# The Key to a Secure Communication: A Quantitative Report on Combining Cryptography with the Image Steganography Methods

##### **Sai Keerthan Kasula 1, Krishna Prasad Puppala2**

*1 Student, Department of MCA, Chaitanya Bharati Institute of Technology, India.*

*E-mail:*  [*saikeerthan.work@gmail.com*](mailto:%20saikeerthan.work@gmail.com)

*2 Assistant Professor, Department of MCA, Chaitanya Bharati Institute of Technology, India.*

*E-mail:*  [*pkrishnaprasad\_mca@cbit.ac.in*](mailto:%20saikeerthan.work@gmail.com)

**ABSTRACT**

Steganography is the art of hiding the information within the information and cryptography is the art of converting the information into an unreadable format using logical and arithmetic algorithms. In today’s world, many different carrier file formats can be used to communicate, but digital images are the most popular ones because of their frequency on the internet. By 2025, it’s estimated that 463 exabytes of data will be created each day globally – that’s the equivalent of 212,765,957 DVDs per day, for such a mass communication rate security will always be a concern. In this paper, I propose a new way of implementing steganography with the combination of cryptography. This method adds an extra layer of security between the communication of the sender and the receiver.

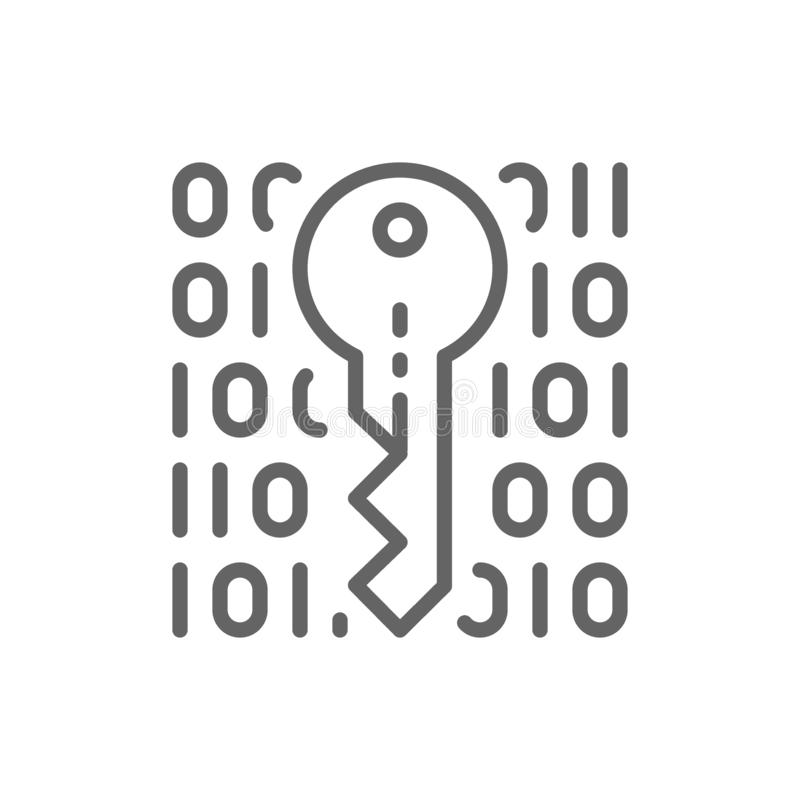
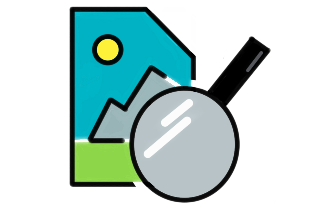
***Keywords:*** ***Image Steganography, LSB algorithm, AES Encryption, Cryptography, Text Embedding.***

1. **Introduction**

Steganography, which translates from Greek as covered or secret writing, has been used to conceal information for many years. While connected to cryptography, they are distinct from one another. While cryptography encrypts a message so that it cannot be decoded, steganography aims to conceal the presence of the message. The aim of steganography, to put it more specifically, "is to conceal communications inside other innocent messages in a way that does not allow any opponent to even identify that there is a second secret message there. There are many different methods for concealing messages in various media, including steganography. Invisible inks, microdots, digital signatures, covert channels, and spread-spectrum communications are a few examples of these techniques. Steganography is now employed on text, photos, music, signals, and more because to contemporary technology. Steganography has the benefit of enabling covert message delivery without the transmission's existence being discovered. Encryption frequently makes the sender or recipient appear to be someone who has something to conceal. For instance, the blueprints for your company's most recent technological advancement may be hidden in that image of your cat. Steganography, however, also has a number of drawbacks. In contrast to encryption, it typically takes a lot of overhead to cover up only a little amount of data. There are, however, workarounds for this.

**1.1 CRYPTOGRAPHY & STEGANOGRAPHY**

A steganographic system is worthless once it is identified. If the concealed data requires a key to be entered and extracted, this issue can also be solved. Before including the concealed message in the cover message, it is customary to encrypt it first. It should be emphasised, nevertheless, that steganography does not need the concealed message to be encrypted. Even though the message is written in straightforward English, it may still be buried. However, the majority of steganographers appreciate the additional level of security that encryption offers. If your hidden message is found, then at least make it as protected as possible and this is where cryptography comes in handy. Before embedding the information into the image, the proposed system makes sure that the text to be embedded is first encrypted with the help of the Advanced Encryption Standard (AES) algorithm and then it is embedded into the image. Through this practice, there will be an additional layer of security that protects the confidentiality of the secret information.

**** 

***Figure 1. Iconic Representation of the Proposed System***

**1.2 TERMINOLOGIES**

**Cover Image:** The image in which the secret data are embedded is acknowledged as a cover image.

**Stego-Image:** The image after embedding secret data is referred to as a stego image. cover medium + embedded message = stego message.

**Payload:** The secret data that is embedded in the cover image is known as payload.

**Payload Capacity:** The embedding rate per pixel.

**Stegokey:** To get the embedded message from the stego image, some piece of secret information is needed, this is acknowledged as stegokey. hvk

**Encrypt:** Concealing data in (something) by converting it into a code.

**Decrypt:** make (a coded or unclear message) intelligible.

**Steganalysis:** The name of attack in steganography.

**Cryptoana**ly**sis:** The name of attack in cryptography.

**2. LITERATURE SURVEY**

G. Prashanti and K. Sandhyarani have surveyed recent achievements of LSB-based image steganography. The changes that increase steganographic outcomes, such as high resilience, high embedding capacity, and un-detectability of concealed information, are covered by the authors in this survey. In addition to this survey, two novel approaches are also suggested. In the first method, the cover picture is used to hide data or secret messages, while in the second method, a hidden grayscale image is inserted inside another grayscale image. These methods generate pseudo-random numbers using a four-state table. Secret information is embedded using this method. These two techniques are more secure because secret information is concealed using pseudo-random numbers produced by the table in discrete, arbitrarily placed LSBs of the picture.

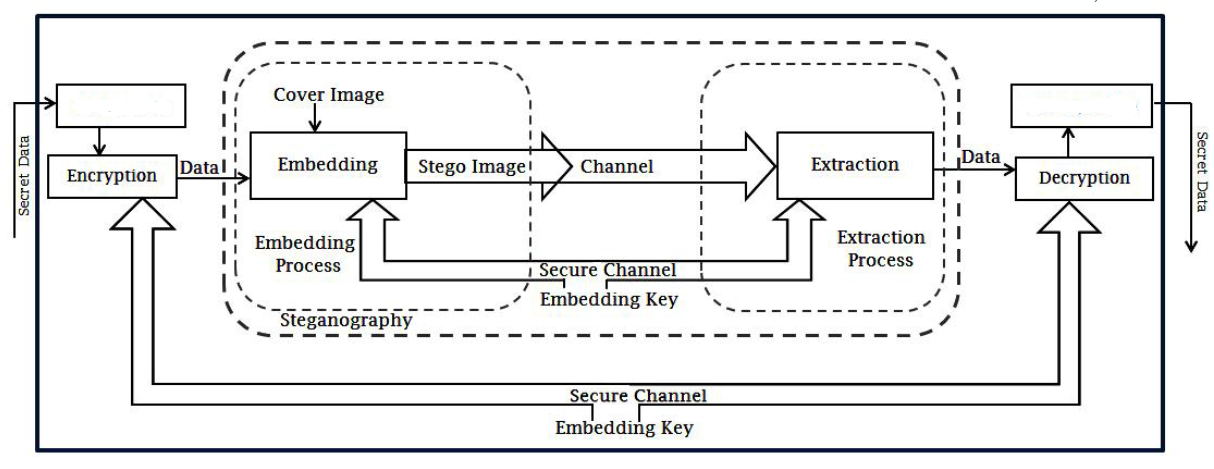
Savita Goel proposed a new method of embedding secret messages in the cover image using the LSB method using different progressions. Several image quality metrics, including Peak Signal to Noise Ratio (PSNR), Mean Square Error (MSE), histograms, CPU time, Structure Similarity (SSIM) index, and Feature Similarity Index Measure, are used by the authors to compare the stego picture's quality to that of the cover image (FSIM). In comparison to standard LSB approaches, their investigation and experimental findings demonstrate that the suggested method is quick and extremely effective.

Bingwen Feng, Wei Lu, and Wei Sun in their paper “Secure Binary Image Steganography Based on Minimizing the Distortion on the Texture” purposed a state-of-the-art approach to binary image steganography. This method is suggested to lessen the texture's distortion. The rotation, complement, and mirroring invariant texture patterns are first retrieved from the binary picture in this form of steganography. They also suggested measuring, and this strategy was really put into practise using the recommended measurement. The suggested steganographic technique provides good statistical security, high stego picture quality, and high embedding capacity, according to practical results.

On based on Huffman Coding, Amitava Nag present a novel steganographic technique of LSB substitution. Their method focuses on providing excellent security, more embedding space, and tolerable stego picture quality. To begin with, a Huffman tree is created to encrypt each 8-bit segment of the secret picture. They split the encoded bits into four pieces, each of which has a decimal value from 0 to 3. These decimal numbers dictate where a message will be included in the cover picture. According to experimental findings, the Huffman table makes it exceedingly difficult for an attacker to obtain the secret information.



***Figure.2 Flowchart of existing Image Steganography Techniques.***

******

***Figure 2. General System architecture of the proposed System***

Original Text

Plain Text

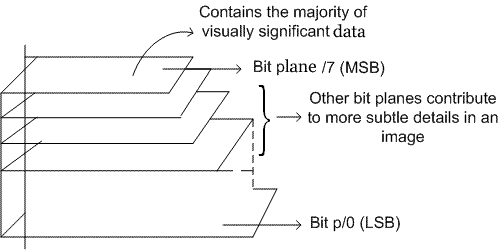
**3. IMAGE STEGANOGRAPHY METHODS**

**3.1 LSB**

A popular and straightforward method for including information in a cover picture is the least significant bit (LSB) insertion. Some or all of the bytes inside of an image have their least significant bit, or the eighth bit, altered to a bit of the secret message. Red, green, and blue colour components may all be utilised in a 24-bit picture as they are each represented by a single byte. In other words, each pixel may hold 3 bits. Thus, an image with a resolution of 800 x 600 pixels has a total data storage capacity of 1,440,000 bits, or 180,000 bytes.

**3.2 BIT PLANE**

It could be preferable to draw attention to particular bits' contributions to the look of the whole image rather than just grayscale images. Assume that an image's pixels are individually represented by 8 bits. Imagine the image is made up of 8 1-bit planes, with bit plane 1-0 (LSB) being the largest (MSB). In terms of the 8-bit bytes that make up an image's pixels, plane 0 includes all the lowest order bits and plane 7 all the highest order bits. A digital picture may be broken down into its bit planes to analyse the relative role that each bit plays in the image, which suggests that doing so might help establish the appropriateness of the amount of bits used to quantize each pixel and aid in image compression.



***Figure3 Bit Plane***

**3.3 SPIRAL EMBEDDING**

To make the embedding more difficult to decode, the spiral embedding arranges the image's pixels in a spiral pattern. The Spiral Embedding can be effectively decoded and helps fend against visual assaults thanks to two key concepts. The first is that known locations have embedded metadata about the image's contents. With the use of this knowledge, the stego object may be decoded and the hidden message retrieved. The data is serialised and placed in a pattern that makes it impossible to decipher the message in a visual assault, which is the second factor. Building a vector containing all the information that will be embedded into the cover, including the message contents and metadata, is the first step in the spiral embedding process. The message's dimensions are stored as unsigned 16-bit integers in the first 32 places of the vector. Pixel by pixel, a bit encoding the content of the vector is written into the LSB of the cover in a spiral pattern. The procedure of reading in the stored dimensions and then adhering to the same spiral pattern that guided the embedding is all that is required to decode a stego object produced using the spiral embedding. A vector is read with the LSB values of the stego object. The vector is divided to produce a new picture with the original dimensions of the message once all contained data has been read. This embedding's outcome may then be seen.

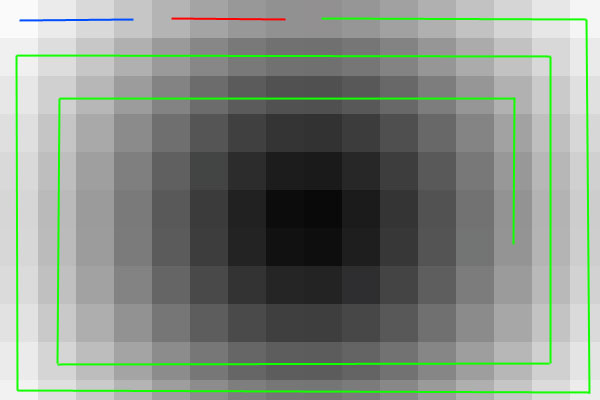
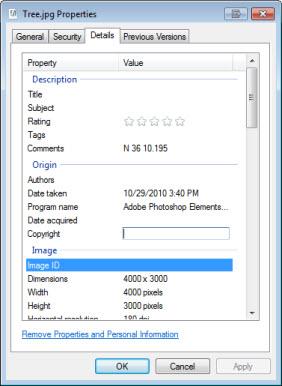


Figure.4 The Spiral Embedding pattern. The height information is embedded into the blue line pixels, the width information into the red pixels, and the message into the green pixels.

**3.4 METADATA MANIPULATION**

Data about data is known as metadata. If we consider an image file to be our data, metadata would comprise details like the image's name, caption, and dimensions (width and height). The same kinds of data are present in both audio and video files.

A digital camera's photos and sound files can have information about them stored in the Exchangeable Image File (Exif) format. Information may be hidden using a variety of data fields. A tiny portion of those fields will appear if we choose properties from the context menu when we right-click on a saved.jpg file.



***Figure 5. Meta Data Using File Properties*** Windows

**3.5 ALTERING THRESHOLD**

The simplest technique for segmenting images is thresholding. Thresholding may be used to produce binary pictures from a grayscale image. The most basic thresholding techniques swap out every pixel in a picture for a black one if the image intensity is below a set constant T or a white one if it is above that constant.

**3.6 APPENDED DATA**

The appending of data to the end of an image file is a straightforward and popular steganography technique. This method works because the majority of picture viewers reject any additional data, keeping the stego message concealed. You may add data to the end of an image file using the Linux "cat" command.

**3.7 PNG CHUNK ANALYSIS**

An 8-byte signature is the first byte of a PNG file. The pieces that follow the header each provide a different piece of information about the image. A programme may safely disregard an ancillary chunk that it does not understand when it encounters one because chunks identify themselves as either critical or ancillary. Steganography is based on this. Data can be included into unidentified ancillary chunks, but picture viewers and decoders will disregard it.

A chunk is made up of four components: length (4 big-endian bytes), chunk type/name (4 bytes), chunk contents (4 bytes), and CRC (4 bytes). The CRC is a network-byte-order CRC-32 that is computed over the type and content of the chunks, but not over their length.

A four-letter case-sensitive ASCII type or name is assigned to chunk kinds; see Four CC. The case of the various letters in the name (bit 5 of the character's numeric value) is a bit field that gives the decoder some knowledge about the make-up of chunks it does not recognise.

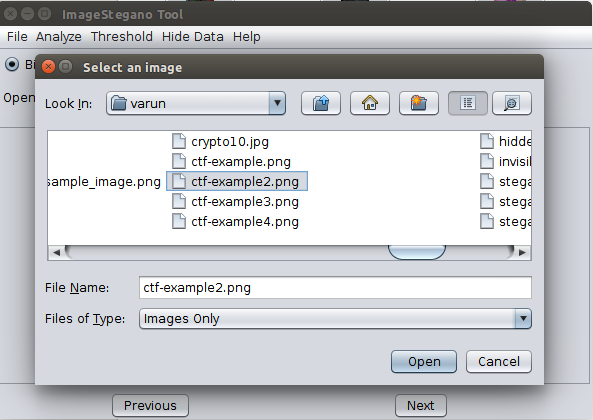
The initial letter's case determines whether or not the chunk is significant. The chunk is crucial if the initial letter is uppercase; if not, the chunk is ancillary. The file's essential information is included in critical portions. A decoder must stop reading a file or provide the user a suitable warning if it comes across a crucial chunk it doesn't recognise. Whether a chunk is "public" or "private" is indicated by the case of the second letter. Lowercase is private, whereas uppercase is visible. This makes sure that the names of public and private chunks can never clash. In order to comply with the PNG standard, the third letter must be capitalised. It is set aside for a potential expansion. A chunk with a lowercase third letter should be treated the same as any other unrecognised chunk by decoders. Editors who are unfamiliar with the chunk can determine if it is safe to copy by looking at the case of the fourth letter. If lowercase, the chunk may be safely copied regardless of the extent of modifications to the file. If uppercase, it may only be copied if the modifications have not touched any critical chunks [6].

**3.8 CHANGING THE COLOUR MAP**

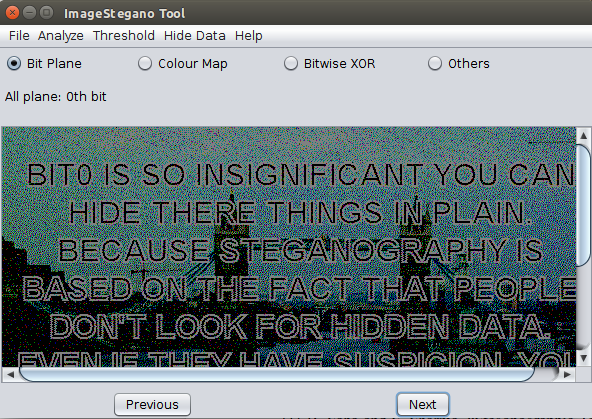
Indexed colour is a computing technology that manages the colours of digital photographs in a constrained way to conserve computer memory and file storage while accelerating display refresh and file transfers. It is a type of compression using vector quantization. When an image is encoded in this fashion, colour information is saved in a different piece of data called a palette, which is an array of colour elements, rather than being directly carried by the image pixel data. A colour is represented by each element in the array, which is indexed by its location in the array. Color registers are some names for the separate entries. Only its index in the palette, not the whole description of the colour, is stored in the image's pixels. Due to the indirect way in which colours are addressed, this approach is also referred to as pseudocolor or indirect colour. Using the right palette or colour map, this approach may be used to conceal data in a picture. We need to test several random colour mappings in order to decode data.

**4. RESULTS**

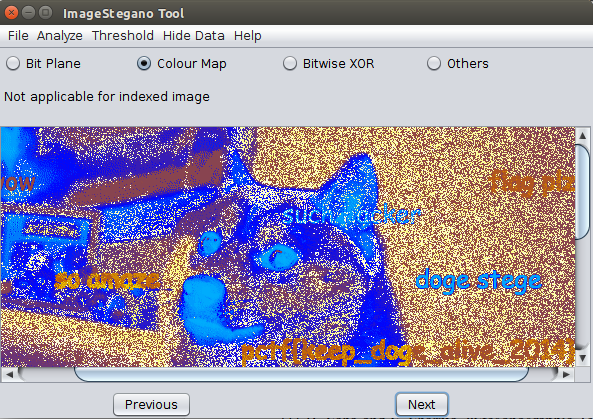
**4.1 SCREENSHOTS**



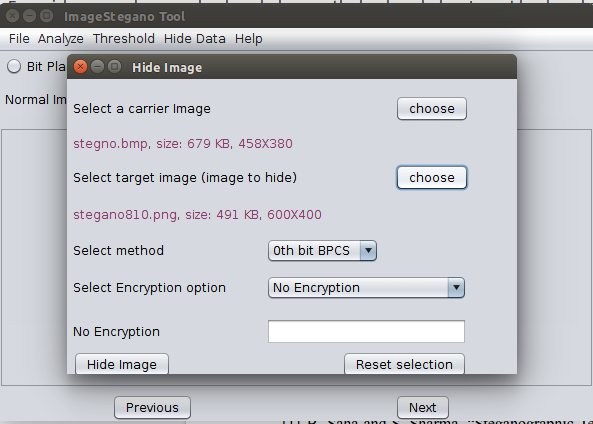
***Figure 6: Opening an image to analyze***



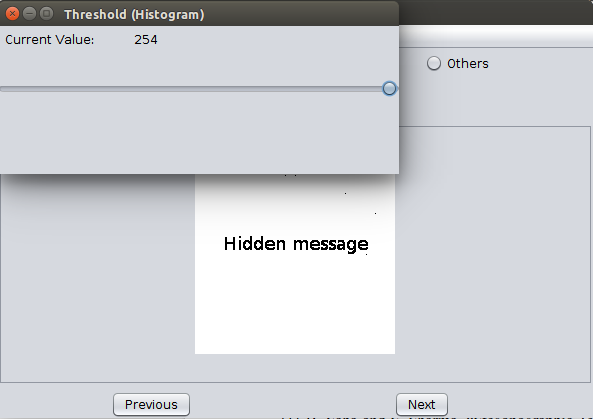
***Figure 7: Hidden Image found in the 0th-bit plane***



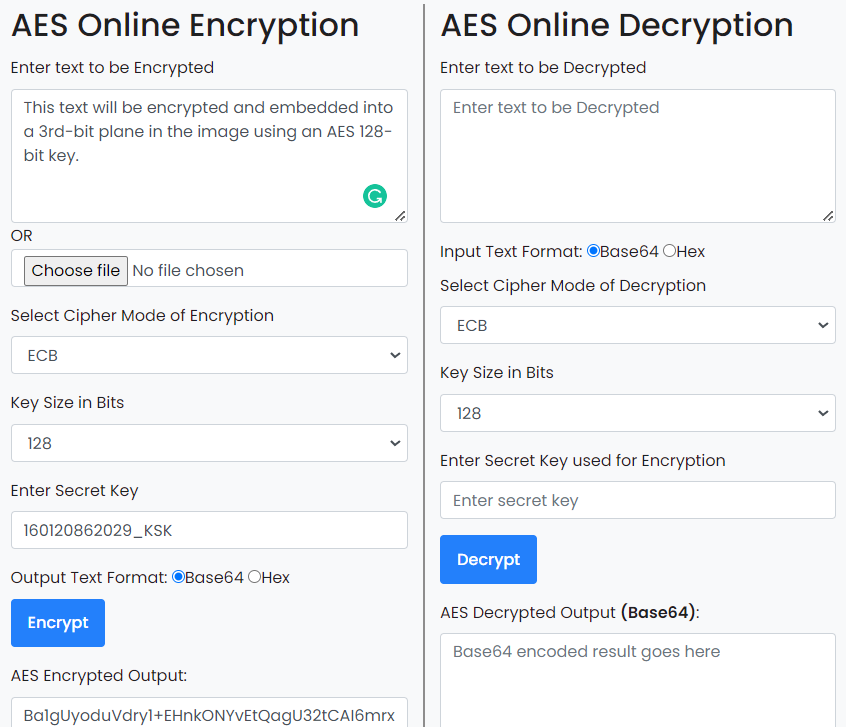
***Figure 8: Changing color map***



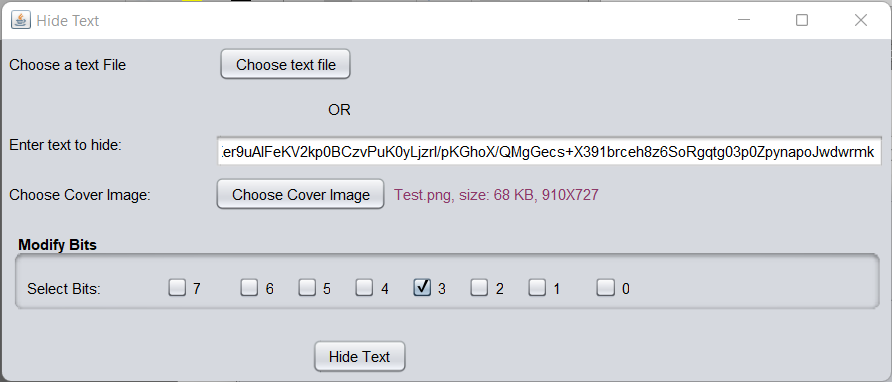
***Figure 9: Hide Image GUI***



***Figure 10: Hidden data visible on altering the threshold***



***Figure 11: Using the Dependency-AES tool***



***Figure 12: Hide Text using AES AND LSB***

**4.2 LIMITATIONS**

Crypstego tool has the following limitations-

• Every steganography technique directly modifies picture pixels in the spatial domain.

• JPEG image appended data extraction is ineffective.

• Compression options are absent from the tool.

• You cannot use a JPEG picture as a cover image to conceal data.

• Since there are difficulties with JAVA base64 decoding when carrying out the last block permutation of AES decryption, encryption within the programme is not possible. Dependency is therefore being employed.

**5. CONCLUSION**

Individuals and organisations can use the crypstego programme to provide an extra layer of security by ensuring that the payload is encrypted before being embedded into the picture. This platform-independent programme is useful for examining typical steganography techniques. Despite the fact that there are many other steganography programmes out there, the crypstego tool may be considered a fusion of numerous tools. Additionally, because it is open source, additional functions will be added over time by volunteer contributors. We have concentrated on non-functional criteria in addition to functional needs. The tool's restriction is that jpeg images cannot be used as carrier images to conceal data. Additionally, compression is not used by this tool.

**REFERENCES**

[1] B. Saha and S. Sharma, “Steganographic Techniques of Data Hiding using Digital Images”, in Defence Science Journal.

[2] Silman, J., “Steganography and Steganalysis: An Overview”, SANS Institute, 2001.

[3] Jamil, T., “Steganography: The art of hiding information in plain sight”, IEEE Potentials, 18:01, 1999.

[4] Johnson, N.F. & Jajodia, S., “Exploring Steganography: Seeing the Unseen”, Computer Journal, February 1998.

[5] Sei-Ichiro Kamata, et al: Depth-First Coding for Multi-Valued Pictures Using Bit-Plane Decomposition, IEEE Trans. on CT, Vol.43, No.5, pp.1961-1969, May 1995.

[6] <https://www.w3.org/TR/PNG-Chunks.html>.

[7] Herbert Schildt, “Java the Complete Reference Eleventh Edition – Mc Graw Hill India Pvt. Limited.

[8] <https://www.section.io/engineering-education/implementing-aes-encryption-and-decryption-in-java/>

[9] <https://www.coursera.org/learn/crypto>

[10] Jean-Philippe Aumasson, “Serious Cryptography: A Practical Introduction to Modern Encryption”

[11] R. Chandramouli, Nasir Memon, Analysis of LSB based Image Steganography Techniques, Proc. International Conference on Image Processing, 2001, Vol. 3, pp. 1019-1022, 2001.

[12] Jessica Fridrich, Miroslav Goljan, Dorin Hogea, Attacking the OutGuess, Proc. of 2002 ACM Workshop on Multimedia and Security, ACM Press, pp. 3-6, 2002.

[13] Jessica Fridrich, MiroslavGoljan, DorinHogea, Steganalysis of JPEG Images: Breaking the F5 Algorithm, Proc. of the 5th Information Hiding Workshop, Springer, vol. 2578, pp. 310-323, 2002.