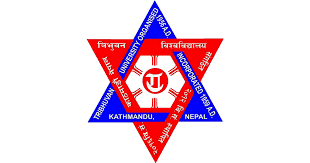
**SOUTHWESTERN STATE COLLEGE**

**TRIBHUVAN UNIVERSITY**

**FACULTY OF HUMANITIES AND SOCIAL SCIENCE**

****

**STEGANOGRAPHY**

**A Project Proposal**

**Submitted to**

**Department of Computer Application**

**In partial fulfillment for the requirements for Bachelor in Computer Application**

**Submitted by**

**Name: Akriti Shrestha**

**TU Reg. No: (6-2-530-26-2019)**



Ref:…………………………….

Date:…………………………

# LETTER OF APPROVAL

The Project “**Steganography”** proposed by **Akriti Shrestha** for the partial fulfillment of the requirements for bachelor in Computer Application of sixth semester we request to approve for further development.

**Proposal Evaluation Committee**

**1. ----------------------------**

**2. ---------------------------**

**………………………**

**Program Incharge** Kiran Ghimire (BCA Department)

ii

# ABSTRACT

In the age of the Internet where everything is virtual, security and privacy are issues that come up repeatedly. Steganography is one of the methods that aims to solve this problem. It can be defined as a method that conceals data within an ordinary file to avoid detection. It creates a stego object that is the encrypted version of the cover file of a secret image or data to be transmitted.

Steganography is a method which conceals data within an ordinary file to avoid detection. Steganography includes the encryption and decryption process, where the secret image or data to be transmitted is hidden inside the cover file and encrypted output is called stego object. Steganography can be done using any digital format like image, video, or audio etc.

The aim of this project is to improve the data hiding ability through various methods.

**Keywords:** *Steganography, AES, LSB, Encryption, Python, Imperceptibility*

iii

**Table of Contents**

[LETTER OF APPROVAL 3](#_Toc131871568)

[ABSTRACT 4](#_Toc131871569)

[LIST OF FIGURES 6](#_Toc131871570)

[LIST OF TABLES 7](#_Toc131871571)

[CHAPTER 1: INTRODUCTION 1](#_Toc131871572)

[1.1 Background 1](#_Toc131871573)

[1.2 PROBLEM STATEMENT 2](#_Toc131871574)

[1.3 OBJECTIVES 2](#_Toc131871575)

[1.4 SCOPE 3](#_Toc131871576)

[CHAPTER 2: LITERATURE REVIEW 4](#_Toc131871577)

[2.1 Artificial Intelligence 4](#_Toc131871578)

[2.2 Existing systems: 4](#_Toc131871579)

[2.3 Literature Review 6](#_Toc131871580)

[CHAPTER 3: REQUIREMENT ANALYSIS AND METHODOLOGY 8](#_Toc131871581)

[3.1 Functional Requirements 8](#_Toc131871582)

[3.1.1 Client Application 8](#_Toc131871583)

[3.1.2 Server Application 8](#_Toc131871584)

[3.2 Non-Functional Requirements 8](#_Toc131871585)

[3.3 METHODOLOGY 10](#_Toc131871586)

[3.3.1Waterfall Model 10](#_Toc131871587)

[3.4 Gantt chart 11](#_Toc131871588)

[3.5 Tools and Technologies 12](#_Toc131871589)

[3.5.1 Hardware and Software Requirements 12](#_Toc131871590)

[CHAPTER 4: IMPLEMENTATION 13](#_Toc131871591)

[4.1 Use Case Diagram 13](#_Toc131871592)

[4.2 Algorithm used in my project 13](#_Toc131871593)

[4.2.1 AES Algorithm 13](#_Toc131871594)

[4.2.2 LSB Algorithm 16](#_Toc131871595)

[CHAPTER 6: CONCLUSION AND RECOMMENDATION 18](#_Toc131871596)

[6.1 Conclusion 18](#_Toc131871597)

[6.2 Recommendation 18](#_Toc131871598)

[REFERENCES 19](#_Toc131871599)

iv

# LIST OF FIGURES

[**Figure 1 : Artificial Intelligence** 4](#_Toc131870870)

[**Figure 2 : Use Case Diagram** 13](file:///C:\Users\DELL\Downloads\Steganographyy.docx#_Toc131870871)

[**Figure 3 : Schematic of AES Structure** 14](file:///C:\Users\DELL\Downloads\Steganographyy.docx#_Toc131870872)

[**Figure 4 : First Round Process of AES** 15](file:///C:\Users\DELL\Downloads\Steganographyy.docx#_Toc131870873)

v

# LIST OF TABLES

[**Table 1: Gantt Chart** 11](file:///C:\Users\DELL\Downloads\Steganographyy.docx#_Toc131870906)

[**Table 2 : Software and Hardware Requirements** 12](#_Toc131870907)

vi

# CHAPTER 1: INTRODUCTION

## Background

One of the reasons that intruders may be successful is since the maximum statistics they gather from a gadget is in a shape that they are able to study and understand. Intruders might also additionally monitor the statistics to others, alter it to misrepresent a character or an organization, or use it to release an attack. One possibility to this hassle is, via the use of steganography.

Steganography is a way of hiding statistics in digital media. In contrast to cryptography, it is not stopping others from understanding the hidden statistics, however it is more so to preserve others from wondering that the statistics even exist.

The encryption section is based on the AES algorithm which is a popular symmetric-key 12 encryption algorithm. As for the message hiding, it is based on the LSB algorithm that embeds binary value into the image pixel using mathematical calculations.

In the development of the system, Python was used for the Backend portion whereas React JS was used for the Front-end portion. The system uses modules such as encode, decode, PIL, tkinter for its functionality. In the initial phase of the development, the UI of the overall system was designed on Figma and was later executed using React JS and CSS. These two ends were later connected using FastAPI at backend and Axios at frontend.

Steganography will become more crucial as greater humans become part of the cyberspace revolution. Steganography is the artwork of concealing statistics in methods that prevent the detection of hidden messages. Steganography consists of an array of secret communique techniques that cover the message from being visible or discovered.

Due to advances in ICT, the maximum of the records is stored electronically. Consequently, the safety of records has emerged as an essential issue. Besides cryptography, steganography may be hired to secure information. In cryptography, the message or encrypted message is embedded in a virtual host earlier than passing it via the network, for this reason, the lifestyles of the message are unknown. Besides hiding statistics for confidentiality, this technique of records hiding may be prolonged to copyright safety for virtual media: audio, video, and images.

The increasing possibilities of modern communication require special security measures, especially in the computer network. Network security is becoming increasingly important the more data is exchanged on the Internet against unauthorized access and use, which has led to an explosive growth in the information field that is hidden**.**

Information hiding is an emerging research area, which encompasses applications such as copyright protection for digital media, watermarking, fingerprinting, and steganography.

In watermarking applications, the message holds information such as owner identification and a digital time stamp, which is typically used for copyright protection.

Fingerprint, the owner of the data record embeds a serial number that uniquely identifies the user of the data record. This is done in addition to the copyright information to trace back unauthorized use of the data set to the user.

Steganography hides the secret message within the host data set and presence imperceptible and is to be reliably communicated to a receiver. The host data set is purposely corrupted, but in a covert way, designed to be invisible to an information analysis.

## PROBLEM STATEMENT

No projects or applications are 100% productive, to overcome the flaw in existing application new projects and innovations are carried out. Following are the problems that creators are facing while creating contents:

* Difficulty in protecting one's intellectual property.
* Difficulty in sending secret messages to destinations without any risks of exposure.

## OBJECTIVES

In the case for developing STEGANOGRAPHY, following are the objective that this project aims to achieve:

* To hide the presence of a message.
* To encrypt messages using AES algorithm before embedding them in the picture.
* To carry out secret communications.

## SCOPE

Steganography is the technique for hiding information in images. Cryptic messages are further hidden in images thereby further encrypting them and making them invisible altogether. One of its various applications is in Digital Watermarking.

* The scope of the project is to limit unauthorized access and provide better security.
* To meet the requirements, we have used a simple and easy approach of Steganography along with the concept of encryption-decryption.
* The project finds suitable algorithms for encrypting the data and embedding it into an image, and later dig out the information and decrypt it into the original text.

# CHAPTER 2: LITERATURE REVIEW

### Artificial Intelligence

John McCarthy, one of the great innovator who is also known as the Father of Artificial Intelligence states that “the science and engineering of making intelligent machines”.AI is technology that can use and evaluate equations and programming of know ledge to carry out operations, challenges and work out how to respond to a range of situations with and without management. Artificial Intelligence (AI) research utilizes tools from numerous fields like software engineering, brain science, neuroscience, intellectual science, semantics, financial matters, control hypothesis etc. (ScienceDaily)



**Figure 1 : Artificial Intelligence**

## 2.2 Existing systems:

**2.2.1Quick Stego**

QuickStego is software that lets you to hide text in pictures so that only other users of QuickStego can explore and read the hidden secret messages. Once text that we choose is hidden in an image the saved picture is still a 'picture', it will load just like any other image and appear as it did before. The image can be saved, emailed, uploaded to the web as before, the only difference will be that it contains hidden text. QuickStego use strong encryption software that offer to design to hide the data of encryption, giving you excellent ease of use at the same time, giving you encryption software that you can trust. QuickStego allows you to secure single and multiple files also. It also can secure folders, sub-folders, passwords and emails. It is quickly and easily to implement, and performs most operations with a single mouse click within familiar. There are no limitations on file type, QuickStego encrypts every kind of file format, whether it is text, video, picture, document or audio - any type of file, on USB, floppy, thumb/flash or hard drives. QuickStego is easy to use.

**2.2.2 Hide in Picture (HIP)**

Hide in Picture is software that allows you to "hide" any kind of file inside the standard bitmap pictures. The pictures look like normal images to everyone, so people will not suspect anything when they see the image. This is called steganography. You can also use a password to hide your files, and only those who know the password are able to view them. The person who doesn’t know the password cannot even be sure there is something hidden in an image.

#### Features of Hide in Picture

* **Encryption** - All data will be encrypted before written to a picture. It is to increase the security. HIP offers several encryption algorithms that you can choose from it; all of them are considered very secure, so you don't have to worry about it unless you have a specific reason for wanting to use a specific algorithm. When view a file, HIP tries using all the available algorithms to find the correct data.

**Transparent color support** - One color of the picture may be set as 'transparent'. Nothing will be stored in areas of this color palette. This can be useful to all. For an example when hiding a file inside an image from a Web page, its transparent area will remain as before. To retrieve a hidden file using this option, you must set the transparent color to the same used when hiding it.

**2.2.3 Chameleon**

Chameleon is steganography software that allows users to hide confidential or important files within a standard digital picture. This is particularly useful in distribute covert communications over the Internet. Images will be embedded with secret documents, for example is it may be transmitted as an email attachment or it is posted on a web forum or bulletin board. The main feature of Chameleon is its adaptive encoding algorithm which optimizes the use of hiding space in a particular cover image. In bitmaps, steganography is usually performed by replacing the least-significant bit (LSB) of the color values of each pixel with the data bits of the file or message to be hidden. Since LSBs represent only a small portion of the actual color values, such changes in an image are often invisible to the human eye. Consider, for example, a 640x480 true-color bitmap. True color images are composed of red, green, and blue color channels. In each pixel, the intensity or color value for each channel is represented by an 8-bit number. Using the LSBs of the three color values of each pixel would generate a hiding space of 115,200 bytes (or 921,600 bits), 1/8 of the total image size.

## 2.3 Literature Review

There have been various approaches towards STEGANOGRAPHY and some of the methods are described below:

T. Morkel et.al [1] have presented an overview of image steganography, its uses, and techniques for hiding the secret text in an image. The requirements of a good steganography algorithm have been discussed.

Pramod Choudhary et.al [2] proposed LSB based method for hiding the key image. Text is embedded within the cover image by rotating the image. Different angular rotations are often applied for hiding the text without losing the duvet image characteristics. The proposed algorithm is straightforward and efficient to make sure the knowledge hiding and may be used for various purposes including, storing patients’ history using his image as cover image, storing significant information about objects in objects image in cloud-based environment.

Ali Ahmed et.al [3] developed an improvement model for securing and hiding text messages into grey scale images. Two-stage processes for encrypted text messages using double XOR binary based operation were performed. Hiding encrypted text was made on LSB of each image pixel byte.

Mohammad Goli et.al [4] proposed a report on three methods including least significant bit (LSB), discrete wavelet transform (DWT) and singular value decomposition (SVD) for the study of image-inimage steganography. Each method has advantages and disadvantages.

Advantages of LSB include high capacity and simplicity, however, as for the disadvantages, it is vulnerable against attacks such as cropping, salt and pepper noise, and compression. Therefore, the other two methods, DWT and SVD, which are robust against the mentioned attack, are used as steganography in frequency domain. Among the methods used in this study, SVD least affects the host image, making it more suitable as compared with its other two counterparts.

Vinita Gupta et.al [5] have presented a review on Image Watermarking for good robustness. In this paper they have discussed numerous factors used in watermarking, properties and application area where watermarking technique need to be used.

# CHAPTER 3: REQUIREMENT ANALYSIS AND METHODOLOGY

## 3.1 Functional Requirements

### 3.1.1 Client Application

The client application is the link between the user and the server application. Its task is to gather information from the users and to allow user to play movies. The information is sent to the server application, where it is stored, and later used to produce recommendations.

In addition, the information is used to measure recommender precision. This allows for investigation of how precision is how precision is influenced by different recommender strategies.

The requirement for the client application:

* + - * Request recommendation

### 3.1.2 Server Application

Application server frameworks are software frameworks for building application servers. An application server framework provides both facilities to create web application and server environment to run them.

The requirements for the server application are:

* + - * Handle Recommendation requests
      * Store evaluations
      * Recommend using algorithm

## 3.2 Non-Functional Requirements

Non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. They can contrasted with functional requirements that define specific behavior or function. The plan for implementing functional requirements is detailed in the system design.

Non-functional requirements specify how the system should do it. The non- functional requirement holds the following:

* + - * 1. Perceptual Transparency
        2. Tamper–resistance
        3. Robustness
        4. Reliability

## 3.3 METHODOLOGY

The methodologies that are going to be used in this project is given below:

### 3.3.1Waterfall Model

This project was based on the Waterfall model which is a software model in which each phase is dependent on the previous phase. These phases are laid in a sequential order and each phase must be completed before moving on to the next one.

This model was considered because of the size of the application and the stability of the requirements.

The Phases of waterfall software development are as follows:

**Design:** During the design phase of development on the basis of the previously listed requirements list the UI flow was created and the UI design of the system was created. In this phase the structure that is suitable for the implementation of the project was created.

**Implementation:** In this phase on the basis of the design, the front end and backend of the system were programmed and integrated.

**Testing:** Testing phase is the phase in the system design model where unit testing and system testing were carried out. Test cases were created and edge cases were also analyzed and documented in this phase.

**Maintenance:** During the maintenance phase it was made sure that the application was functioning properly and the defects encountered by the user were noted.

## 3.4 Gantt chart

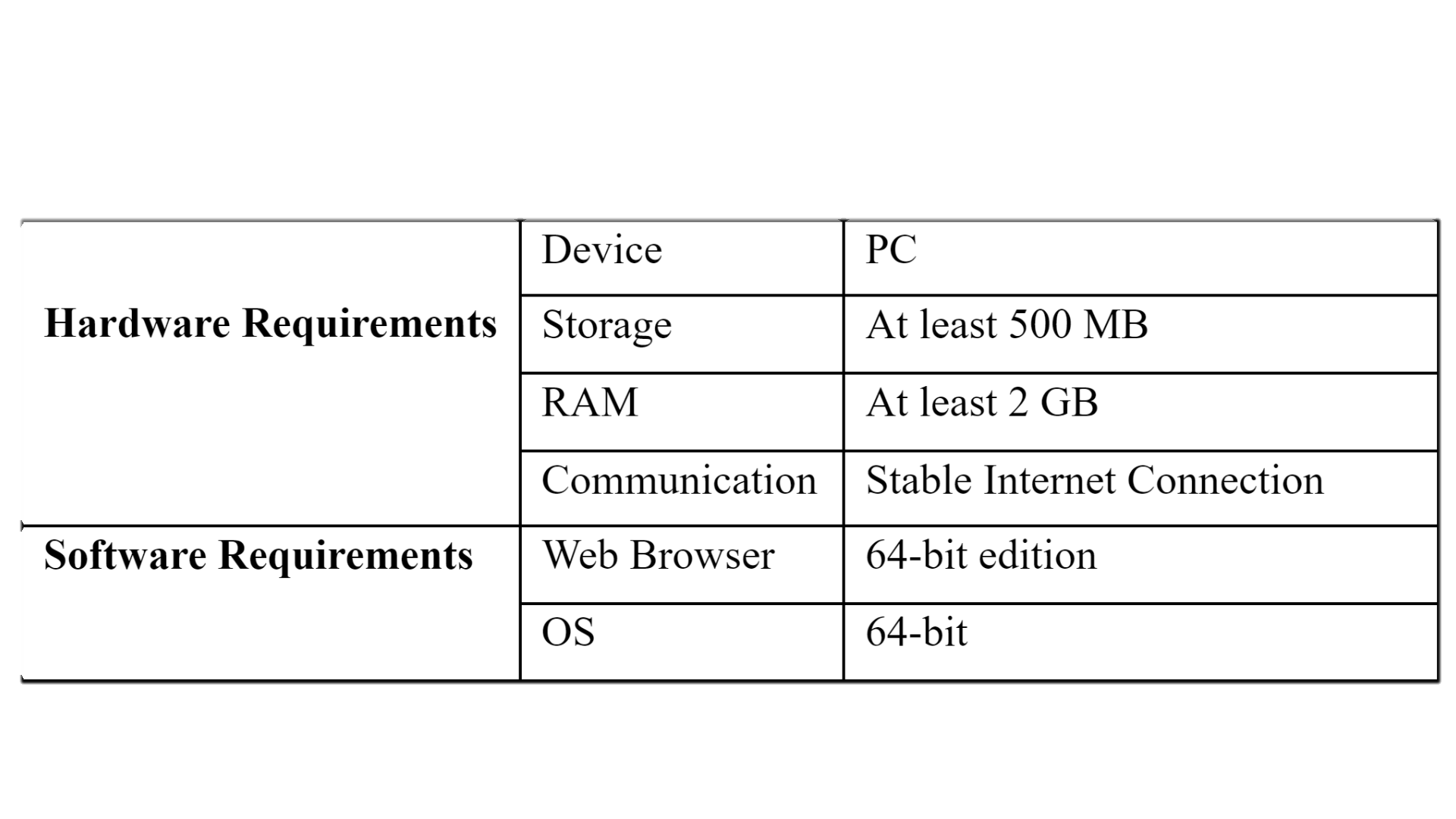
**Table 1: Gantt Chart**

## 3.5 Tools and Technologies

### 3.5.1 Hardware and Software Requirements

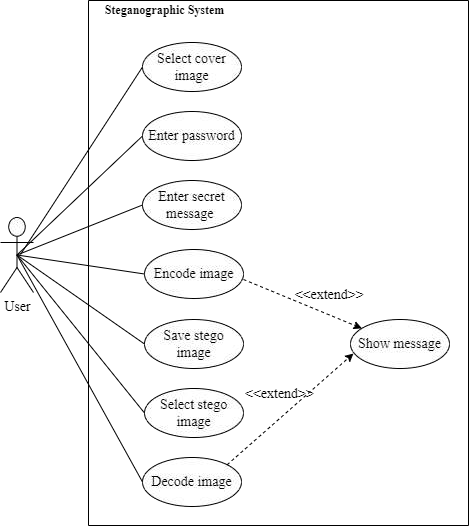
The proposed project would use a pre-built framework, upgrade the existing system and the project would be platform independent. And following all the standard guidelines for projects, we do not expect there to be any technical problem for the proposed project.

**Table 2 : Software and Hardware Requirements**



# CHAPTER 4: IMPLEMENTATION

## 4.1 Use Case Diagram



**Figure 2 : Use Case Diagram**

## 4.2 Algorithm used in my project

### 4.2.1 AES Algorithm

Advanced Encryption Standard (AES) Algorithm is a replacement of DES (Data Encryption Standard) Algorithm. It is a popular and widely populated symmetric key 12 encryption algorithm (uses the same key for both encryption and decryption). This algorithm is found six times faster than triple DES which was designed to overcome the drawbacks of DES Algorithm but was later found slow. [6]

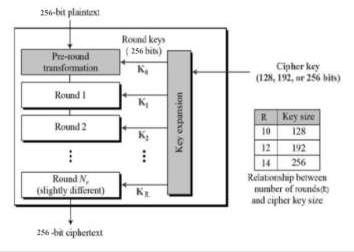
**Operation of AES**

AES is an iterative rather than Feistel cipher. It is based on ‘substitution–permutation network’. It includes of a series of linked operations, some of which involve replacing inputs by specific outputs (substitutions) and others involve shuffling bits around (permutations).

Interestingly, AES performs all its computations on bytes rather than bits. Hence, AES treats the 128 bits of a plaintext block as 16 bytes. These 16 bytes are arranged in four columns and four rows for processing as a matrix −

Unlike DES, the number of rounds in AES is variable and depends on the length of the key. AES uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. Each of these rounds uses a different 128-bit round key, which is calculated from the original AES key.

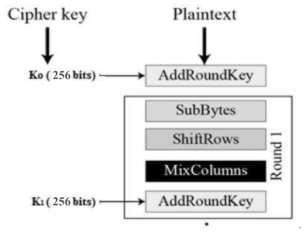
The schematic of AES structure is given in the following illustration –



**Figure 3 : Schematic of AES Structure**

**Encryption Process**

Here, we restrict to description of a typical round of AES encryption. Each round includes of four sub-processes. The first-round process is depicted below−



**Figure 4 : First Round Process of AES**

**Byte Substitution (Sub Bytes)**

The 16 input bytes are substituted by looking up a fixed table (S-box) given in design. The result is in a matrix of four rows and four columns.

**Shift rows**

Each of the four rows of the matrix is shifted to the left. Any entries that ‘fall off’ are re- inserted on the right side of row. Shift is carried out as follows –

* First row is not shifted.
* Second row is shifted one (byte) position to the left.
* Third row is shifted two positions to the left.
* Fourth row is shifted three positions to the left.
* The result is a new matrix consisting of the same 16 bytes but shifted with respect to each other.

**MixColumns**

Each column of four bytes is now transformed using a special mathematical function. This function takes as input the four bytes of one column and outputs four completely new bytes,

which replace the original column. The result is another new matrix consisting of 16 new bytes. It should be noted that this step is not performed in the last round.

**Addroundkey**

The 16 bytes of the matrix are now considered as 128 bits and are XORed to the 128 bits of the round key. If this is the last round, then the output is the ciphertext. Otherwise, the resulting 128 bits are interpreted as 16 bytes and we begin another similar round.

**Decryption Process**

The process of decryption of an AES ciphertext is like the encryption process in the reverse order. Each round consists of the four processes conducted in the reverse order –

* Add round key
* Mix columns
* Shift rows
* Byte substitution

Since sub-processes in each round are in reverse manner, unlike for a Feistel Cipher, the encryption and decryption algorithms need to be separately implemented, although they are very closely related.

### 4.2.2 LSB Algorithm

Firstly, each character of secret message and each pixel of cover image are converted into binary values. The user must input stego-key as the password (stego-key is used to embed the secret message in a cover file). After inserting secret message into cover image file, the resulting stego-image is sent to the receiver through the desired communication channel. While defining the starting point of embedding LSB, the stego-key is firstly collected from the user. The summation of the ASCII value of each character of stego key is calculated and then the average of those characters value is computed. While substituting the secret message into LSB of cover image, the first LSB position is chosen according to the calculated average value of input stego-key characters. Then the substitution processing will continue until the end of secret message. [7]

**A. The embedding algorithm at the sender side**

* 1. Get the input cover image and secret message.
  2. Accept the stego-key from the user and calculate average value of them.
  3. Convert each character of secret message and each LSB bit of cover image (R channel) from the position of average of stego-key.
  4. Substitute the LSB bit of cover image (R channel) with binary values of secret message with respect to the starting point until the end of secret message.
  5. Insert the end character value at the end of secret message.
  6. Calculate the PSNR, SNR of original and resulting images.
  7. Send a stego-image to the receiver.

**B. The extracting algorithm at the receiver side**

* 1. Get the input stego calculate average value
  2. Load the stego-image that is sent from the sender.
  3. Extract each of LSB bit from the stego image until to find out the end bit.
  4. Reconstruct the collecting LSB bits from the stego-image.
  5. Transform the LSB bits to correspondent characters.

# CHAPTER 6: CONCLUSION AND RECOMMENDATION

### Conclusion

“Steganography” as the name suggests is a Steganography system that encrypts messages and hides them inside images. The system can hide both text as well as images inside of the image provided by the user. It uses the AES algorithm for the encryption of texts and images, and the LSB algorithm for Steganography. The main objective of this system was to provide an upgrade to cryptography, through inability to acknowledge the presence of hidden message, inside of an image.

The system was developed using Python as the main backend language and React JS for front end. Using modules such as encode, decode, PIL, tkinter, etc the main working for the system was constructed. As for the front end, the UI of the system was first designed using Figma and later shaped using React JS. Finally, using FastAPI on Python and Axios on React we integrated the two ends.

After the completion of the project, system testing was conducted. These led to the revelation of the limitations our system had. One of them being, that the secret image that we are trying to hide must be of lower resolution than that of the …

#### Future Enhancements

Some of the Future enhancements that can be made in the system are:

* + - The system can encompass the feature of hiding Audios as well as Videos inside Image.
    - Can be further expanded to working on mobile apps as well.

### Recommendation

* + - It can be used by Content creators to claim their media by watermarking.

# REFERENCES

1. T. Morkel, J.Ellof and M.Olivier, “AN OVERVIEW OF IMAGE STEGANOGRAPHY,” *Proceedings of the Fifth Annual Information Security South Africa Conference (ISSA2005),* 2005.
2. P. Choudhary, T. S. Jaiswal and P. K. Gupta, “IMAGE STEGANOGRAPHY BASED ON LSB TECHNIQUE,” *International Research Journal of Modernization in Engineering Technology and Science,* vol. 02, no. 05, 2020.
3. A. Ahmed and A. Ahmed, “A Secure Image Steganography using LSB and Double XOR,” *IJCSNS International Journal of Computer Science and Network Security,* vol. 20, p. 139, May 2020.
4. M. S. Goli and A. Naghsh, “A Comparative Study of Image-In-Image Steganography Using Three Methods of Least Significant Bit, Discrete Wavelet Transform and Singular Value Decomposition,” 2016.
5. V. Gupta and A. Barve, “International Journal of Mechanical Engineering and Information Technology,” *A review on Image Watermarking and its Techniques,* vol. 2, no. 1, p. 8, 2014.
6. “Advanced Encryption Standard,” [Online].Available: https:/[/www](http://www.tutorialspoint.com/cryptography/advanced_encryption_standard.htm).[tutorialspoint.com/cryptography/advanced\_encryption\_standard.htm.](http://www.tutorialspoint.com/cryptography/advanced_encryption_standard.htm)
7. A. Aggarwal, A. Sangal and A. Varshney, “IMAGE STEGANOGRAPHY USING LSB ALGORITHM,” International Journal of Information Sciences and Application (IJISA), 2019.
8. “Wikipedia,”[Online].Available: https://en.wikipedia.org/wiki/Python\_(programming\_language). [Accessed 9

February 2022].

1. S. Gyawali, “Encryptstego,” Kathmandu, 2021