3AES, RSA, SHA256, and many. We utilise the key, which is extra information we provide to make safe encryption, to encrypt data. There are two types of keys: public keys and private keys. As the names suggest, public keys are known by everyone, while private keys are only known by authorised users[2]. So by following any one of the encryption techniques we can achieve the integrity goal.

To encrypt RSA is one of the best algorithms since it has advantages like time of execution is less, chances of attacking are high, and most secure algorithm[3]. Similar to RSA there is another technique that is more secure than it is AES. This AES is a technique that encrypts the original text and sends it to NAND flash for storage this takes less time for execution. This algorithm has a unique structure to encrypt and decrypt sensitive information and is applied in hardware and software all over the world[4]. Steganography is also an encryption technique that is not used widely, the data is encrypted into stego files, and those files can be either image or audio, or text files. As the data is being embedded into files attacker can not guess that there is information hidden in that file and also as it was not used widely we attacker also finds it difficult to decrypt the information[5]. Image steganography is a type of steganography in which data is hidden behind pictures. Storing data inside image pixels is known as the LSB technique which later when encrypted it still displays the same picture similar to the original picture, this helps to enhance the security level as both original and encrypted images are indifferent[6].

Through encrypting the information using any one of the algorithms makes sure that our data transferred via a medium is secure. The current dependence of users on the network to send and receive messages was drastically raised due to lack of time and impatience and growth in technology. As all are using networks to communicate and encrypt their messages using some encryption algorithm the attackers also learned ways to decrypt the encrypted content. Attackers learned new ways to steal information from authorized users and again there is a threat to integrity, availability. To make information more secure and attack resistant encrypting the data or information twice was very helpful[7]. So one such combination is the combination of steganography with Blowfish-RC4 encryption technique. Through this combination attempt, we can make sure that double encryption is attained. Firstly the data was encrypted using the Blowfish-RC4 algorithm so data is encrypted and that encrypted data is stored into pixels of an image file i.e LSB.By merging the chances of information leakage are decreased because even attackers decrypt the message and obtain the data it is still encrypted data they receive as that data

embedded into an image is already encrypted using another algorithm and they by we feel at ease because of double encryption. This paper deals with demand response programs required for business models for securing their secret information. The Mean Square Error (MSE) test was used to compare the steganography image to the original image's degree of imperceptibility, as well as the Peak Signal to Noise Ratio (PSNR) test and histogram analysis from both test images. LSB - Steganography is a steganography technique in which the image's least significant bit is replaced with bits of the hidden message. These studies yielded good PSNR with an average of 56dB. Simultaneously the performance of the cryptographic technique is tested by entropy tests shows that to be very good because it approached with perfect values[8]. The attacker will have a difficult time decrypting the message using this technique because it is not frequently utilized. Because individuals nowadays rely on the internet to get their work done successfully and quickly, there are a rising number of attacks involving credit cards and money being taken, among other things.

So the combination of encryption produce high secure results and provide us secure connection between authorized users hence data transmission can be done without any worries. These types of technologies will be more helpful as a result of the loss of our personal information, ensuring that no one is readily assaulted. Steganography is not to be confused with cryptography, which requires changing the meaning of a communication such that it is undecipherable to those who are following it. A steganographic system's notion of failure differs from that of a cryptographic system in this context. The cryptographic system malfunctions and the algorithm fails when an attacker gains access to the encryption key and reads the secret message. When an attacker simply notices that the stenographic system has been utilized and that she can read the embedded message, the steganography procedure fails. Steganography should not be confused with cryptography, which is altering the meaning of a message in such a way that it is undecipherable to those who receive it. Payload: the information that will be hidden and sent surreptitiously, or the data that will be conveyed covertly;Any item in which the payload is surreptitiously implanted is referred to as a carrier.
The methods and tools utilized to build a hidden channel for conveying information are referred to as a stego-system. The carrier is sent over a data communication channel called a channel. The payload is extracted from the carrier using a key (not always applied). Throughout the twentieth century, steganography and steganalysis the process of identifying whether or not disguised information is being sent inside a carrier were actively developed. Steganalysis is basically the practice of attacking stego-systems [9].

In our digital era, where all information exchanges are boundless, cyber-threats have become more prevalent. One of them is the security of the data itself. A lot of information has been released or corrupted. For a business, a university, or a person, data security and confidentiality are critical. Because the data or information does not always reach the intended receivers, or when it does, the recipient who

should get the original data does not receive it because the data was mistakenly hacked. It's also conceivable that the data is corrupted in the first place. When it comes to receiving and exchanging data, computer networks are extremely convenient. In some situations, sending confidential data across public networks like the internet might make it insecure and vulnerable to attacks and unauthorised access by persons looking to steal it. In most incidents of data theft, data security is crucial. As a result, a tool is required to protect sensitive data from unwanted access. Data security and confidentiality can be protected using a combination of steganography and encryption [10].

The LSB method of encoding The 8th bit in a photograph is the least significant bit, and it has been changed as part of the hidden message. By altering the red, green, and blue values sequentially in a 24-bit image, three bits can be stored in each pixel. It is feasible to combine many colour components into a single colour component because each is represented by a byte The message has been successfully disguised because the human eye is unable to detect these modifications. You can make a fantastic first impression with the appropriate music. The smallest detail may be exploited to conceal the message. Steganography is used to conceal the difference between the second and least important significant. PNG, JPEG, TIFF, BMP, and other image file formats were used to test the suggested technique. Between the sender and the receiver, a common shared key is exchanged. The key might be updated at regular intervals to prevent man-in-the-middle attacks. The sender's data is encrypted with the shared secret key, resulting in a cipher text that must be embedded in an image to create a stego picture. Until now, encryption has always played the most important function in safeguarding the sender's and receiver's privacy. However, in addition to cryptography, steganography techniques are increasingly being employed to offer additional levels of protection to the hidden data. The benefit of utilising steganography instead of only cryptography is that the intended hidden message does not draw attention to itself as a target of investigation. In nations where encryption is unlawful, plainly visible encrypted messages, no matter how impenetrable they are, raise curiosity and may be damning in and of themselves. A digital picture may be described as a finite set of digital values known as pixels. Pixels are the tiniest individual elements in a picture, storing values that describe the brightness of a single colour at any given position. As a result, we may think of a picture as a pixel matrix (or two-dimensional array) with a set number of rows and columns.
The Least Significant Bit (LSB) approach modifies and replaces the last bit of each pixel with the secret message's data bit [11].

For message insertion, the K Means Clustering method is utilised. This method divides the message to be put into the image using clustering. Furthermore, an embedding approach is employed to determine where the message will be placed. The K Means Clustering method and message insertion process can both be seen in general. In 2015, Sown and Manikan proposed combining the LSB substitution and clustering approaches to overcome the LSB steganography method's shortcomings. A version of the

LSB method is the LSB substitution strategy. The length of the secret message bits is divided into a number of blocks by LSB, which is beneficial for multiplying and growing it. In K MEANS, the centroid is selected based on histogram data on pixel appearance, rather than at random. To make the position of the message harder to detect and reconstruct by outsiders, the centeroid value undergoes an updating procedure to generate a smooth region. AES 256 cryptography is employed as an added layer of protection. Because it executes numerous rotations, each of which consists of several stages, AES 256 has a bigger key size of 256 bits and a more complicated encryption procedure. Unsupervised learning method K-Means clustering Unlike supervised learning, there is no labelled data for this grouping. K-Means divides things into clusters based on their similarities and differences with objects in other groups.

The letter 'K' stands for a number. You must inform the system of the number of clusters you require. K = 2 denotes two clusters, for example. There is a method for determining the best or optimum value of K for a given set of data.

Let's use cricket as an example to better grasp what k-means are. Consider receiving data on a large number of cricket players from throughout the world, including statistics on runs scored. [12].

Many academics in this field have questioned the usage of cryptographic methods to incorporate encrypted messages in digital envelopes. The symmetric key and the public-private key are two methods for encrypting a communication before it is utilized for steganographic purposes. For symmetric algorithms, a symmetric key is defined, whereas for asymmetric algorithms, a public-private key is defined. The kind of application determines the application of each form of cryptography. ROC curves and confusion matrices are also generated using stag analysis approaches. Stag analysis algorithms are built and utilized to rely on two types of steganography methods: blind and steganography. Furthermore, each of these can be implemented. Data concealing is the study of concealed communications. In other terms, data concealment is the ability to conceal data in a media carrier, which allows a security institution to hide data in the form of a sender and receiver. Synthetic keys should be used in banking for encryption and decryption, for example. It is also feasible to generate these keys using RNG. FIPS 140-2, for example, is a set of industrial standards. Asymmetric cryptography is divided into two categories. Algorithms. The first are
algorithms that use block cyphers to encrypt data. A collection of bit lengths in electronic data blocks that are encoded with a unique algorithm a key that is hidden The data is kept safe and secure by encrypting it [13].

Data security refers to the protection of data against various types of attacks while it is being transmitted via a secure or insecure channel, in addition to when it is saved on a device. Privacy, integrity,

confidentiality, authenticity, resilience, payload capacity, and imperceptibility are all guaranteed by data security. Balancing all of the aforementioned criteria for execution has always been a difficult undertaking. As a result, the factors that ensure data security have been prioritized and met in accordance with the requirements of the applications. Cryptography and steganography are two branches of study that deal with the techniques and procedures used to assure data security. Secure system characters to ensure an efficient and safe system, several characteristics must be met.

- Visual quality
- Robustness
- Payload capacity
- Invisibility
- Privacy
- Confidentiality

In terms of contrast, brightness, and other factors, visual quality measures how comparable an original image is to its processed form. The capacity of a system to withstand and recover from hard attacks is referred to as robustness. The largest quantity of data that may be hidden in a cover media without changing any of the medium's characteristics is known as payload capacity. The ability of data to remain entirely hidden without drawing the intruder's attention is referred to as imperceptibility. The capacity of a user to choose which of their publicly available information should be made visible or invisible to the general public or a limited group of people is known as privacy. Confidentiality refers to a system's ability to keep sensitive data secure [14].

Steganography is a method of hiding secret information in such a way that it is not instantly detectable. In this situation, the medium utilized to conceal secret information is digital photos, because steganography research that employs images as cover media is quite popular, It's also fairly usual to receive and transfer picture data. Because it causes only minor noise, the procedure of inserting secret data utilizing blue values on the RGB image cover. The image has been distorted in order to create a stego image. Greater quality than utilizing RGB red and green values cover image MSE, PSNR, and histogram tests are all available. Analysis is used to determine the quality of a stego picture. The largest quantity of data that may be hidden in a cover media without changing any of the medium's characteristics.

The use of LSB and Blowfish-RC4 in combination has proven to be effective. The quality of the stego image was excellent. An imperceptibility test revealed an average PSNR value of above 56dB, with the lowest score of only 49.73dB. The stego picture's histogram indicated a little difference from the original image. Encrypted files employing the Blowfish-RC4 algorithm have been shown to yield very excellent encryption, with entropy values ranging from 7.99. Because it is so close to 8, the entropy value of the Blowfish-RC4 algorithm combo is virtually ideal. The extraction and decryption processes

may also be faultless. The stego image is required for extracting the ciphertext, as well as the Blowfish and RC4 keys required for decrypting the extracted ciphertext. The cover's ciphertext was successfully recovered. Because it is practically identical to the cover image, the stego image is outstanding. A good histogram must have minor changes between the two photos. To obtain the original file, the image is decrypted using the RC4 algorithm with the RC4 key and then decrypted again using the Blowfish algorithm with the Blowfish key. As a result of the decryption method, the original file .

RC4 was previously utilized in a variety of applications, including SSL/TLS and WEP, until serious flaws in the protocol were discovered in 2003 and 2013. Because WEP employed RC4, attackers could practice cracking it as often as they wanted. A defect in RC4 was discovered as a result of this behaviour, and the encryption key used by RC4 could be broken in less than a minute. RC4 keys can be 64 or 128 bits in length, with the 128-bit key being produced in seconds. Because WEP was the only Wi-Fi security protocol available at the time, the following phase, Wi-Fi Protected Access (WPA), had to be pushed into production.

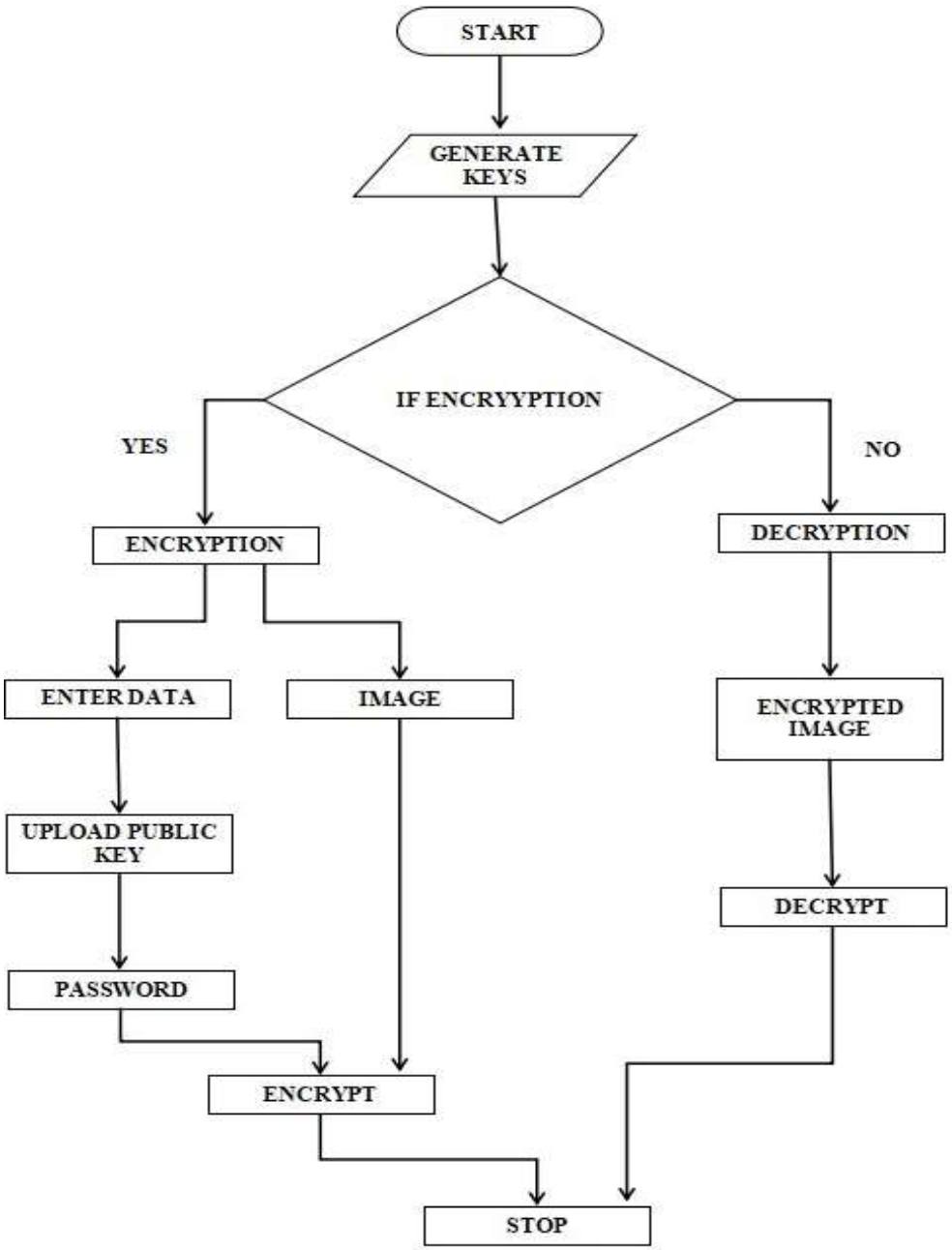
In 2013, another flaw in RC4 was identified while it was being used as a workaround for a cypher block. Bruce Schneier is the creator of Blowfish. It's a symmetric key cryptography algorithm with a lot of strength.Blowfish has the following characteristics:
On 32-bit microprocessors, the Blowfish encryption state is quick.
Blowfish is small and light, requiring less than 5KB of RAM to run.
Blowfish is simple to build and manipulate since it just requires elementary operations like addition, XOR, and table lookup.
Blowfish has a configurable key length of up to 448 characters, making it both versatile and secure.
The four variants are:
Spritz is a tool for creating cryptographic hashes and deterministic random bit generators.
RC4A — This is a version of the RC4 encryption that was designed to be quicker and more secure than the standard RC4 cypher.
It was discovered that the RC4A encryption does not use really random numbers.
VMPC - Variably Modified Permutation Composition (VMPC) is a variation of RC4 that, unlike RC4A, has not been discovered to employ really random values in its cypher.
RC4A+ — A more sophisticated form of RC4A, RC4A+ is lengthier and more complex than RC4 and RC4A, but it is also more powerful as a result of its complexity[15].

CHAPTER-3 THEORETICAL ANALYSIS

3.1PROPOSED SYSTEM

- Generate Keys(Private, Public)
- Give the Input the IP Address which we want to hide or encrypt.
- Upload the image in which we want to hide.
- It will read the image Pixel values.
- For each element of pixel, replace rightmost 2 bits in R or G or B
- For remaining values, replace the rightmost bit.
- Continue step 5,6
- Output: Stegano Image.
- Enter the pass.
- Upload the public key which we have generated.
- Encrypt data using RSA.
- Use the AES encryption technique to encrypt the encrypted text once more.Decryption: Enter the encrypted image.
- Click Decrypt the data which we have hidden is displayed

3.2Flow Chart



3.3Implementation:

We have 3 files named
int.py-which is the main program,
e1- for encryption,
d1- for decryption.

3.3.1 int.py

```
import tkinter as tk
from tkinter import ttk,filedialog
import sys
import ipaddress
import re
import socket
from cryptosteganography import CryptoSteganography
import os
from Crypto.Cipher import AES
from Crypto.Hash import SHA256
from Crypto import Random
import random

maxplength = 1000000000000
impath=""
fpath=""
fname=""

def dgcd(c,d):
    if c == 0:
        return (d, 0, 1)
    else:
        g, m, n = dgcd(b % a, a)
        return (g, n- (b // a) * m, m)



def gcd(c,d):
    while d!= 0:
        c,d = d, c % d
```

```
    return c


def isprimecheck(number):
    if number == 2:
        return True
    if number < 2 or number% 2 == 0:
        return False
    for n in range(3, int(number**0.5)+2, 2):
        if number % n == 0:
            return False
    return True

def grandomprim():
    while(1):
        rprime = random.randint(0,maxplength)
        if isprimecheck(rprime):
            return rprime

def gkpair():
    p = grprim()
    q = ()

    n = p*q
    '''eul(n) = eul(p)*eul(q)'''
    eul = (p-1) * (q-1)

    '''choose e coprime to n and 1 > e > eul'''
    e = random.randint(1, eul)
    g = gcd(e,eul)
    while g != 1:
        e = random.randint(1, eul)



    print("e=",e," ","eul=",eul)
    '''l[1] = modular inverse of e and eul'''
    l= dgcd(e, eul)[1]
```

```
        '''make sure l is positive'''
        l= l % eul
        if(l < 0):
                l += eul

        return ([e,n],[l,n])

def decry(ciptext,privatekey):
        try:
                key,n = privatekey
                text1= [chr(pow(char,key,n)) for char in ciptext]
                return ''.join(text1)
        except TypeError as e1:
                print(e1)



def decrypted1(key, fname):
        chunk1sizelength = 64 * 1024
        ofile = fname

        with open(fname, 'rb') as inputfile:
                filesize = int(inputfile.read(16))
                IV = inputfile.read(16)

                decryptoralg= AES.new(key, AES.MODE_CBC, IV)

                with open(oFile, 'wb') as outputfile:
                        while True:
                                chunk11= inputfile.read(chunk1sizelength)

                                if len(chunk11) == 0:
                                        break

                                outputfile.write(decryptor1.decry(chunk11))
                        outputfile.truncate(fsize)
```

```
def gettingkey(pass):
    hasher = SHA256.new(pass.encode('utf-8'))
    return hasher.digest()


def gkeys():
    publickey,privatekey  = gkpair()

    print("Private:  ",privatekey)
    with open('public.txt',  'w') as fhandle:
        fhandle.writelines("%s\n"  % place for place in publickey)
    with open('private.txt',  'w') as fhandle:
        fhandle.writelines("%s\n"  % place for place in privatekey)
    generate.config(state="disabled")

def encrypt1(key,  fname):
    chunk1size  = 64 * 1024

    fsize = str(os.path.getsize(fname)).zfill(16)
    IV = Random.new().read(16)

    Encryptor1 = AES.new(key, AES.MODE_CBC, IV)

    with open(fname,  'rb') as inputfile:
        with open(outputFile,  'wb') as outputfile:
            outputfile.write(filesize.encode('utf-8'))
            outputfile.write(IV)

            while True:
                chunk1 1= inputfile.read(chunk1sizelength)

                if len(chunk1)  == 0:
                    break
                elif len(chunk1)  % 16 != 0:
                    chunk1 += b' ' * (16 - (len(chunk1)  % 16))

                outputfile.write(encryptor1.encrypt(chunk1))
```

```
def gettingkey(pass):
    hasher1 = SHA256.new(pass.encode('utf-8'))
    return hasher1.digest()


def encry(text,publickey):
    key,n = publickey
    ciphtext = [pow(ord(char),key,n)  for char in text1]
    return ciphtext

def gettingimagefile():
    global fname
    fname = tk.filedialog.askopenfname(filetypes=[("Image  Files",(".png"))])
    print(fname)
    decryptb.config(state="normal")

def gettingipath():
    global ipath
    ipath = tk.filedialog.askopenfname(filetypes=[("Image  Files",(".jpg"))])


def getfname():
    global fpath
    fpath = tk.filedialog.askopenfname(filetypes=[("Text  Files",(".txt"))])

def decrypted1():
    global fname
    ciphtext=[]
    privatekey=[]
    with open('cipher.txt',  'r') as fhandle:
        ciphtext= [current_place.rstrip() for current_place in fhandle.readlines()]
    with open('private.txt',  'r') as fhandle:
        privatekey= [current_place.rstrip() for current_place in fhandle.readlines()]
    ciphtext = [int(i)  for i in ciphtext]
    privatekey = [int(i)  for i in privatekey]
    pass= decrypt(ciphtext,privatekey)
    print(ciphtext)
```

```
        decrypted1(gettingkey(pass), fname)
        crypstegano = CryptoSteganography('My secret pass key')
        secretimage = crypstegano.retrieve('encrypted.png')

        print("this is given ip adress",secretimage)
        print("Done.")


def encryption():
        global ipath,fpath
        domainname = domainentry.get()
        ip1=0
        try:
                ip1=socket.gethostbyname(domainname)
                error.config(text="Valid Address")
        except socket.gaierror:
                error.config(text="Enter Valid IP Address")
        x=ipaddress.ip_address(ip1)
        passwrd = passentry.get()
        #encryption starts
        # Save the encrypted file inside the image
        crypstegano = CryptoSteganography('My secretimage pass key')

        img=Image.open("encrypted.png")
        img.save("encrypted.png")
        #fname = input("File to encrypt: ")
        fname="encrypted.png"
        encrypt1(gettingkey(passwrd), fname)
        publickey=[]
        places=[]
        with open('public.txt', 'r') as fhandle:
                places = [current_place.rstrip() for current_place in fhandle.readlines()]
        publickey=[int(i) for i in places]

        comp.config(text="Cipher Text created")
        with open('cipher.txt', 'w') as fhandle:
                fhandle.writelines("%s\n" % place for place in ciphtext)
```

```
        encrypt.config(state="disabled")
        fnameentry.config(state="normal")



ui =tk.Tk()

tcontrol = ttk.Notebook(ui)
tab1=ttk.Frame(tcontrol)
tab2=ttk.Frame(tcontrol)
tcontrol.add(tab1,text  = "Encrypt")
tcontrol.add(tab2,text  = "Decrypt")
tcontrol.select(tab2)
tcontrol.grid(row=0,column=0)

dom=  tk.Label(tab1,text="Enter  Domain")
dom.grid(row=0,column=0)
doment  = tk.Entry(tab1)
doment.grid(row=0,column=1)
doment.focus()

error1 = tk.Label(tab1)
error1.grid(row=1,column=1)

image1  = tk.Label(tab1,text="Upload  Image")
image1.grid(row=2,column=0)

button2  = tk.Button(tab1,text="Browse  Image")
button2.config(command=getipath)

pass = tk.Label(tab1,text="Enter  Pass")
pass.grid(row=3,column=0)
passentry = tk.Entry(tab1)
passentry.grid(row=3,column=1)

publickey  = tk.Label(tab1,text="Upload  Public  Key")
publickey.grid(row=4,column=0)
```

```
button2  = tk.Button(tab1,text="Browse  File")
button2.grid(row=4,column=1)
button2.config(command=getfname)

encrypt1 = tk.Button(tab1,text="Encrypt")
encrypt1.grid(row=6,column=1)
encrypt1.config(command=encryption)

compute = tk.Label(tab1)
compute.grid(row=5,column=1)

generate1 = tk.Button(tab2,text="Generate  Keys")
generate1.grid(row=0,column=0)
generate1.config(command=gkeys)

fname=tk.Label(tab2,text="Enter  Fname")
fnameentry = tk.Button(tab2,text="Browse  Image")
fnameentry.grid(row=1,column=1)
fnameentry.config(command=getimagefile,state="disabled")

output 1= tk.Label(tab2,text="IP  Address")
output1grid(row=3,column=0)
outputentry1  = tk.Entry(tab2)
outputentry1.grid(row=3,column=1)

decryptb1  = tk.Button(tab2,text="Decrypt")
decryptb1.grid(row=2,column=1)
decryptb1.config(state="disabled",command=decrypted1)

ui.mainloop()
```

3.3.2 d1.py

```
import os
from Crypto.Cipher import AES
from Crypto.Hash import SHA256
from cryptosteganography import CryptoSteganography
from PIL import Image

import random
maxplength = 1000000000000


def dgcd(a, b):
    if a == 0:
        return (b, 0, 1)
    else:
        g, y, x = dgcd(b % a, a)
        return (g, x - (b // a) * y, y)


def gcd(a, b):
    while b != 0:
        a, b = b, a % b
    return a


def isprimecheck(num):
    if num == 2:
        return True
    if num < 2 or num % 2 == 0:
        return False
    for n in range(3, int(num**0.5)+2, 2):
        if num % n == 0:
            return False
    return True
```

```
def grprim():
    while(1):
        rprime  = random.randint(0,maxplength)
        if isprimecheck(rprime):
            return rprime

def gkpair():
    p = grprim()
    q = grprim()

    n = p*q
    '''eul(n) = eul(p)*eul(q)'''
    eul = (p-1) * (q-1)

    '''choose e coprime  to n and 1 > e > eul'''
    e = random.randint(1,  eul)

    while  g != 1:
        e = random.randint(1,  eul)
        g = gcd(e, eul)

    print("e=",e,"  ","eul=",eul)
    '''d[1] = modular  inverse  of e and eul'''
    d = dgcd(e,  eul)[1]

    '''make sure d is positive'''
    d = d % eul
    if(d < 0):
        d += eul

    return ([e,n],[d,n])
def decrypt(ciphtext,privatekey):
    try:
        key,n  = privatekey
        print(key,n)
        #print(pow(char,key,n)  for char in ciphtext)
        text = [chr(pow(char,key,n))  for char in  ciphtext]
```

```
            print(text)
            return "".join(text)
        except TypeError as e:
            print(e)


def decrypted(key, fname):
    chunk1size = 64 * 1024
    outputFile = fname[11:]

    with open(fname, 'rb') as inputfile:
        filesize = int(inputfile.read(16))



        decryptor = AES.new(key, AES.MODE_CBC, IV)

        with open(outputFile, 'wb') as outputfile:
            while True:
                chunk1 = inputfile.read(chunk1size)

                if len(chunk1) == 0:
                    break

                outputfile.write(decryptor.decrypt(chunk1))
            outputfile.truncate(filesize)
def gettingkey(pass):
    hasher = SHA256.new(pass.encode('utf-8'))
    return hasher.digest()

def Main():
        a=int(input("Enter choice "))
        if(a==1):
            publickey,privatekey = gkpair()
            print("Public: ",publickey)
            print("Private: ",privatekey)
            with open('public.txt', 'w') as fhandle:
                fhandle.writelines("%s\n" % place for place in publickey)
```

```
            with open('private.txt', 'w') as fhandle:
                fhandle.writelines("%s\n" % place for place in privatekey)
        elif(a==2):
            ciphtext=[]
            privatekey=[]
            with open('cipher.txt', 'r') as fhandle:
                ciphtext= [current_place.rstrip() for current_place in fhandle.readlines()]

                privatekey= [current_place.rstrip() for current_place in fhandle.readlines()]
            ciphtext = [int(i) for i in ciphtext]
            privatekey = [int(i) for i in privatekey]
            print(privatekey)
            pass= decrypt(ciphtext,privatekey)
            print(pass)

            fname = input("File to decrypt: ")

            print(ciphtext)
            decrypted(gettingkey(pass), fname)
            crypstegano = CryptoSteganography('My secretimage pass key')
            secretimage = crypstegano.retrieve('encrypted.png')
            print("this is ip adress",secretimage)
            print("Done.")
if __name__ == '__main__':
    Main()
```

3.3.3 e1.py

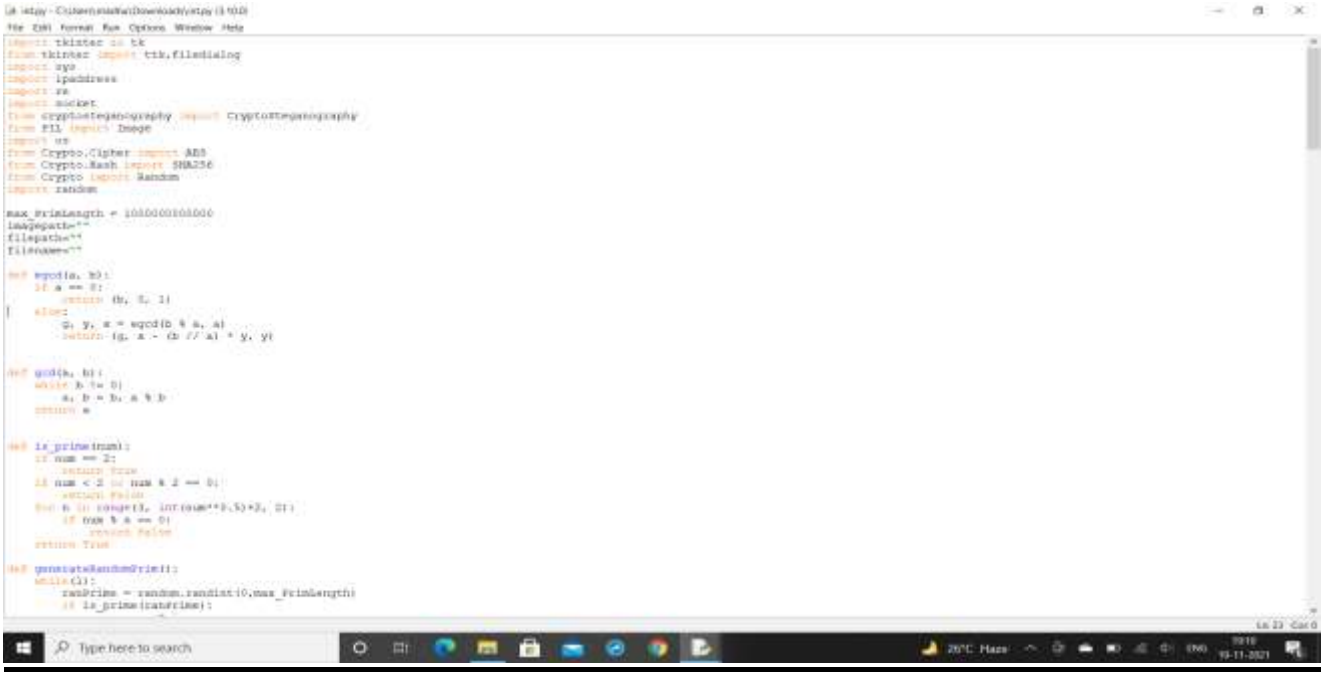
```
import sys
import ipaddress
import re
import socket
from cryptosteganography import CryptoSteganography
from PIL import Image
import os
from Crypto.Cipher import AES
from Crypto import Random
import random
maxplength = 1000000000000


def checking(ip):
    if(re.search(regexexp, ip)):
        print("Valid ip address")
        return 1
    else:
        print("Invalid ip address")
        return 0
#encryption

def encrypt(key, fname):
    chunk1size = 64 * 1024
    outputFile = "(encrypted)" + fname
    IV = Random.new().read(16)

    Encryptor1 = AES.new(key, AES.MODE_CBC, IV)

    with open(fname, 'rb') as inputfile:
        with open(outputFile, 'wb') as outputfile:
            outputfile.write(filesize.encode('utf-8'))
            outputfile.write(IV)
```

```
                while True:
                    chunk1 = inputfile.read(chunk1size)

                    if len(chunk1) == 0:
                        break
                    elif len(chunk1) % 16 != 0:
                        chunk1 += b' ' * (16 - (len(chunk1) % 16))

                    outputfile.write(encryptor.encrypt(chunk1))
def gettingkey(pass):
    hasher = SHA256.new(pass.encode('utf-8'))
    return hasher.digest()


def encrypted1(text,publickey):
    key,n = publickey
    ciphtext = [pow(ord(char),key,n) for char in text]
    print(ciphtext)
    return ciphtext
if __name__ == '__main__':

    hostname=input()

    ipadd1=socket.gethostname(hostname)
    print('host name is:'+ipadd1)



    x1=ipaddress.ip_address(ipadd1)
    if(check(ipadd1)==1 and x1.version==4):
        print("it is ipv4 address")

    else:
        print("please enter a valid ip address")
    crypstegano = CryptoSteganography('My secretimage pass key')

    inputimage=input("Enter image path: ")
    crypstegano.hide(inputimage, 'encrypted.png', ip1)
```

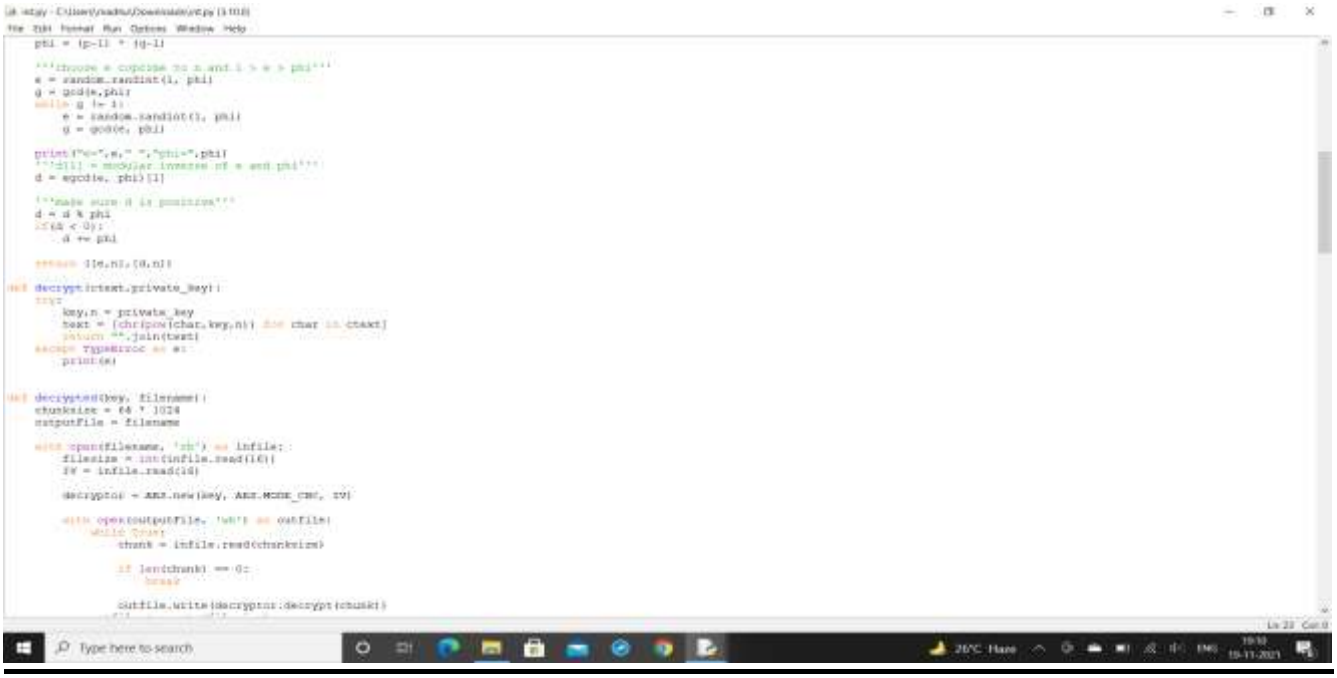
```
img1=Image.open("encrypted.png")
img1.save("encrypted.png")
fname="encrypted.png"
pass = input("Pass: ")
encrypt1gettingkey(pass), fname)

places=[]
with open('public.txt', 'r') as fhandle:
        places = [current_place.rstrip() for current_place in fhandle.readlines()]
publickey=[int(i) for i in places]
ciphtext = encrypted(pass,publickey)

with open('cipher.txt', 'w') as fhandle:
        fhandle.writelines("%s\n" % place for place in ciphtext)
```

CHAPTER-4 EXPERIMENTAL INVESTIGATION

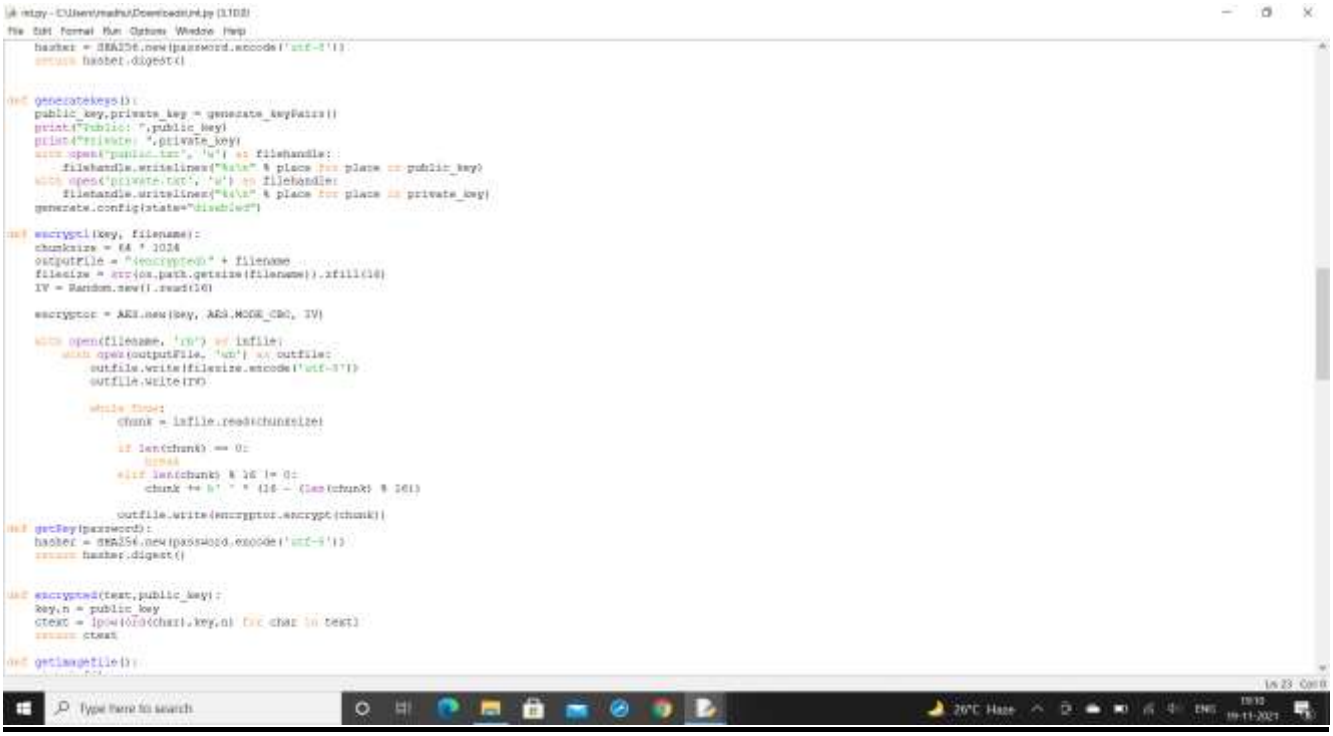
We have imported necessary packages and we have written RSA algorithm code.



With the private key, decrypt the encrypted text.

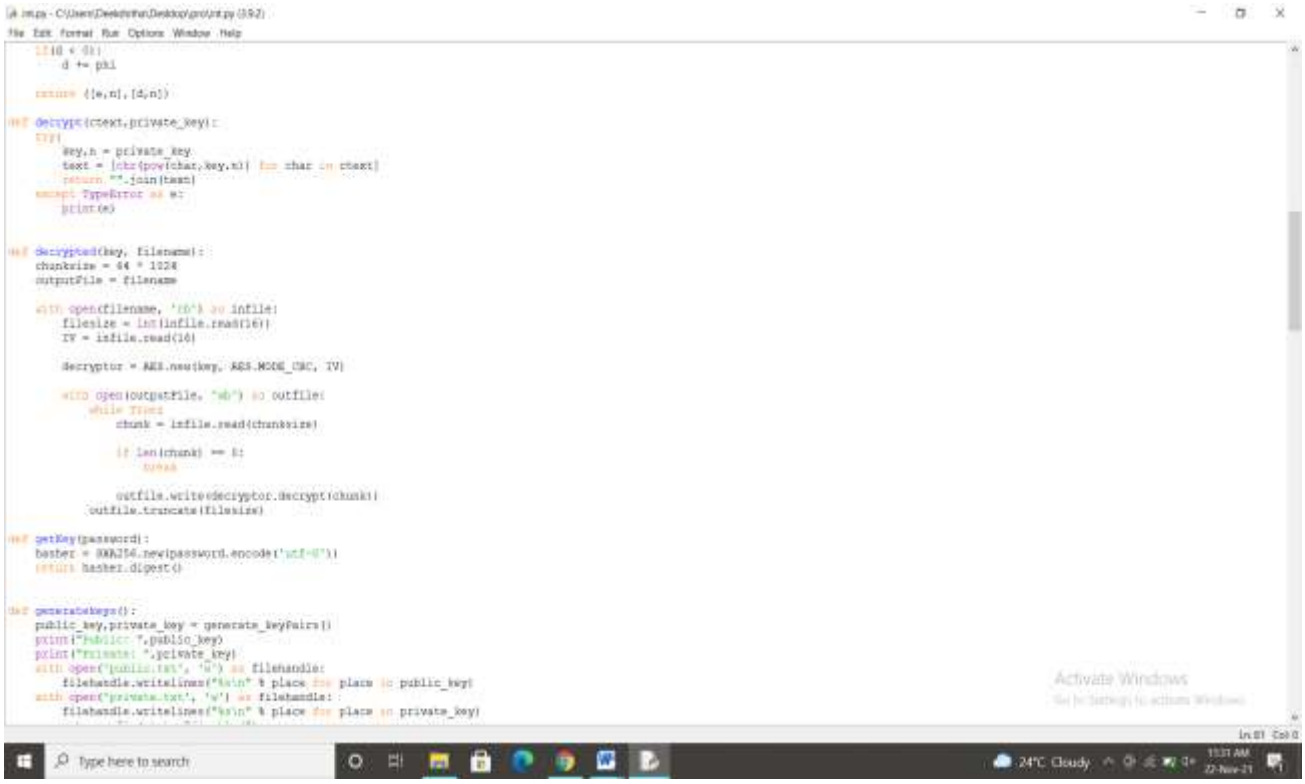


The AES algorithm is used to create keys and to encrypt data.



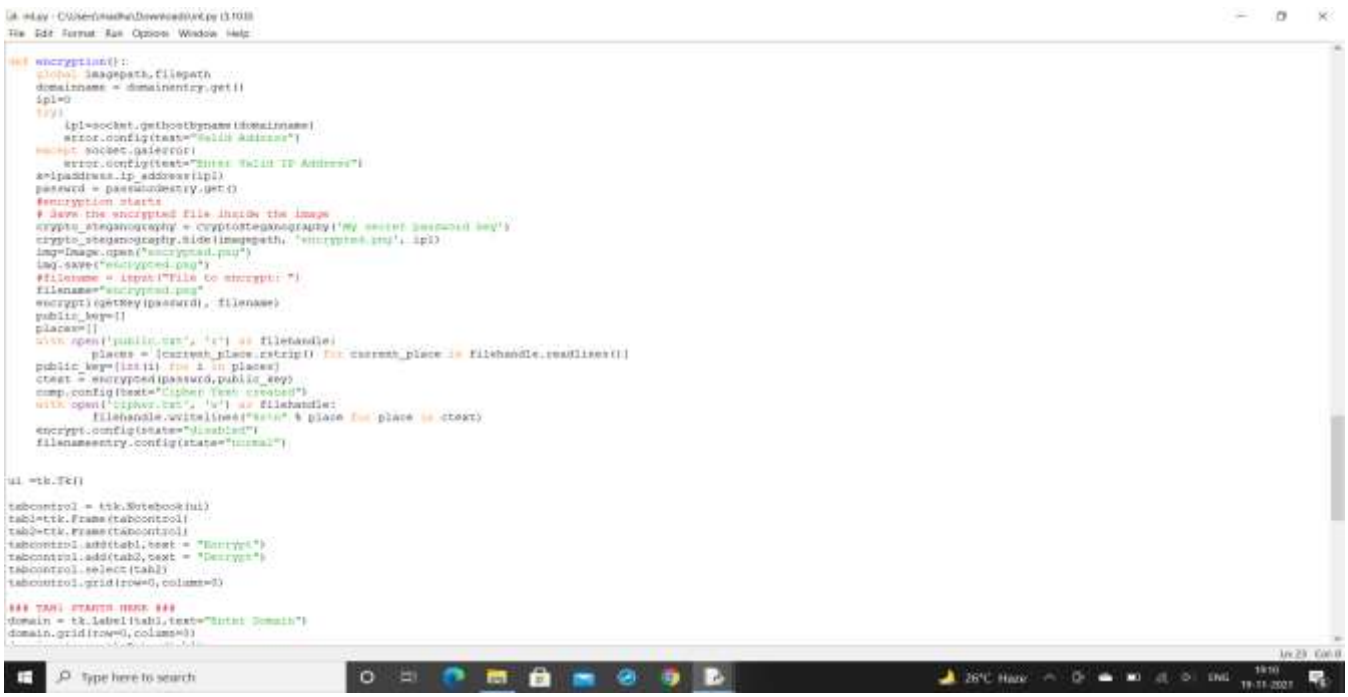
```
    hasher = SHA256.new(password.encode('utf-8'))
    return hasher.digest()

def generatekeys():
    public_key,private_key = generate_keyPairs()
    print("Public: ",public_key)
    print("Private: ",private_key)
    with open('public.txt', 'w') as filehandle:
        filehandle.writelines("%s\n" % place for place in public_key)
    with open('private.txt', 'w') as filehandle:
        filehandle.writelines("%s\n" % place for place in private_key)
    generate.config(state="disabled")

def encrypt(key, filename):
    chunksize = 64 * 1024
    outputFile = "(encrypted)" + filename
    filesize = str(os.path.getsize(filename)).zfill(16)
    IV = Random.new().read(16)

    encryptor = AES.new(key, AES.MODE_CBC, IV)

    with open(filename, 'rb') as infile:
        with open(outputFile, 'wb') as outfile:
            outfile.write(filesize.encode('utf-8'))
            outfile.write(IV)

            while True:
                chunk = infile.read(chunksize)

                if len(chunk) == 0:
                    break
                elif len(chunk) % 16 != 0:
                    chunk += b' ' * (16 - (len(chunk) % 16))

                outfile.write(encryptor.encrypt(chunk))
def getKey(password):
    hasher = SHA256.new(password.encode('utf-8'))
    return hasher.digest()

def encrypted(text,public_key):
    key,n = public_key
    ctext = [pow(ord(char),key,n) for char in text]
    return ctext

def getimagefile():
```

Code for entering the pass and all the functions necessary for uploading image and uploading the public key file.



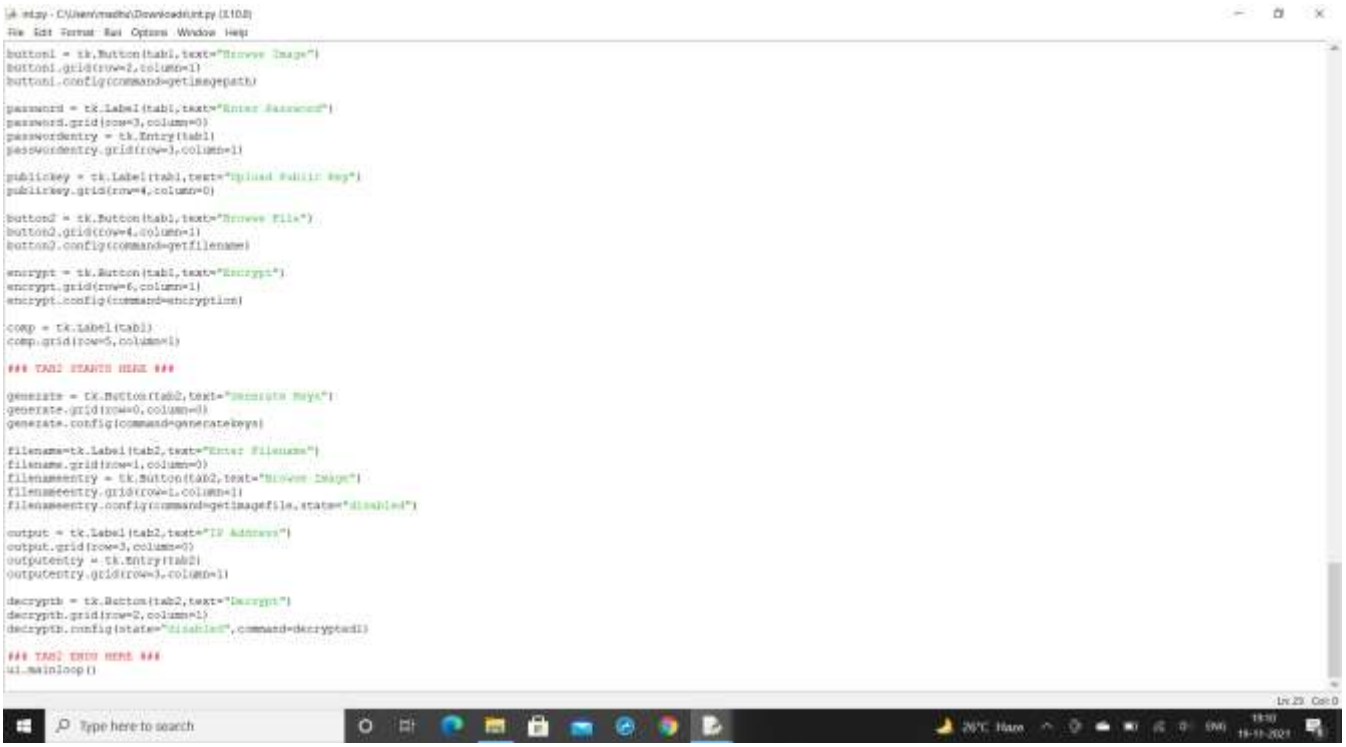
```
    if d < 0:
        d += phi

    return ([e,n],[d,n])

def decrypt(ctext,private_key):
    try:
        key,n = private_key
        text = [chr(pow(char,key,n)) for char in ctext]
        return "".join(text)
    except TypeError as e:
        print(e)


def decrypted(key, filename):
    chunksize = 64 * 1024
    outputFile = filename

    with open(filename, 'rb') as infile:
        filesize = int(infile.read(16))
        IV = infile.read(16)

        decryptor = AES.new(key, AES.MODE_CBC, IV)

        with open(outputFile, "wb") as outfile:
            while True:
                chunk = infile.read(chunksize)

                if len(chunk) == 0:
                    break

                outfile.write(decryptor.decrypt(chunk))
            outfile.truncate(filesize)

def getKey(password):
    hasher = SHA256.new(password.encode('utf-8'))
    return hasher.digest()


def generatekeys():
    public_key,private_key = generate_keyPairs()
    print("Public: ",public_key)
    print("Private: ",private_key)
    with open('public.txt', 'w') as filehandle:
        filehandle.writelines("%s\n" % place for place in public_key)
    with open('private.txt', 'w') as filehandle:
        filehandle.writelines("%s\n" % place for place in private_key)
```

Main function for encrypting the IP Address using both the algorithms.



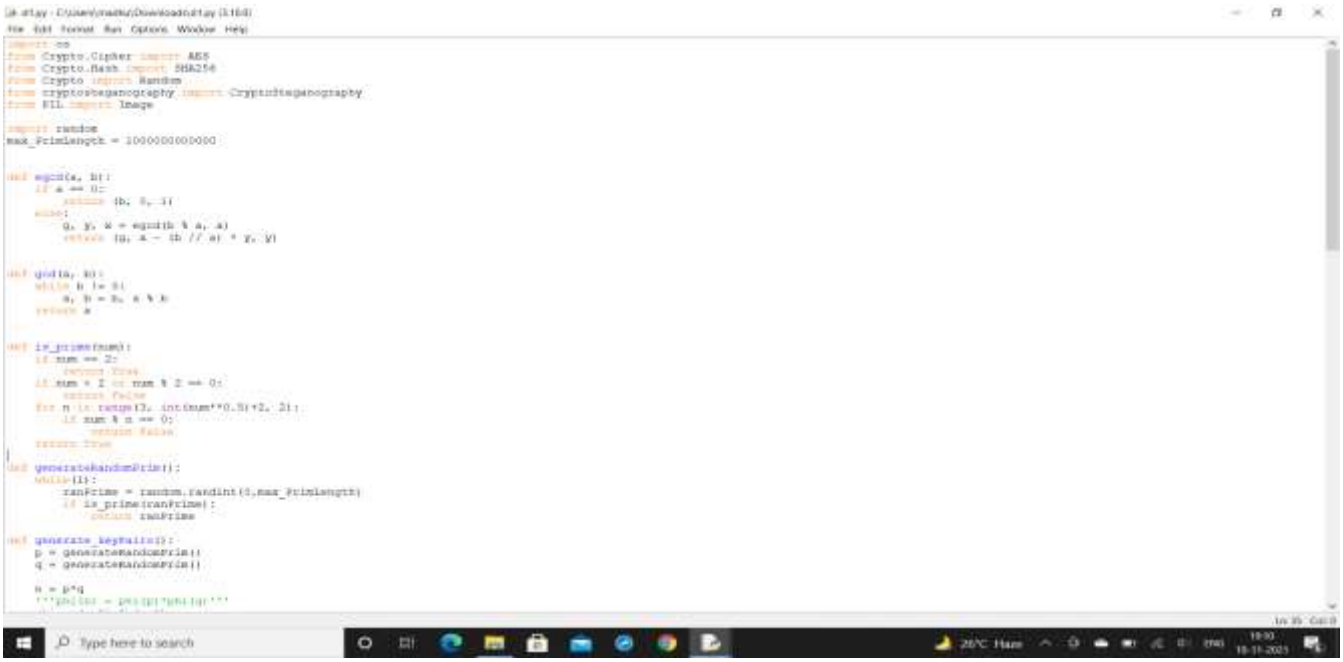
```
def encryption():
    global imagepath,filepath
    domainname = domainentry.get()
    ipl=0
    try:
        ipl=socket.gethostbyname(domainname)
        error.config(text="Valid Address")
    except socket.gaierror:
        error.config(text="Enter Valid IP Address")
    a=ipaddress.ip_address(ipl)
    password = passwordentry.get()
    #encryption starts
    # Save the encrypted file inside the image
    crypto_steganography = cryptoSteganography('MY secret password key')
    crypto_steganography.hide(imagepath, 'encrypted.png', ipl)
    img=Image.open("encrypted.png")
    img.save("encrypted.png")
    #filename = input("File to encrypt: ")
    filename="encrypted.png"
    encrypt(getKey(password), filename)
    public_key=[]
    places=[]
    with open('public.txt', 'r') as filehandle:
            places = [current_place.rstrip() for current_place in filehandle.readlines()]
    public_key=[int(i) for i in places]
    ctext = encrypted(password,public_key)
    comp.config(text="Cipher Text created")
    with open('cipher.txt', 'w') as filehandle:
            filehandle.writelines("%s\n" % place for place in ctext)
    encrypt.config(state="disabled")
    filenameentry.config(state="normal")


ui =tk.Tk()

tabcontrol = ttk.Notebook(ui)
tab1=ttk.Frame(tabcontrol)
tab2=ttk.Frame(tabcontrol)
tabcontrol.add(tab1,text = "Encrypt")
tabcontrol.add(tab2,text = "Decrypt")
tabcontrol.select(tab2)
tabcontrol.grid(row=0,column=0)

### TAB1 STARTS HERE ###
domain = tk.Label(tab1,text="Enter Domain")
domain.grid(row=0,column=0)
```

Code for executing and taking the values from Textfields and buttons.

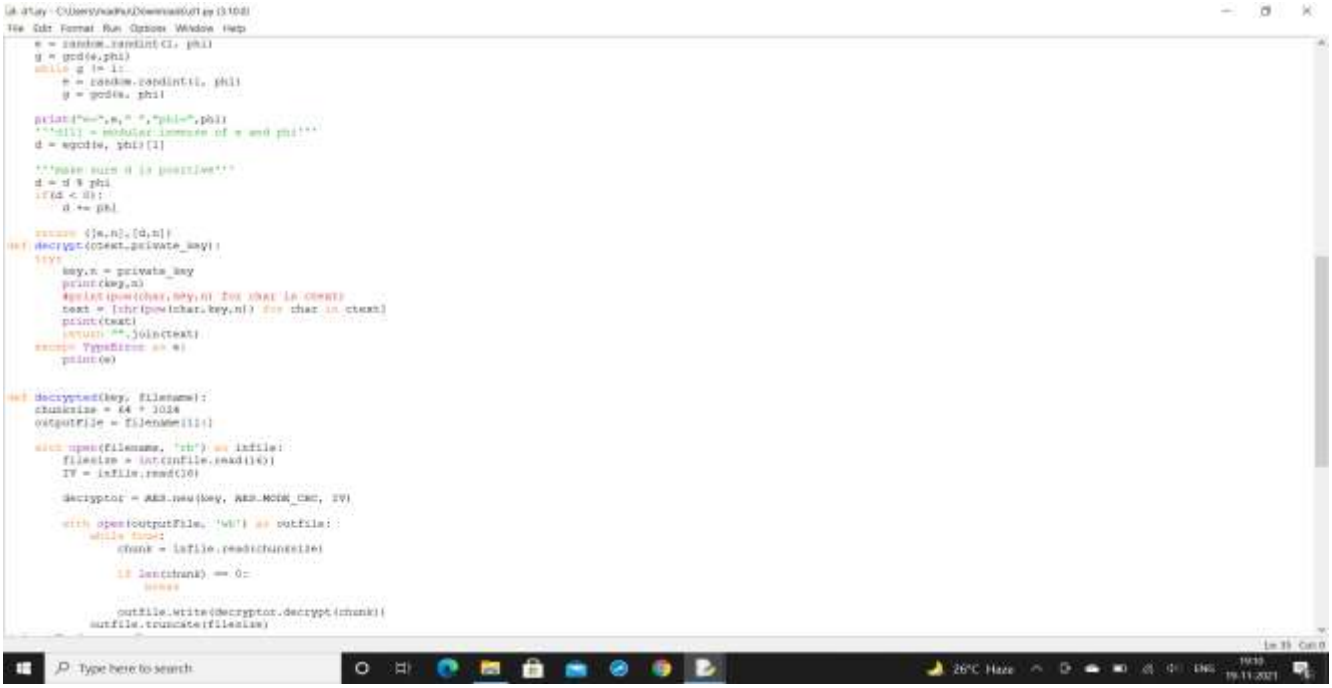


```
button1 = tk.Button(tab1,text="Browse Image")
button1.grid(row=2,column=1)
button1.config(command=getimagepath)

password = tk.Label(tab1,text="Enter Password")
password.grid(row=3,column=0)
passwordentry = tk.Entry(tab1)
passwordentry.grid(row=3,column=1)

publickey = tk.Label(tab1,text="Upload Public Key")
publickey.grid(row=4,column=0)

button2 = tk.Button(tab1,text="Browse File")
button2.grid(row=4,column=1)
button2.config(command=getfilename)

encrypt = tk.Button(tab1,text="Encrypt")
encrypt.grid(row=6,column=1)
encrypt.config(command=encryption)

comp = tk.Label(tab1)
comp.grid(row=5,column=1)

### TAB2 STARTS HERE ###

generate = tk.Button(tab2,text="Generate Keys")
generate.grid(row=0,column=0)
generate.config(command=generatekeys)

filename=tk.Label(tab2,text="Enter Filename")
filename.grid(row=1,column=0)
filenameentry = tk.Button(tab2,text="Browse Image")
filenameentry.grid(row=1,column=1)
filenameentry.config(command=getimagefile,state="disabled")

output = tk.Label(tab2,text="IP Address")
output.grid(row=3,column=0)
outputentry = tk.Entry(tab2)
outputentry.grid(row=3,column=1)

decrypth = tk.Button(tab2,text="Decrypt")
decrypth.grid(row=2,column=1)
decrypth.config(state="disabled",command=decryptad1)

### TAB2 ENDS HERE ###
ui.mainloop()
```

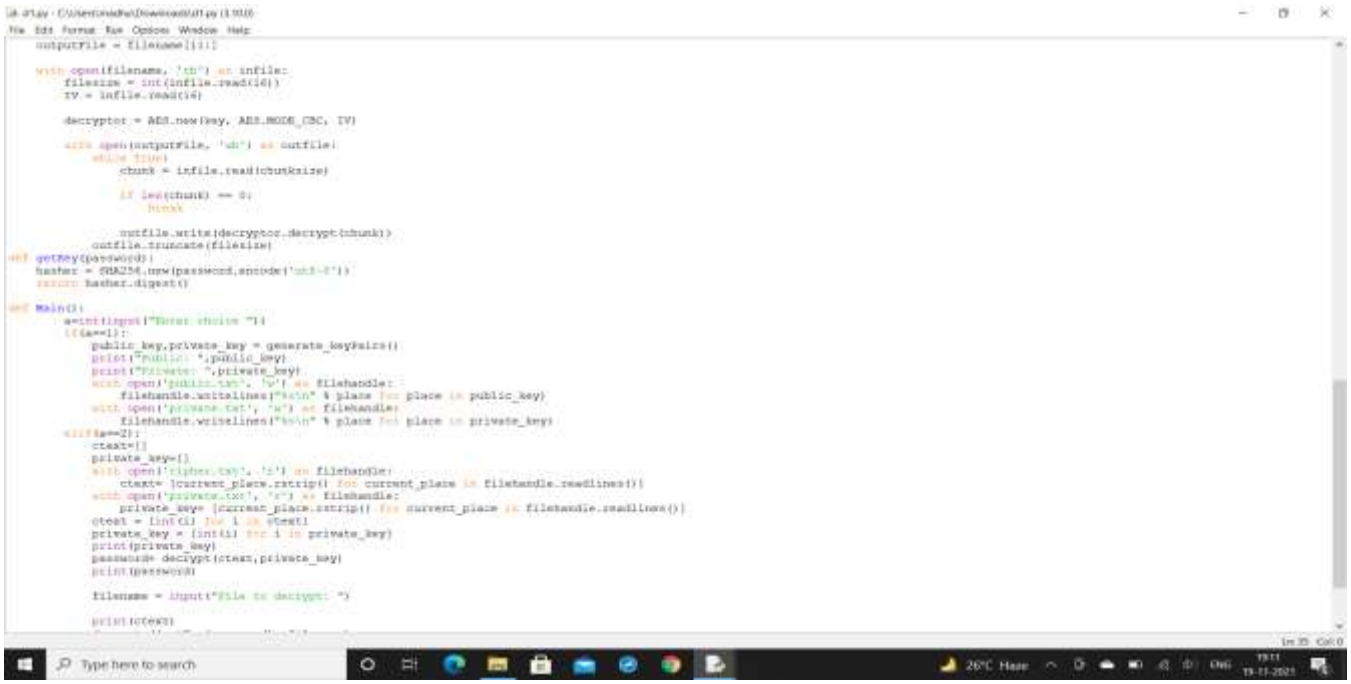
This is the code for decrypting data and code for RSA Algorithm.



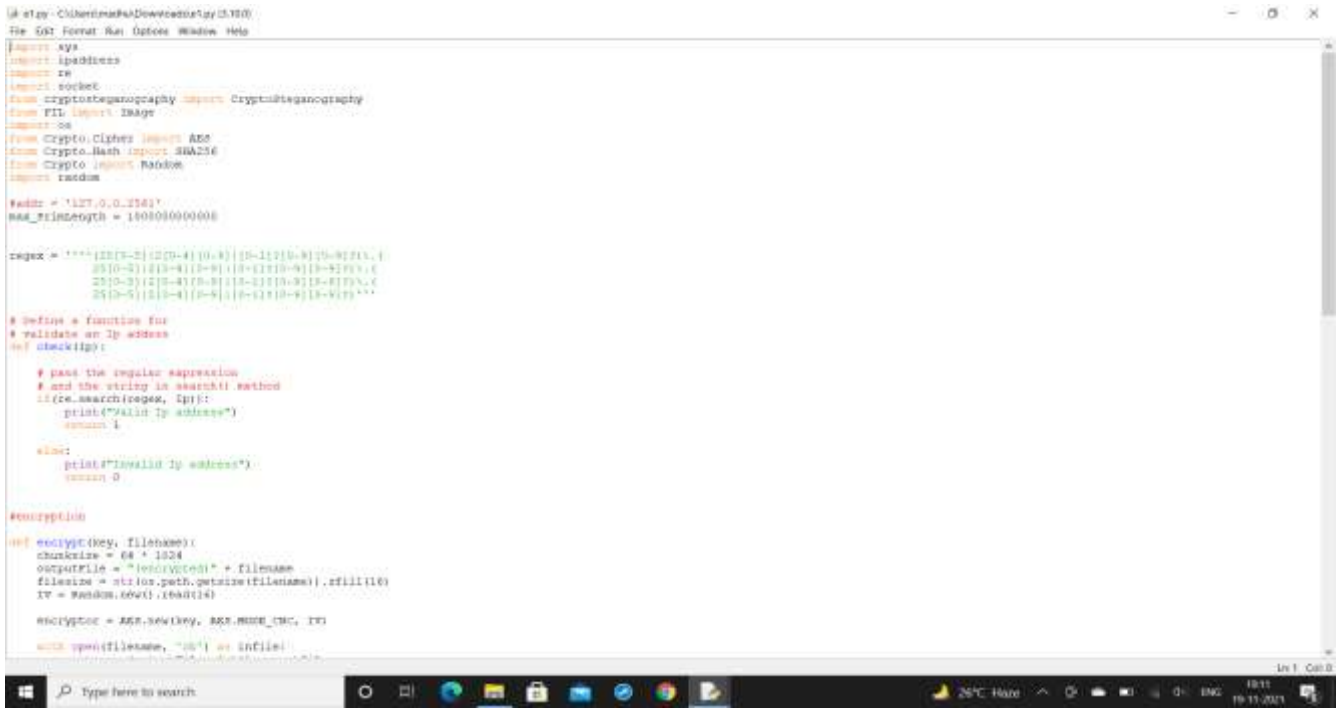
Code for decrypting data using AES Algorithm.



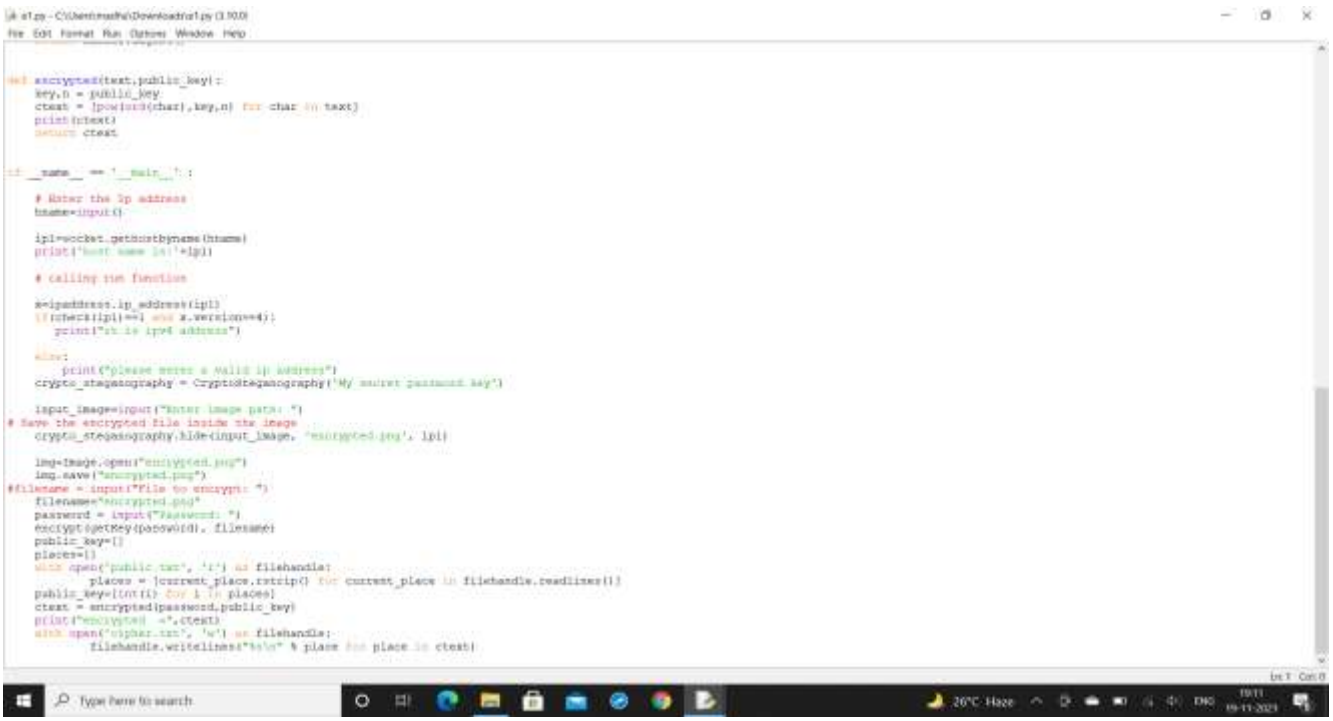
Decrypting the encrypted text with the private key and the picture is the main approach.



Python file for encrypting the data and checking the data is valid or not.



The main function for verifying the user's input and obtaining data.



4.1Usecase diagrams

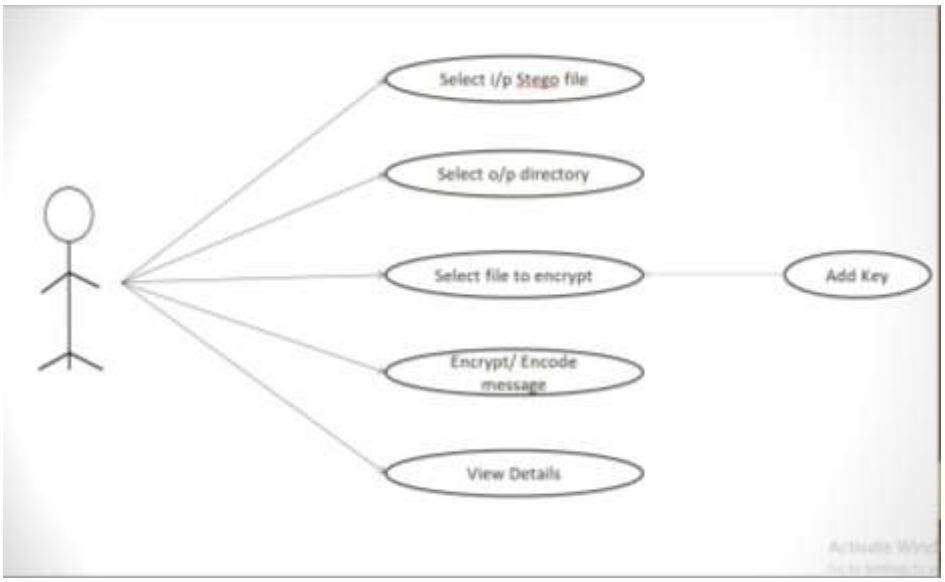


Fig.4.1.Encryption usecase diagram

- To encrypt we select input stego file(image/audio/txt), also choose a particular directory. Now select file which we want to hide data in it and add key to that file.
- Now the encode file is generate and if we view details both encrypted file and original files have no difference.

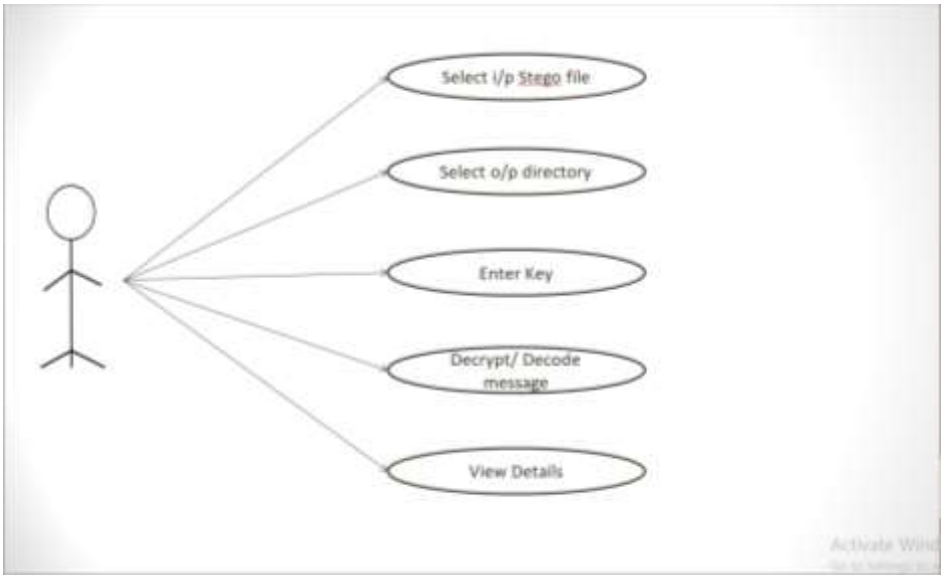
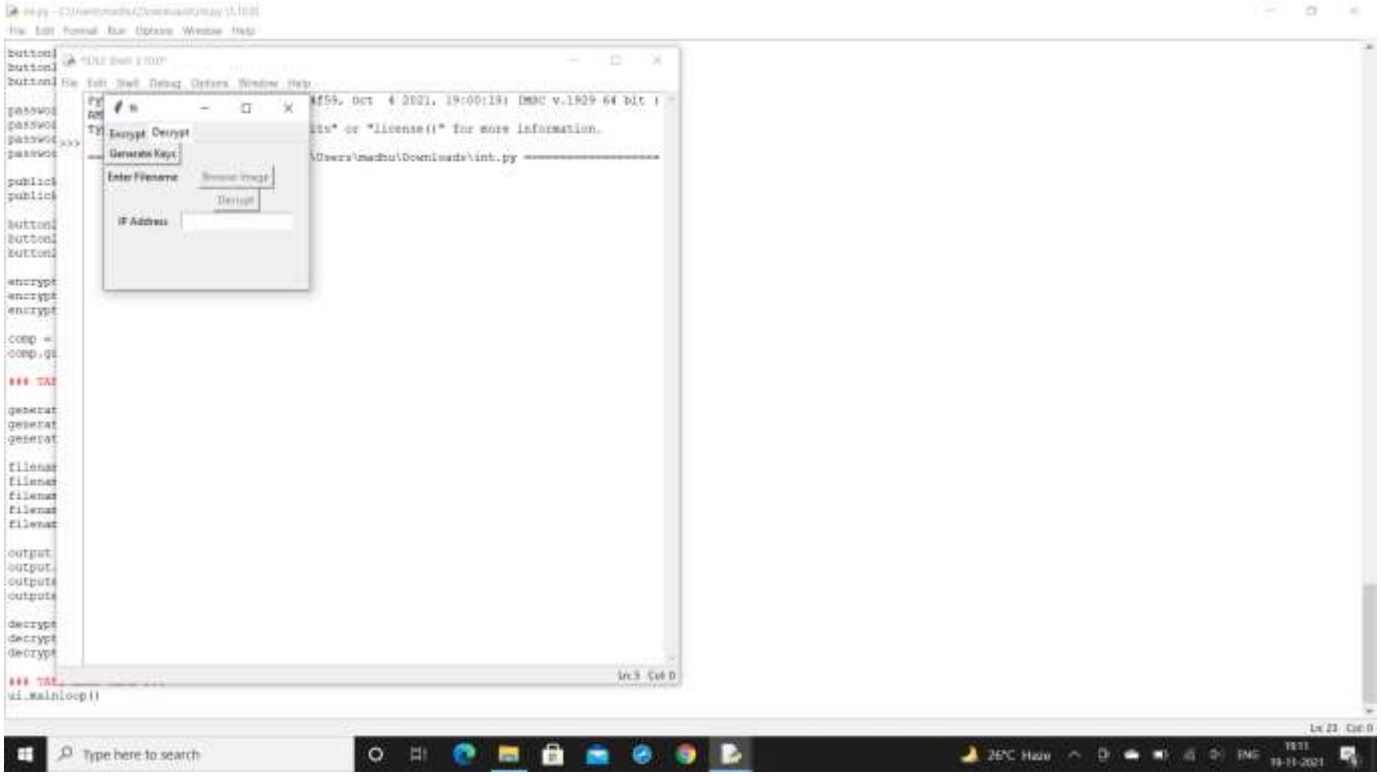


Fig.4.2.Decryption usecase diagram

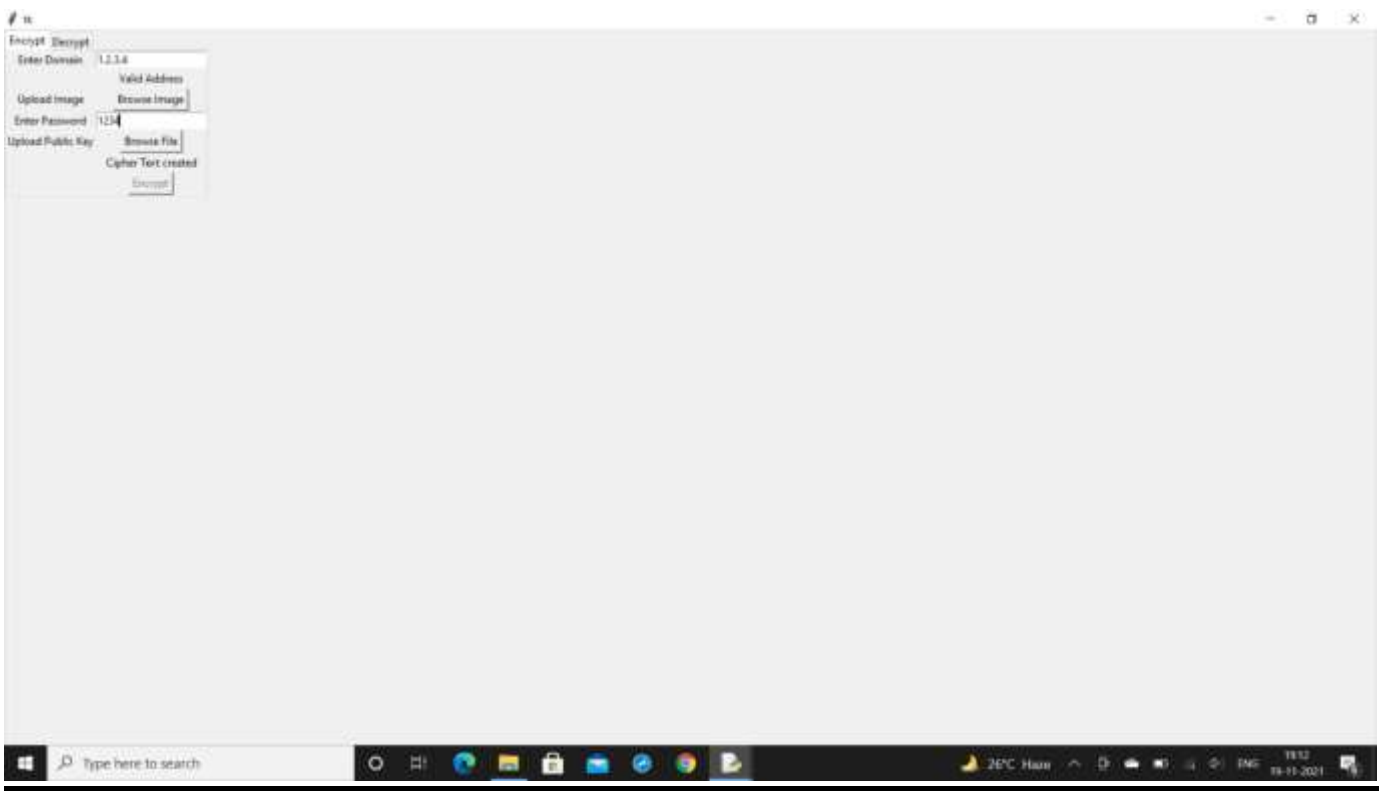
- To decrypt the content from stego file we need to select that particular file and choose output directory.
- Now enter the encryption key, and the encrypted message will appear.

CHAPTER -5 EXPERIMENTAL RESULTS

First, click the produce keys button in the GUI application for encryption and decryption.



Enter the Input i.e valid IP Address.



Upload the image in which the data has to be hidden.

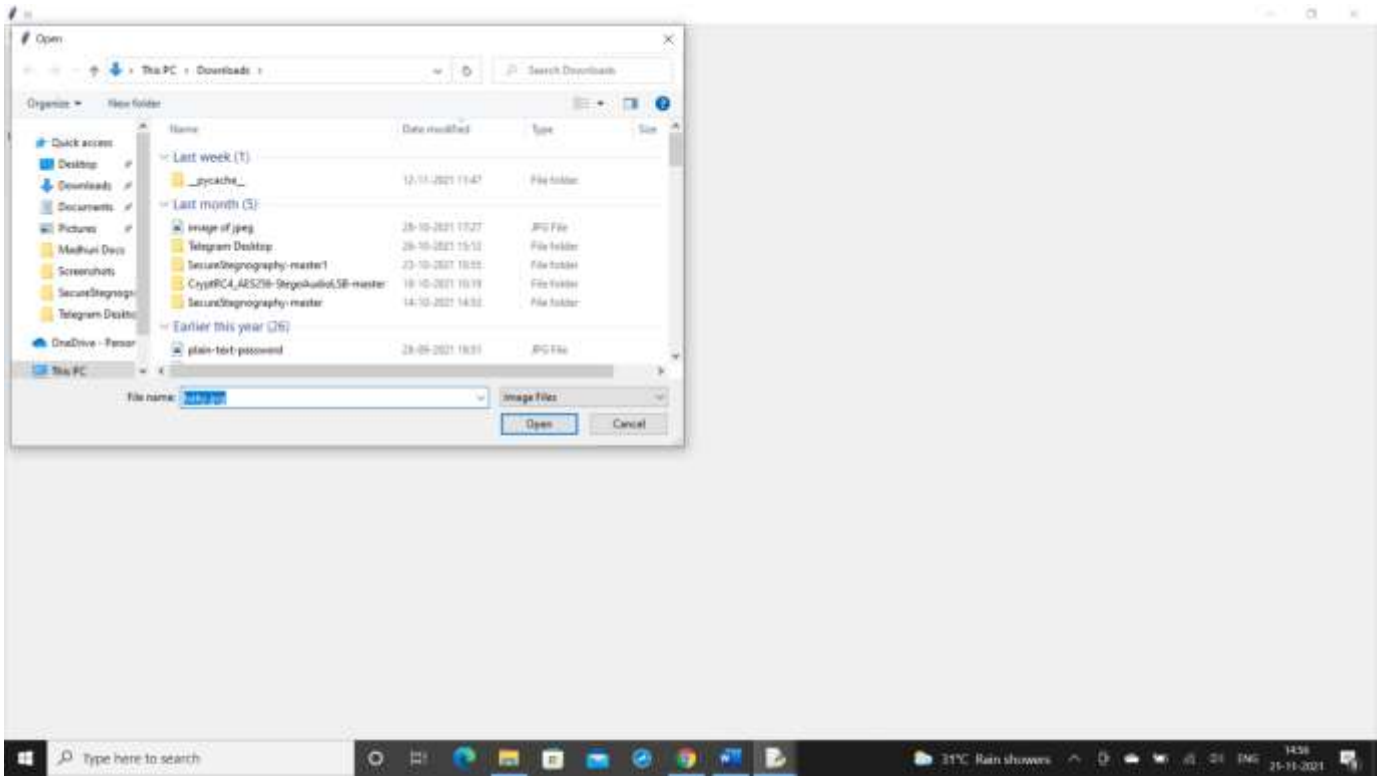
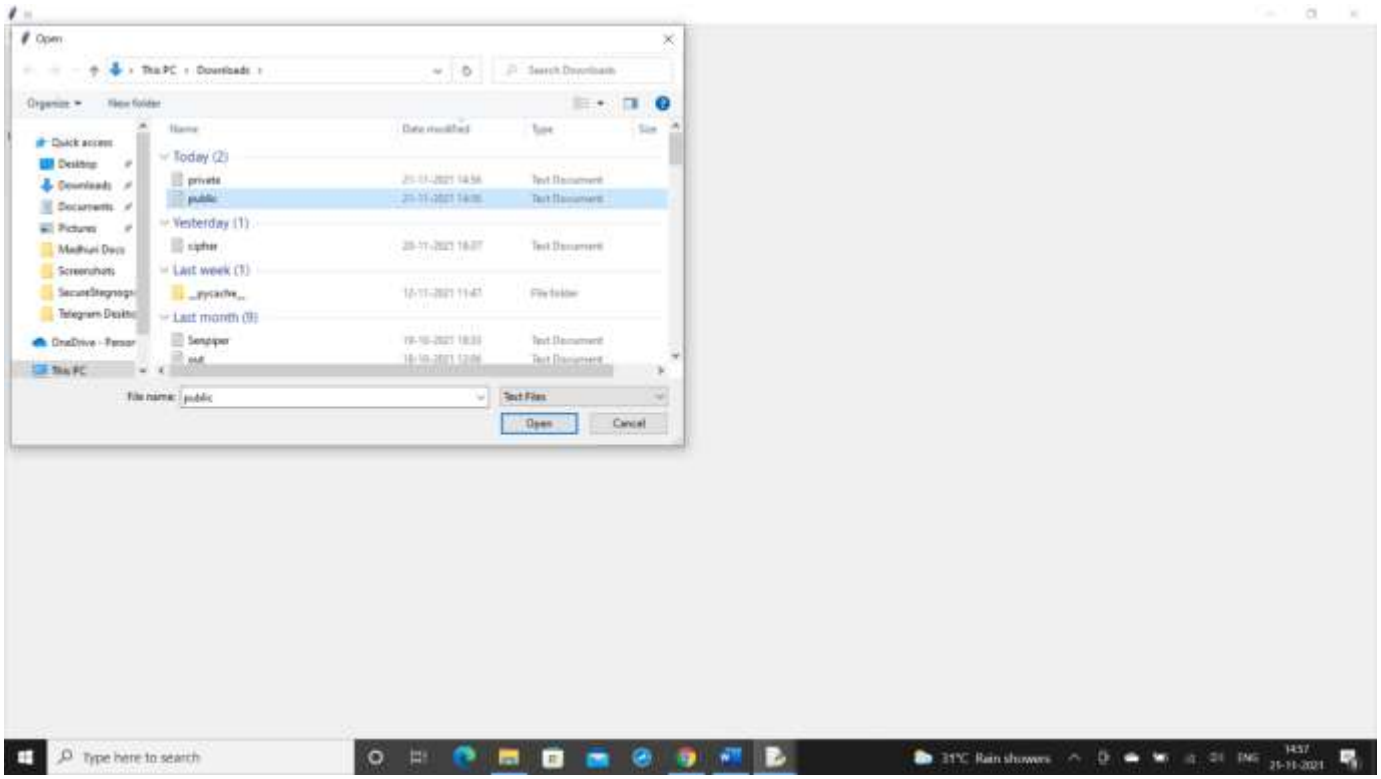


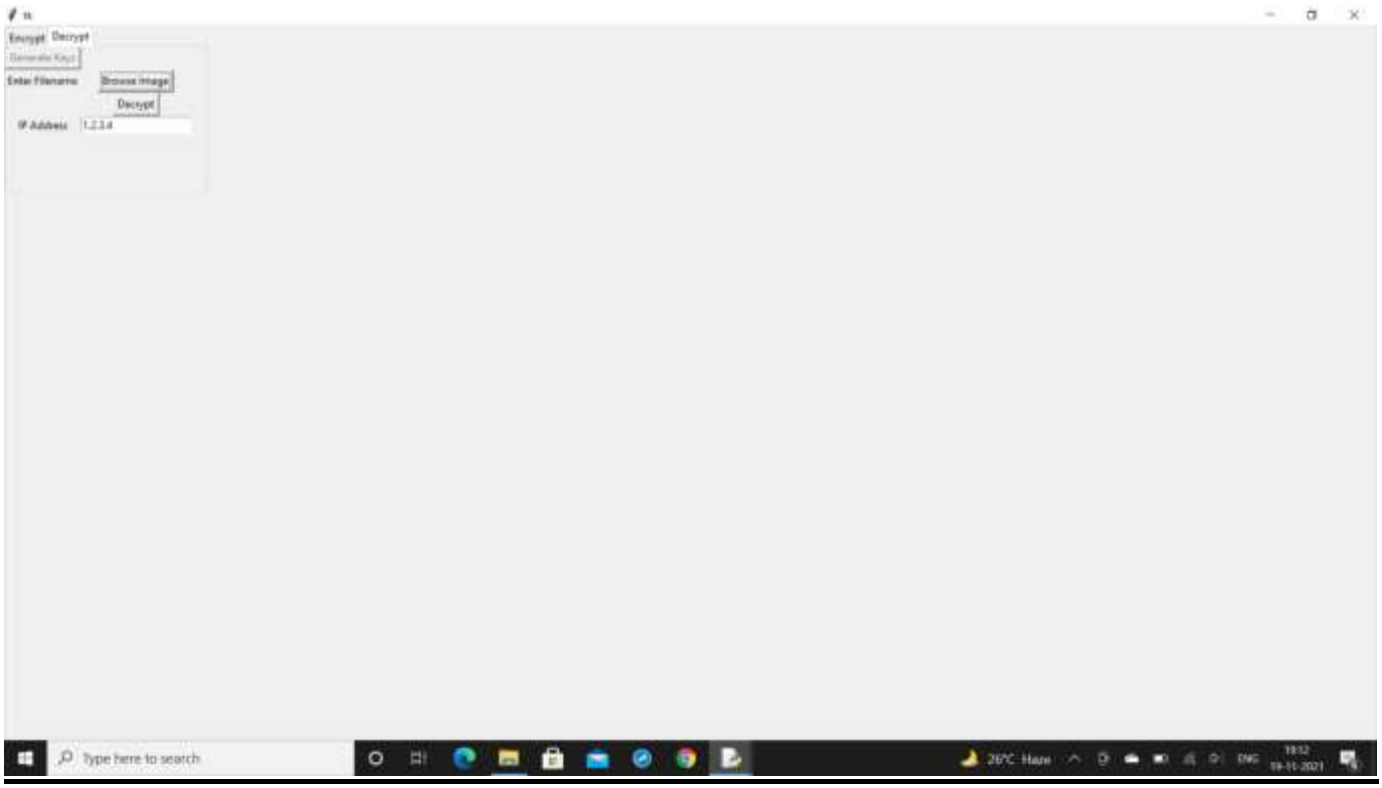
Image used for encryption

Fig.5

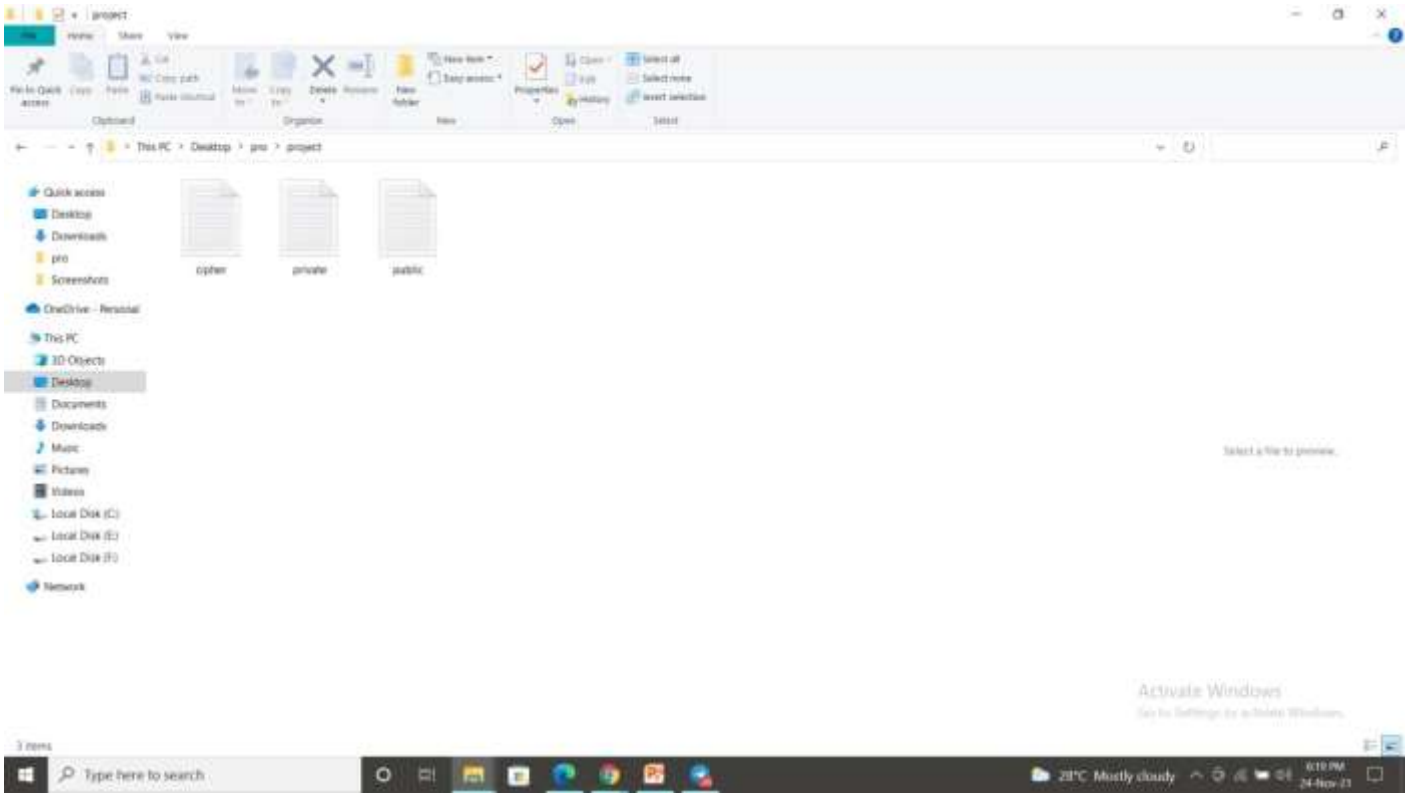
Upload the public key file which we have generated before.

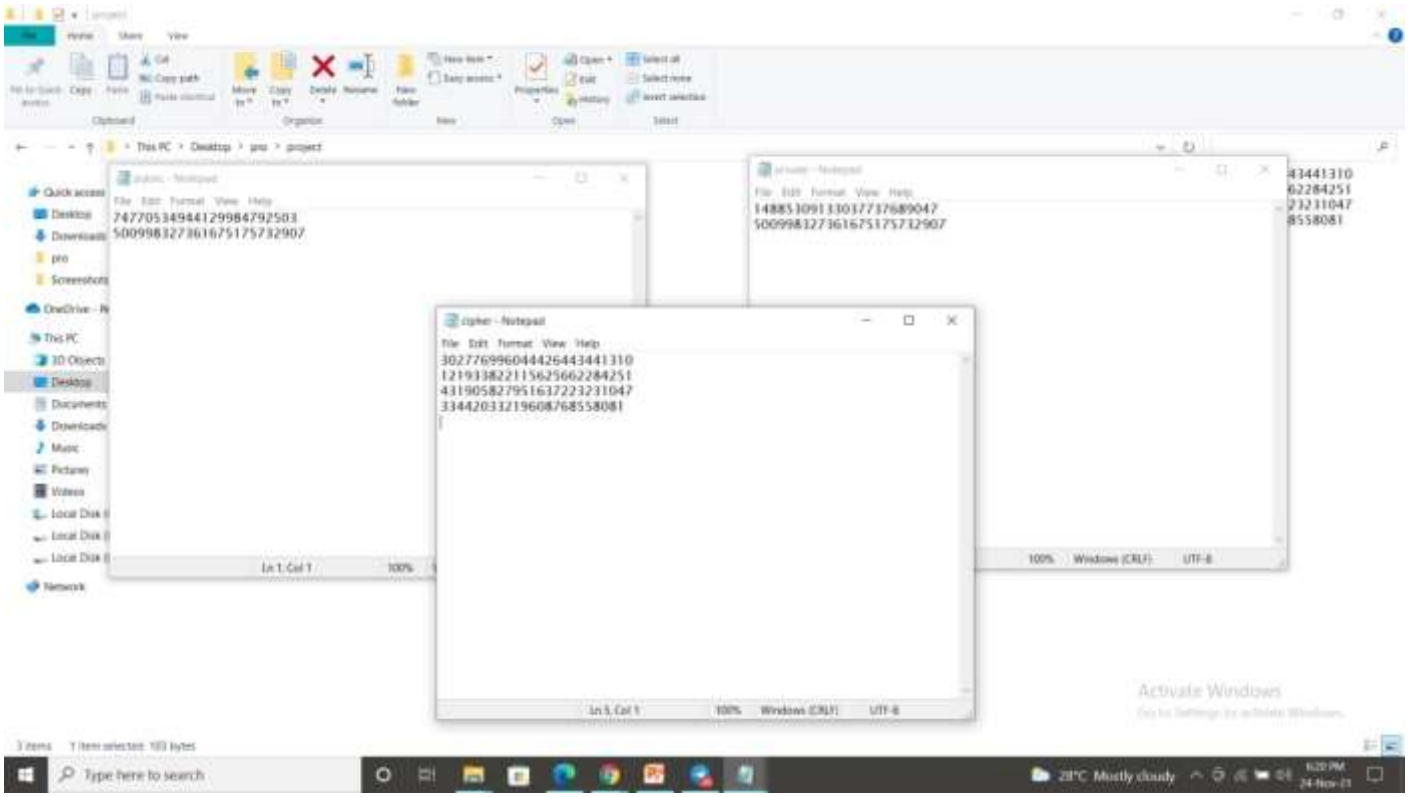


Enter the pass and then click encrypt.

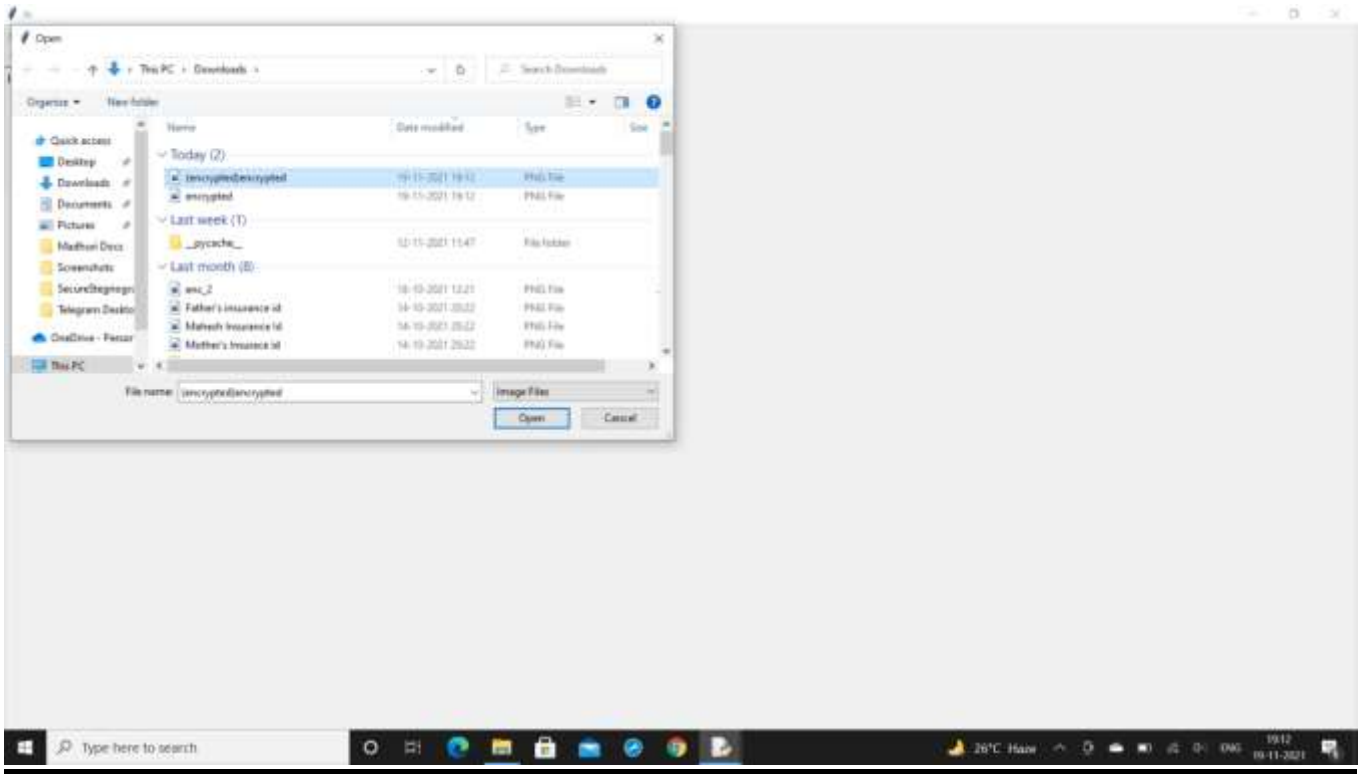


Generated key files

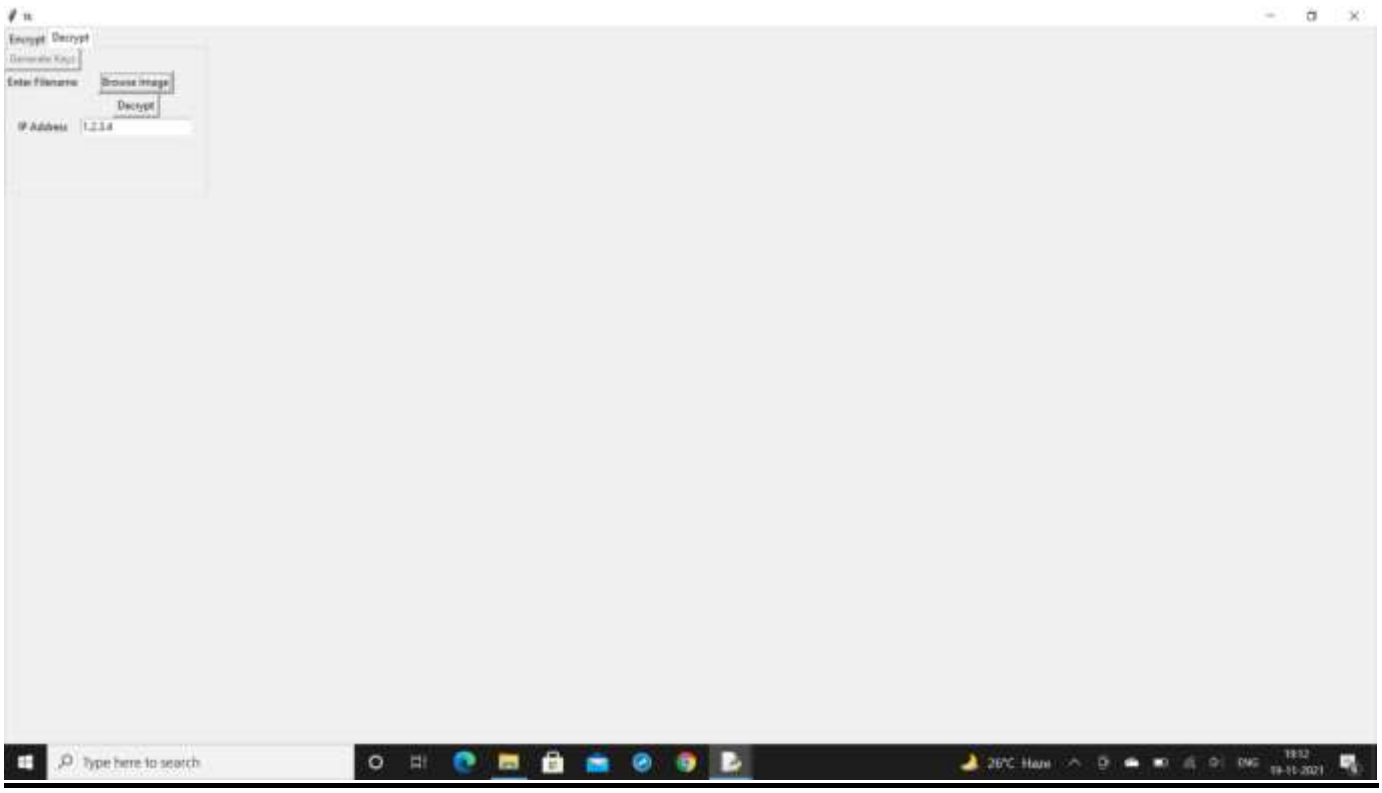




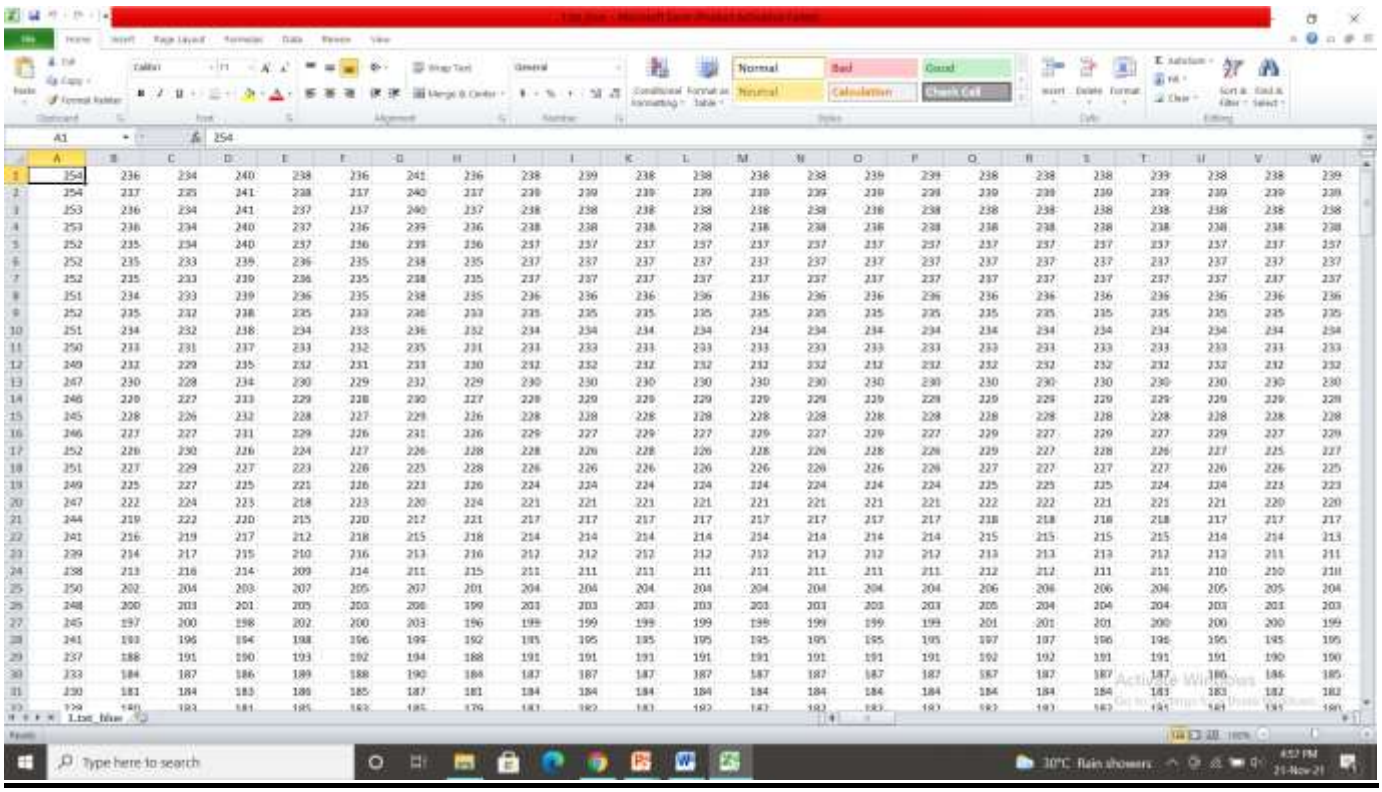
In decryption first upload the encrypted image file.



And then click decrypt the data which we have hidden will be visible.



Flow of execution in Python IDLE.



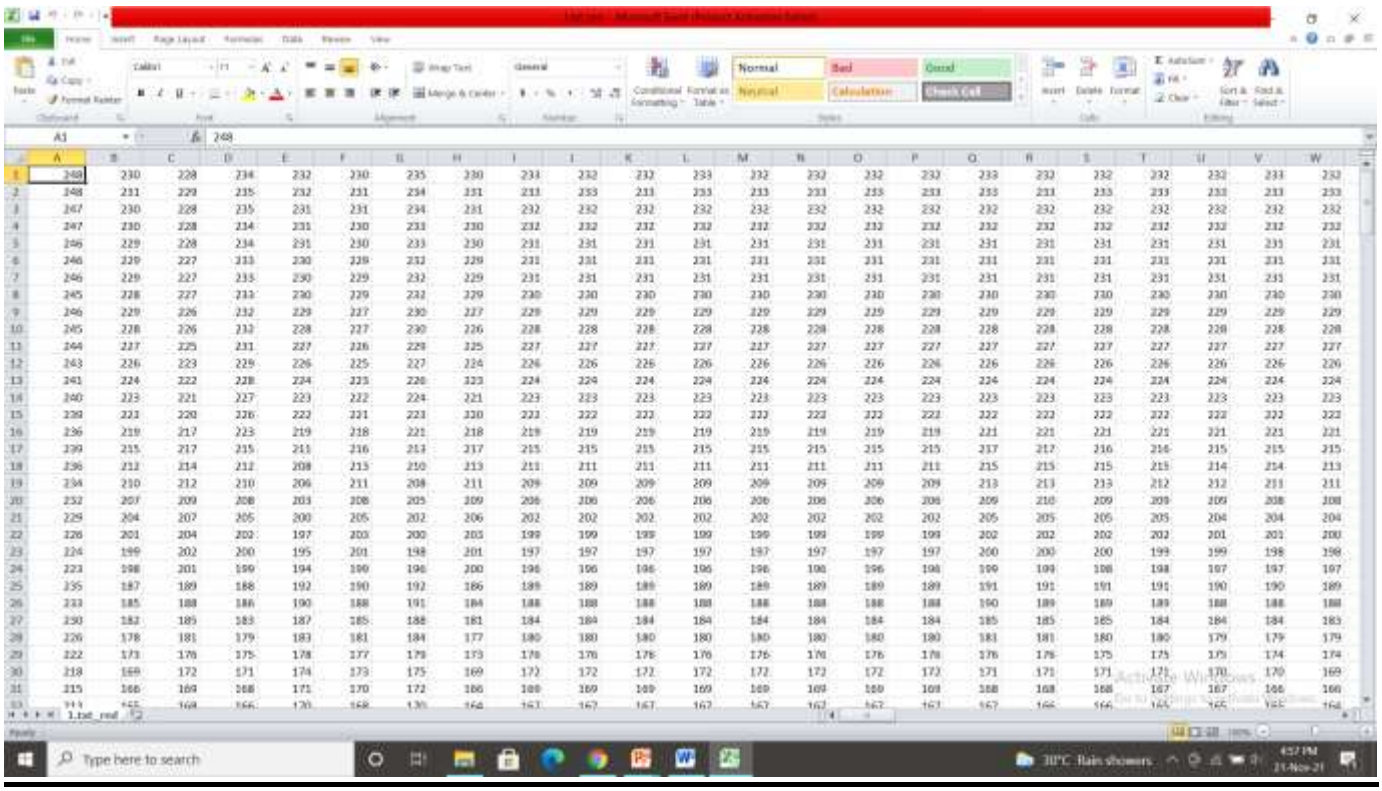
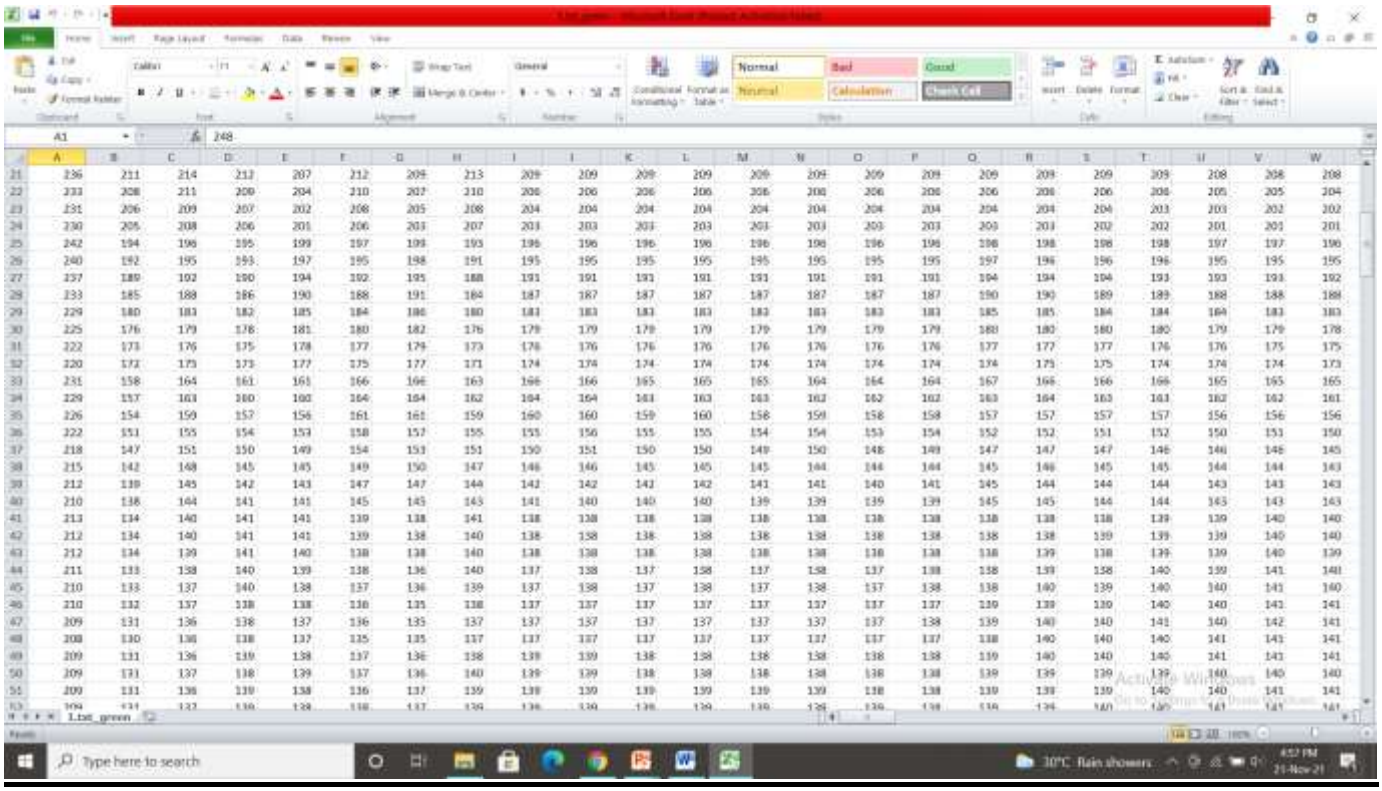


Image Pixel values in XL sheet



CHAPTER-6: DISCUSSION OF RESULTS

- According to the proposed system the results have occurred and the Hiding of data is done successfully.
- First, we use the RSA cryptographic procedure to generate the keys (public key and private key).
- Enter the input data to hide.
- And then we are uploading the image in which we want to hide the data.
- We must upload the public key once we have uploaded the picture.
- Later enter the pass or the secret key and then click encrypt the cipher text is created. If the IP address is incorrect, an error notice appears.
- Then the data is successfully hidden in that image and encryption is done.
- Coming to Decryption Upload the encrypted image from the files in the computer names encrypted image.
- Then enter decrypt and the IP Address which we have entered will be shown.
- We employed the LSB steganography technique so that the image does not display any difference after encryption, making it impossible for anybody to tell that this image is different from the original.

Fig.6.1:Original Image

Fig.6.2:Stego Image

- The execution time is less for AES and RSA than any other cryptograeulc algorithms and so AES is faster and more secure.
- The below shown graph is proof that the time take to encrypt the message/data into stego image using RSA and AES is less when compared with other techniques.

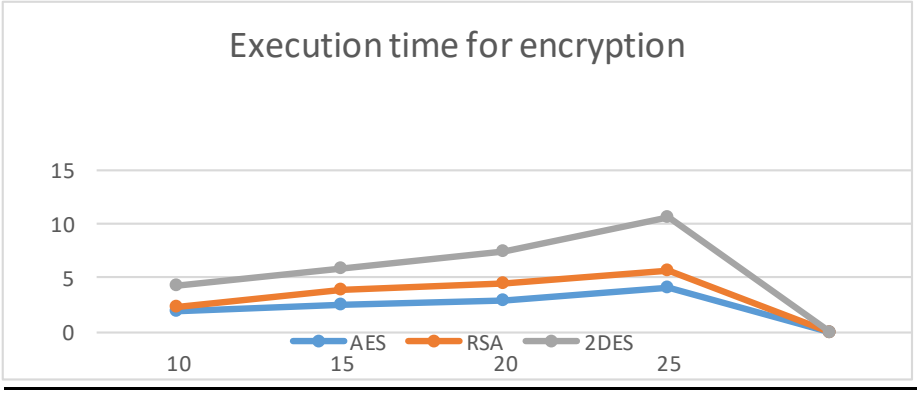


Fig6.3.Graph between RSA,AES and 2DES

- Hence from this graph we can clearly say that AES takes less time for encryption and its computational power is high than DES,3DES.
- We have used 2 different types of cryptograeulc techniques one with symmetric algorithm and another one an asymmetric algorithm here so that it can provide a double layer security for the information.

CHAPTER-7: CONCLUSION

As Nowadays people are depending mostly on network for getting their work done effectively and fastly many attacks are increasing regarding credit cards and money being stolen etc... Due to loss of our personal information these type of systems will be more helpful so that no can easily get attacked. So we used cryptography and steganography to create double layer security in this system. LSB - Steganography is a stego technique where we hide data inside an image by replacing Least significant bit of image with bits of message to be hidden. This technique was not very popular in use hence it is not used widely decryption of the message will be difficult for the attacker. Despite the current theoretical attacks and any potential side-channel attacks, AES and RSA itself remains secure. The RSA algorithm is an asymmetric cryptography algorithm, which means it encrypts data using both public and private keys. AES encrypts a text to a cipher text, which can be decrypted back to the original text by using common private key. Because only two people have access to the private key, an attacker will have a difficult time decrypting it. The encrypted message is created by encrypting plain text with the RSA method. i.e RSA cipher text is now considered as plain text then again encrypted using AES algorithm which result in double encryption. We have used 2 different types of cryptography algorithms which are more faster and provide best and strong security than other algorithms.

CHAPTER-8: REFERENCES

[1]. Nurhayat Varol Tbmyo, Abdalbasit Mohammed Qadir, “A Review Paper on Cryptography”, Turkey, 2019.

[2]. E. Thambiraja Dr. Pauls Engineering College, G.Ramesh Research and Development Centre, Dr. R. Umarani, Sri Sarada college for women, Salem -16,” A Survey on Various Most Common Encryption Techniques”,2016.

[3]. Huang, C.-W., Che-Hao Chiang, Chien-Lun Yen, Yi-Cheng Chen, Kuo-Huang Chang, & Chi-Jeng Chang,”AES Application in image using Different Operation modes”, 2019.

[4]. Ako Muhamad Abdullah , Cyprus," Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt Data”, Eastern Mediterranean University – Cyprus, 2017.

[5]. Akash Nag, “Low-Tech Steganography for Covert Operations”, CSE, Burdwan 713104, WB, India, 2019.

[6]. U. A. Md. Ehasn Ali1 , Md. Sohrawordi1 , Md. Palash Uddin1 1,” A Robust and Secured Image Steganography using LSB and Random Bit Substitution”, Bangladesh, 2019.

[7]. Sulaiman, Rahmat; Kirana, Chandra; Sugihartono, Tri; Laurentinus, ; Panca Juniawan, Fransiskus (2020), Indonesia,” RC4 Algorithm and Steganography to Double Secure Messages in Digital Image”2020 8th International Conference on Cyber and IT Service Management (CITSM).

[8]. Pedro Faria, Member, João Spínola, and Zita Vale, Senior Member,” Aggregation and Remuneration of Electricity Consumers and Producers for the Definition of Demand-Response Programs”, 2019.

[9]. Sofyane Ladgham Chikouche, Noureddine Chikouche Computer Science Department University of M’sila, Algeria,” An Improved Approach for LSB-Based Image Steganography using AES Algorithm”,2017.

[10]. Kusuma, Edi Jaya; Indriani, Oktaviana Rena; Sari, Christy Atika; Rachmawanto, Eko Hari; Setiadi, De Rosal Ignatius Moses," An Imperceptible LSB Image Hiding on Edge Region Using DES Encryption", Indonesia,2017.

[11]. Menon, N., & Vaithiyanathan.V," Triple Layer Data Hiding Mechanism using Cryptography and Steganography", Kerala, 2018.

[12]. Ali Akbar Lubis, Ronsen Purba, Iran Adiputra Pardosi, "COMBINATION OF STEGANOGRAPHY WITH K MEANS CLUSTERING AND 256 AES CRYPTOGRAPHY FOR SECRET MESSAGE", 2019.

[13]. Hadipour, A., & Afifi. R," Advantages and disadvantages of using cryptography in steganography", Iran,2020.

[14]. Gopika Rajan J, R.S.Ganesh," Review of Recent Strategies in CryptographySteganography Based Security Techniques",
India, 2018.

[15]. Bo Bo Oo, May Thu Aung, "Enhancing Secure Digital Communication Media Using Cryptograeulc Steganography Techniques", Myanmar, 2020.

PLAGIARISM REPORT: