Applied Information Technology And Computer Science Vol. 4 No. 1 (2023) 040-058
© Universiti Tun Hussein Onn Malaysia Publisher's Office

AITCS

Homepage: http://publisher.uthm.edu.my/periodicals/index.php/aitcs
e-ISSN :2773-5141

LH StegTool for Company Lian Heng Management

Yu Jia Jun[1], Nordiana Rahim[1]*

[1]Faculty of Computer Science and Information Technology,
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

*Corresponding Author Designation

DOI: https://doi.org/10.30880/aitcs.2023.04.01.003
Received 17 September 2022; Accepted 27 May 2023; Available online 30 June 2023

Abstract: LH StegTool is an image steganography tool that allows Lian Heng (LH) Management employees to conceal sensitive information within images. Steganography is a technique used to concealing information with various medium such as images, videos, audio, text and protocol, in order to prevent information from being disclosed to a third party. LH StegTool was created using steganography and cryptography; Cryptography was used to encrypt the hidden data by using Advanced Encryption Standard algorithm, encrypt the image by using Data Encryption Standard and steganography with the Least Significant Bit algorithm was used to embed the encrypted hidden data into the image. After the process of embedded message, an image with the hidden data is generated, which is known as a stego-image. The Object-Oriented Software Development Model is used to develop LH StegTool. This paper offered an overview of steganography, discussion of testing results for the stego-image and proved that LH StegTool can successfully handle the problem that the target company was experiencing, as well as secure the sensitive information of the company and their customers.

Keywords: Advanced Encryption Standard, Data Encryption Standard, Least Significant Bit, Object-Oriented Software Development Model, Steganography

1. Introduction

Nowadays, in the age of information technology advancements, the Internet has given enormous ease. Technology has gradually revolutionized our lives, and humans have gotten increasingly accustomed to preserving or sharing sensitive information on their mobile phones, computers, or social media accounts. However, it is not fully secure because all sensitive information might be destroyed, voyeur, manipulated, or intercepted, resulting in massive losses to the company, society, and even their financial future. As a result, data security should be regarded as a critical element during data transmission. Thus, Steganography is a technique notion of the security was coined by Johannes Trithemius in 1499 in his books on the subject "Steganographia" [1].

**Corresponding author: nordiana@uthm.edu.my*
2023 UTHM Publisher. All rights reserved.

The word of Steganography is the combination of two Greek words, Stegano, which means covered and Grafia, which means writing [1]. Steganography is different with the cryptography. Steganography is the technique for information hiding to prevent information from being disclosure to a third party and retrieved at its destination [2]. Cryptography refers to the techniques of protect the original message or word, often known as plaintext, by using encryption algorithm and converting it into cipher text and sending it to the receiver [2]. In short, the aim of cryptography is making the data unreadable, and the aim of steganography is to hide the data to prevent data cannot be seen by the third party. Steganography can be divided into five types which are text steganography, image steganography, video steganography, audio steganography and protocol steganography.

Every company in the world should face a slew of issues related to security, finance, technology, and exploding data, and Lian Heng Management is no exception. The security of the sensitive information or document is not secure. During my interview with Mrs. Chuah, I discovered that there are numerous account books and cheques on the employee's desktop and cabinet. Even their customer information, company information, or sensitive information is simply saved in their desktop without any safeguards. This can easily lead to information being lost, modified, or damaged by others.

The second problem that they encountered was the staff had sent the confidential and sensitive information by email to the incorrect company or customer due to their carelessness, resulting the sensitive information being disclosures to another person who would use the information to blackmail or selling the information to the others. Lastly, internal Lian Heng Management staff can easily gain access to other people's computers in order to obtain confidential company information. Usually, their PC does not have a password because the staff will call a colleague to assist them with some work when they are on holiday.

Therefore, the aim of this project is to develop an image steganography tool within an image to be employed by the two employees of Lian Heng (LH) Management who are in-charge of hiding the secret information. In order to achieve the aim above, few objectives have been set. They are:

1. To design an image steganography tool that can help Lian Heng Management by hiding the secret information within an image to prevent the secret information disclosure to the third party.
2. To develop an image steganography tool using Least Significant Bit techniques, Advanced Encryption Standard algorithm, and Data Encryption Standard algorithm.
3. To test the performance and functionality of the image steganography tool by using PSNR of the stego-image.

The domain of LH StegTool will be given priority in the category of business. The target users for this tool is two in-charge employees of Lian Heng Management. Besides that, employees should be login before using LH StegTool. To be used as a cover image to hide the message, LH StegTool only supports four types of image files: JPG, JPEG, BMP, and PNG. Each type of image file has a separate limit character; the message's capacity is determined by the image format and size. The following papers is organized as follows: Section 2 focused on related work of steganography. Section 3 will discuss more details about the methodology, followed by Section 4.

2 Overview of Steganography

Steganography is an effective method for protecting the secret message as well as the various media carriers in order to avoid message exposure to third parties as well as modifying, deleting, and copying. There must be a way to provide availability, integrity, confidentiality services to the information exchanged [3]. The basic model of steganography has three major components: cover medium, embedded message, and Stego-key. The cover medium, also known as the cover object, might be a file, picture, or video that will convey the secret message that will be concealed. The embedded message is the secret message that the sender must deliver to the receiver; it might be the original message or

something else. The following is Stego-key, which is specified as a secret key used to encode or decode the embedded message. Through the embedding process with the Stego-key, the embedded message will be hidden within a cover medium. After the embedding process, the stego file is sent to the receiver, who may view the message by extracting it.

2.1 Types of Steganography

Steganography can be divided into five categories of file formats which are text steganography, audio steganography, image steganography, protocol steganography and video steganography. Although there are many types of steganography, nowadays image steganography is widely used in any area because it is extensively utilized in our daily lives. The types of steganography are discussed in Table 1.

Table 1: Types of Steganography

Types of Steganography	Description
Text Steganography	Text steganography is a technique of concealing hidden information in a text file. The secret data is hidden behind every nth letter of every words of text message [4]. To achieve message hiding in text, a large amount of white spaces, tabs, and capital letters were utilized. Text steganography rarely used because text files containing large amount of redundant data.
Audio Steganography	Audio steganography can be defined as a technique of hiding secret information into digitized audio signal. It has become very significant medium due to voice over IP (VOIP) popularity [5]. MP3, WAV and au sound files can be used to conceal the information.
Image Steganography	Image steganography is a technique of concealing hidden information within an image by using a secret key. Least Significant Bit algorithm is commonly used to embed secret information into image cover. The pixel of the image will be substituted or replaced by the secret message after converted into binary form. The stego image is the outcome after embedding the hidden message inside an image.
Protocol Steganography	Protocol steganography is a technique of concealing hidden information within network protocols such as HTTP, FTP, TCP and so on [3], where protocol is used as a carrier. It is a more secure and advanced type of steganography when compared to other types of steganography, as well as the most complicated to implement.
Video Steganography	Video steganography can be defined as a technique of hiding secret information in a video format. Video formats such as MP4, AVI, and MPEG can be used to conceal the information. The techniques for image steganography and audio steganography can be applied into video steganography. The advantage of video steganography that it can be hidden vast quantity of information into video and it cannot be seen by human eyes [5].

2.2 Steganography Techniques

Spatial Domain method and Transform Domain method are the two domains of image steganography techniques. It is known as the adaptive steganography technique [16]. Spatial domain techniques will be used for this project, as will be discussed in the following section.

2.2.1 Spatial Domain Techniques

Spatial domain techniques, also known as image domain techniques, are used to directly embed information in the intensity of pixels [6]. When concealing data, there are many versions of spatial domain techniques that rely on transforming data into binary form and changing the bits in the image

pixel values. This technique is classed as Least Significant Bit (LSB), one of the most basic techniques used in image steganography, as will be discussed in the next section.

2.2.1.1 Least Significant Bit

Least Significant Bit (LSB) is one of the most fundamental picture steganography techniques. It works by embedding the message bits in the image's least significant bits after the message has been converted to ASCII in binary form. In the instance of image quality protection, this can ensure that the stego image does not vary from the original image [6]. When embedding the secret message, the majority of the image's bits were changed. The LSB technique described above restricts the size of the secret data to one-eighth the size of the cover image [12].

2.3 Cryptography

Cryptography is used to encrypt and decrypt messages using an encryption technique. Plaintext is the initial message in cryptography. As a consequence of encrypted plaintext, ciphertext is referred to as a result. The type of cryptography algorithm used and some forms of key are used to encrypt and decode information [13].

2.3.1 Advanced Encryption Standard (AES)

Advanced Encryption Standard (AES) is a symmetric key encryption standard frequently used because it has a high efficiency, it can encrypt a reasonably long plaintext. The National Institute of Standards and Technology (NIST) announced the AES block cypher encryption method in 2000 [14]. It includes three block ciphers which are 128-bit (16 bytes), 192-bit (25 bytes) and 256-bit (32 bytes) key lengths, and the iteration cycle number of rounds is determined by the key size, uses 10, 12 and 14 rounds respectively [15]. The encryption process is broken down into four parts in each round which are detailed below.

2.3.1.1 Substitute Bytes Transformation

In this part, it substitutes each byte of the state matrix with another byte using a nonlinear Sbox substitution. In AES, for example, if the state contains hexa 78, it must be replaced with hexa BC. The intersection of 5 and 3 resulted in the creation of ED.

2.3.1.2 ShiftRows Transformation

The bytes in the first row are not modified in this procedure. Only the initial bytes of the second row must be shifted circularly to the left. The first two bytes of the third row must be shifted to the left, and the last row of the last byte must be shifted circularly to the right.

2.3.1.3 MixColumns Transformation

MixColumns Transformation is using a matrix multiplication to multiply each row of matrix transformation and each column of the state. It used XOR function to produce the new four bytes after multiplication. For example, $S_{0,0} = (S_{0,0} * 02)$ XOR $(S_{0,1} * 03)$ XOR $(S_{0,2} * 01)$ XOR $(S_{0,3} * 01)$.

2.3.1.4 AddRoundKey Transformation

Bitwise XOR is used to add a roundkey to the state in this transformation and it also is the most vital part in AES algorithm. It creates a connection between the key and the ciphertext, which were obtained in the previous stage. The user specifies the key in order to increase the security of the ciphertext.

2.3.2 Data Encryption Standard (DES)

One of the symmetric-key methods for secret key cryptography is the Data Encryption Standard. It is most used to encrypt data in the 64-bit range with 56-bit key lengths to generate a 64-bit ciphertext [16].

The encryption key is the secret key used to keep the encrypted image safe. It is the inverse of the encryption process for the decryption procedure. It is unable to decrypt the encrypted image if the secret key is incorrect.

2.4 Mean Square Error (MSE)

Mean Square Error is calculated by averaging the original's squared intensity pixels from the (input) image and the (output) image [17]. The smaller the MSE value, the less the error. MSE using the equation following: $MSE = \sum_{M,N}[I_1(m,n) - I_2(m,n)]^2$, where the m and n are the number of rows and columns in the input images.

2.5 Peak signal-to-noise ratio (PSNR)

PSNR is used to evaluate the quality image between the original image and compressed image. The higher the PSNR ratio, the better the image quality. PSNR using the equation, $PSNR = 10\ Log10(R2/MSE)$, where the R is the maximum fluctuation in the input image data type [18]. For example, R is 255 for an 8-bit image.

2.6 Existing Work

According to Raju and Mohit Dhanda [7] proposed a unique algorithm to improve the techniques of Least Significant Bit that apply in grayscale and Red Green Blue (RGB) image sets. This proposed method that along the message hidden in LSB bits a part of message also resides at selective bites using a key. The key used in the operation is derived from the cover image itself. Because the key is embedded in the image, only the receiver will be aware of it and will be able to decode the message. After testing and comparing stego-image with the original image, there is no significant difference between the grayscale stego-image and the original image. For the RGB image, it has the lighter hue than the original image. Despite having certain issues owing to the RGB image, the secret message's security was higher and more secure.

Meanwhile Hemang et al [8] discovered that Dual Steganography is the more secure way of using image steganography. Dual Steganography is the combination of steganography process and cryptography algorithm. The original text, also knowns as plaintext, is first will be encrypted by using cryptography algorithm like Rivest Shamir Adelman (RSA) or Data Encryption Standard (DES), and then it is converted to ciphertext. Then, using the embedding algorithm, the ciphertext will be hidden inside the image. This technique allows for more secure communication.

JiaoHua et al [19] proposed a new method to apply in coverless steganography based on Generative Adversarial Network (GAN) to address the shortcomings of the work of Zhang [20], which had proposed a method for hiding arbitrary binary data in images using GAN. They proposed a new method that employs CNN, which was designed by Xu, used to provides a great detection performance in steganalysis, and GAN to achieve coverless steganography. After the experiments result and analysis, their models show that they have a high payload, and that the quality of the cover image, in particular, will not be modified during the process of hiding and extracting secret information. Compare with the traditional steganography algorithm and Zhang [20], this method is more convenient and secure.

Lastly, Khan Muhammad [10] proposed a method of the idea of transposition, bitxoring, bits shuffling, secret key, and cryptography to map secret data 6 to one of the three channels of the RGB image. After experiments, their proposed method had a high average PSNR of 58dB, RMSE with 0.6673, and NCC with 0.9917 after trials when compared to the traditional steganography algorithm. In their study report, they also included a comparison diagram of the PSNR with variable image dimensions, variable amount of embedded cipher and different images.

2.7 Compared of Existing System with LH StegTool

Table 2 compares the existing tools with LH StegTool from this view point: login for specific user, image supported, hidden message format, hidden message size, encryption key, and encryption types.

Table 2: The Comparison of existing tools and LH StegTool

	SteganographX Plus	Xiao Steganography	LH StegTool
Login for specific user	No	No	Yes
Supported image	BMP file	BMP and Wave files	Yes
Format of hidden message	Text only	All supported files	Text only
Size of hidden message	< 450 words	< 99 kb	JPG & JPEG (<=285 words) BMP (<= 240 words) PNG (<= 310 words)
Encryption key	Yes	Yes	Yes
Encryption key types	Make image into scrambled	RC2, RC4, DES, Triple DES, Triple DES112 And hash algorithm	AES and DES

3. Methodology/Framework

The Object-Oriented Software Development Model is used to develop the proposed tool in comparison to other methodology because it is suitable for developing LH StegTool. Java was chosen as the programming language to develop the proposed tool. Almost everything in Java is an object. All program code and data reside within objects and classes [11]. The Object-Oriented Software Development Model will make it much easier for the project to be completed on time and to meet all of the requirements.

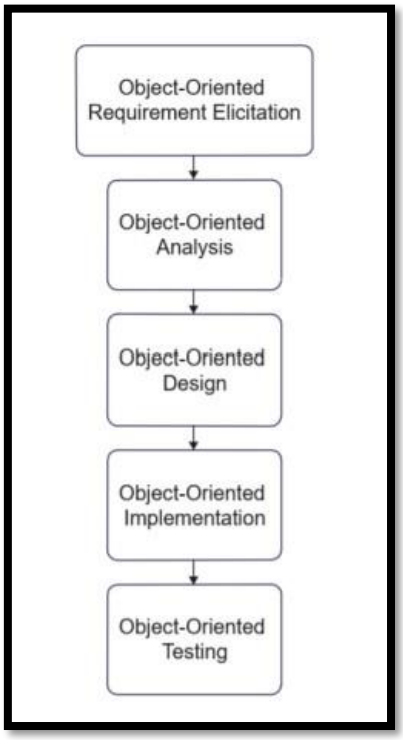

Figure 1: Object-Oriented Software Development Model

Figure 1 shows the Object-Oriented Software Development Model used to develop the tool. The activities are decomposed into processes in the Object-Oriented software development model. The next section will go over the five steps that will be used to develop the tool: object-oriented requirement

elicitation, object-oriented analysis, object-oriented design, object-oriented implementation, and object-oriented testing.

3.2 Object-oriented Requirement Elicitation

The requirement elicitation is the stage that focuses on describing the purpose of the new system. All the functional and non-functional requirements to solve the problem are gathered. Table 3 will be showing the list of functional requirement and Table 4 will be showing the non-functional requirement for LH StegTool.

Table 3: Functional Requirement for LH StegTool

Functional Requirements	Description
Login	• Only in charge staff can login by entering their username and password. • Error message will be displayed when in charge staff entered incorrect username and password. • LH StegTool will stop working for five seconds after entered incorrect username and password and it will be forced terminated if the incorrect username and password are input again.
Select Image	• User only can select JPG, JPEG, PNG and BMP image file format
Embed text within an image	• Only in charge staff can embed the text within an image after select the image and input the sensitive information (text). • Error message will be displayed when the incharge staff does not select any image or input empty text
Extract text from stego-image	• User can extract the text from the image after select the stego-image
Encryption and decryption text	• User can encrypt and decrypt the text by entering the key/password. • Error message will be displayed when user entered incorrect key/password
Encryption and decryption stego-image	• User can encrypt and decrypt the image by entering 8 characters key/password. • Error message will be displayed when user entered incorrect key/password

Table 4: Non-Functional Requirement for LH StegTool

Non-Functional Requirements	Description
Security	Embedding the secret message into an image after encryption and without saving it in any database to prevent those who would try to steal the message from the database from doing so. The image after embedded will not deviate different from the original image and will not easily be detected by human eye.
Usability	let the user feel at ease when using it, rather of being confused about how to use the tool or what they should do for the tool. For example, a user can anticipate that clicking an upload image button will access the user's desktop and allow the user to select the image as a cover image to conceal the message
Performance	The process of encoding and decoding the image would not take too long, and all of the buttons would also assign the user to the relevant session and obtain the correct result.

Based on the Table 5, the hardware used was a laptop and a USB Flash Drive. The laptop, HUAWEI Matebook D15, with the CPU AMD Ryzen 5 and 8GB RAM, is used to conduct some image steganography research, download software, and code; nevertheless, a USB Flash Drive is used to back

up all the data and files to protect the laptop from being destroyed and data loss. Based on the Table 6, LH StegTool will be developed using NetBeans as software. NetBeans is the software chosen to design the interfaces and code for the tool using Java programming.

Table 5: Hardware Requirement

Hardware	Specification
Laptop	Matebook D15, AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx 2.10 GHz with 8GB RAM
USB Flash Drive	SanDisk with 128GB

Table 6: Software Requirement

Software	Specification
Netbeans	A software that enables the development of tools or applications in Java programming and is also used to build interfaces.

3.2 Object-oriented Analysis

The analysis can assist the developer in verifying the previous stage's specification of LH StegTool. During this phase, the Class Diagram, Use Case Diagram and Sequence Diagram is defined at the Section 3.2.1, Section 3.2.2 and Section 3.3.3 respectively. 3.2.1 Class Diagram.

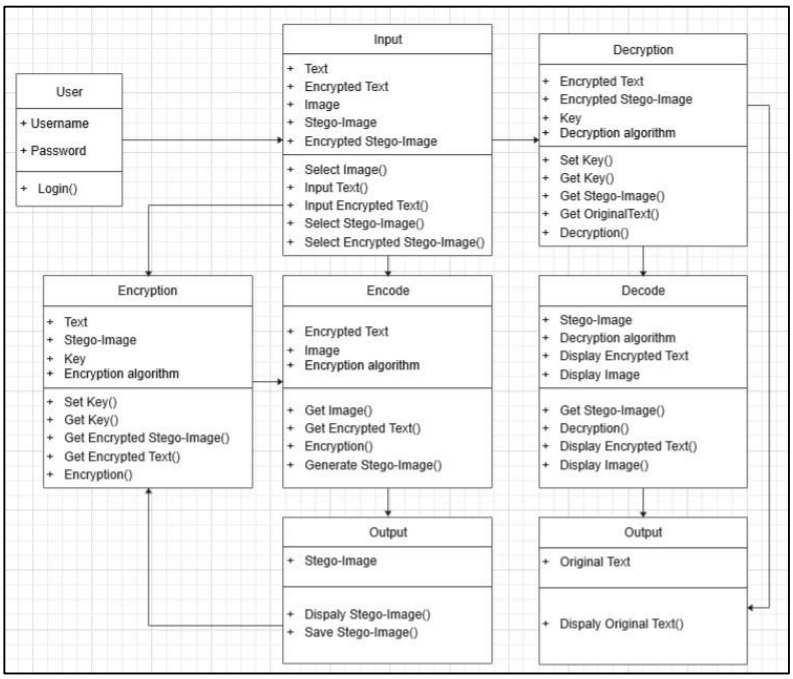

Figure 2: Class diagram of LH StegTool

Figure 2 shows the LH StegTool class diagram. This tool has seven classes: User, Input, Encode, Decode, Encryption, Decryption and Output. The User Class attributes are username and password, which the user must input before logging in to the tool. The Input Class has five attributes: text, encrypted text, image, stego-image and encrypted stego-image. Before beginning the encode process, the user can enter the text (sensitive information) and encrypt it, then select an image or stegoimage for encoding or decoding.

The Encryption class has three attributes: text, stego-image and key. It is used to encrypt the text by entering a key and may also encrypt the stego-image after encoding process is completed. Following that, Encode Class has two attributes: encrypted text and image. It will check the encrypted text and the

selected image, and if there are no errors, it will begin to embed the encrypted text within an image. Following that, Output class is display the stego-image and allow the user to save it.

For the Decode Class, the tool will decode the stego-image and extract the encrypted text, which will then be shown. Lastly, Decryption class three attributes: encrypted text, encrypted stego-image and key. User can decrypt the encrypted text with the correct key that had been extract from the stego-image to recover the original text, and the encrypted stego-image may also be decrypted before the decoding process.

3.2.1 Use Case Diagram

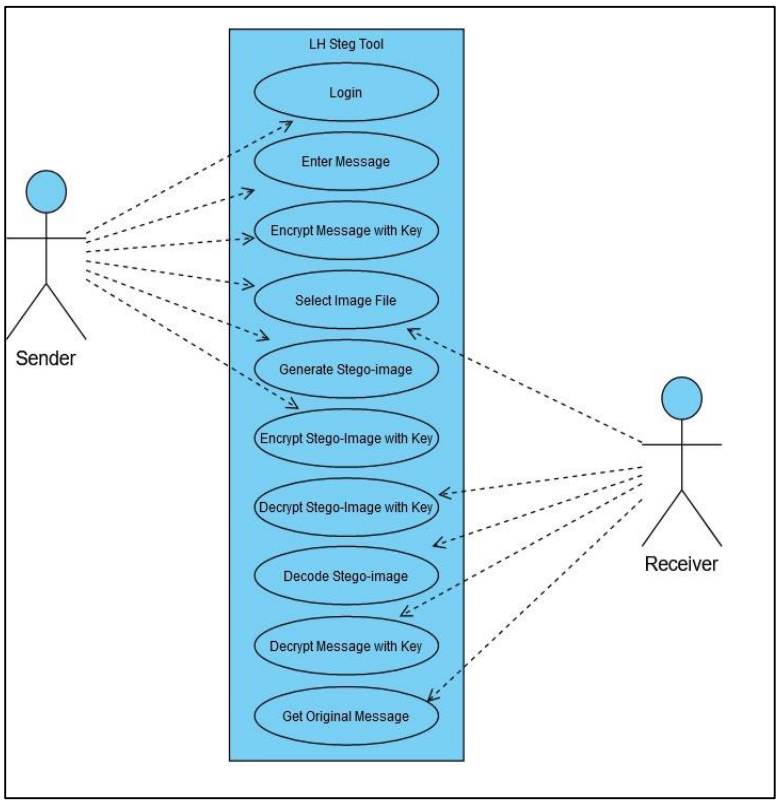

Figure 3: LH StegTool Use Case Diagram

The LH StegTool use case diagram is shown in Figure 3. This tool will be used by two individuals: the sender and the receiver. Only the sender must log in to the tool by entering their username and password. The sender can enter the message as the secret message to be hidden within an image, select the image as the cover media to conceal the secret message, and encrypt the message with the key. The sender may then encode the image, wait for the stego-image to be produced, and save it. After that, sender could encrypt the stego-image with the key as the final layer of security for the message. Receiver can also choose the image such as stego-image and encrypted stego-image, then begin decoding or decrypting it respectively. Following that, the receiver can extract the message and get the encrypted message. Lastly, decrypt the encrypted message with the correct key to recover the original message.

3.2.1 Sequence Diagram

The sequence diagram for user login is shown in Figure 4(a). This tool will be used by two individuals: the sender and the receiver. Only the sender must log in to the tool by entering their username and password, it will validate the user's information by using the staff-provided information. If it matches the database information, the user is allowed to login to LH StegTool; otherwise, the user receives a notice that says "Wrong username or password!".

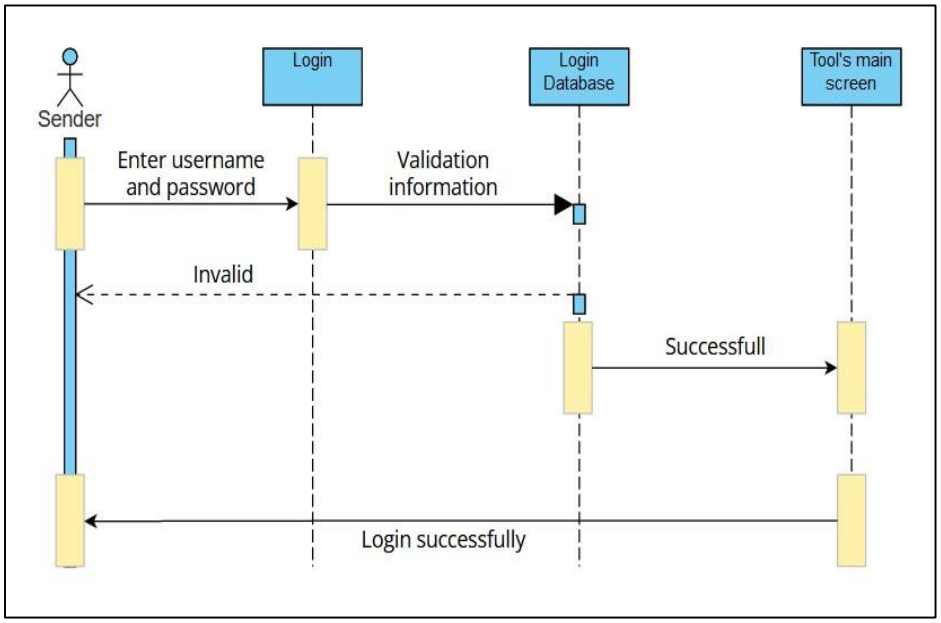

Figure 4(a): Sequence Diagram for User login

The sequence diagram for user encode image is shown in Figure 4(b). To hide the message, the sender must input the message and choose an image as a cover image. Sender could encrypt the message with the key, and then begin to embed the encrypted message within an image. The stego-image will be generated and displayed, and the sender will be able to save it to their computer. Following that, sender could encrypt the stego-image with the key and the encrypted stego-image directly save at the desktop.

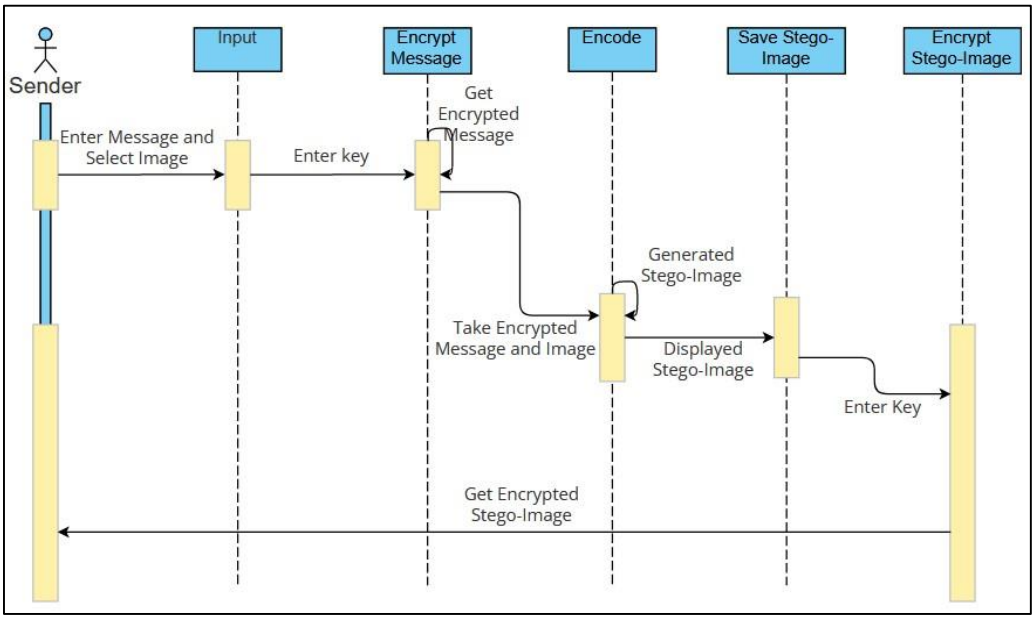

Figure 4(b): Sequence Diagram for User Encode Image

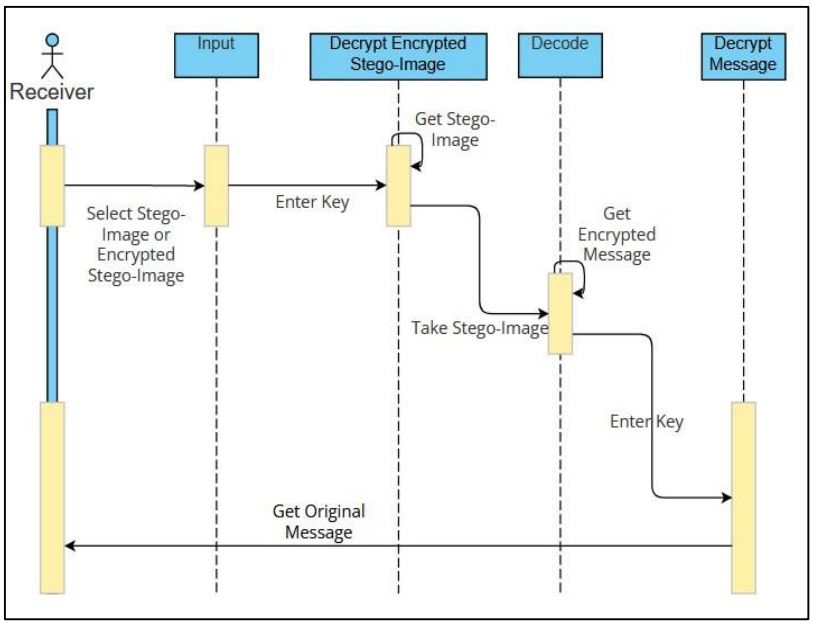

Figure 4(c): Sequence Diagram for User Decode Image

The sequence diagram for user decode image is shown in Figure 4(c). The user may choose the encrypted stego-image to decrypt by entering the correct key. Next, the selected stego- image will then be decoded, and the encrypted message will be displayed. The receiver needs to decrypt the encrypted message with the correct key to recover the original message.

3.3 Object-oriented Design

In the design phase, interface design activities will be implemented. To begin, design the interface for the main page, login page and also the function pages of LH StegTool. Next, the background of LH StegTool should be appropriate and be comfortable for the in-charge employees to use, the background color should not be too bright because it is difficult to see and hurts the eyes. Due to the high age range of the employees, it is critical that the font color and font size be clear and easy to read, as well as tool features such as buttons and user input fields need to be designed.

3.3 Object-oriented Implementation

In object-oriented implementation phase, the tool is being build based on the decision from the previous phase. The development of the tool will begin with the programming of the tool using a specific programming language, such as assigning a function to a button or designing the input and output procedures. All the results of the previous chapter will be implemented in this phase. The main task of developer is to translate the object model to the source code that will follow all of the requirements and make sure the requirements meet in the system. The interface design will also be used to develop the GUI.

3.3 Object-oriented Testing

Object-Oriented testing is a technique for verifying and validating software. During the testing phase, it will be tested that the text and file may be successfully hidden within the image, the quality of the stego-image, the process of decoding the stego-image, the validation of the proposed tool such as the maximum file size, the type of image, and the text limitation. After finishing testing, the tool will be evaluated by the target company's staff, Lian Heng Management, to obtain user feedback and fresh suggestions for improving the tool.

4. System Implementation and Testing

4.1 System Development

Implementation is a very important process in the whole project, this process will show how a system built, ensure the whole system runs in good condition and meets the quality standards. The following section will show and explain how LH StegTool's interface and modules are coded and implemented. LH StegTool has contained several modules which are login, encode image, decode image, encrypt text, decrypt text, encrypt stego-image and decrypt stego-image. Figure 5 shows that the home page of LH StegTool. There have two button and the user can choose between two options: Encode button and Decode button.

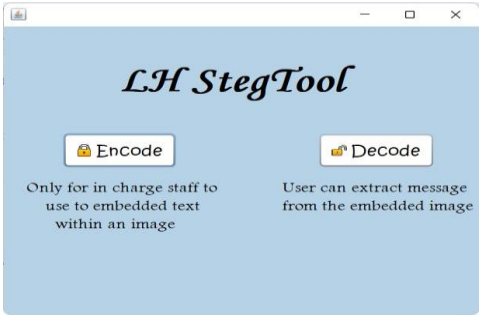

Figure 5: Home Page

After user clicks on the Encode button, the login page will be displayed as shown in Figure 6. The user needs to enter their username and password before they use the LH Steg Tool.

Figure 6 Login Module

Once the user has successfully logged in, the Encode Image Interface will be presented, as illustrated in Figure 7. The encrypted message can be typed in or copied and pasted into the textbox. Besides that, user could select an image file from their desktop by clicking the Open button. Besides that, user could select an image file (JPG, JPEG, BMP and PNG) from their desktop by clicking the Open button. It will generate the stego-image and display it alongside the original image if the message is successfully embedded into the image by clicking Encode button.

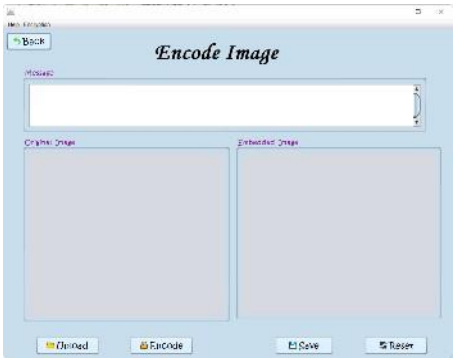

Figure 7: Encode Image Module

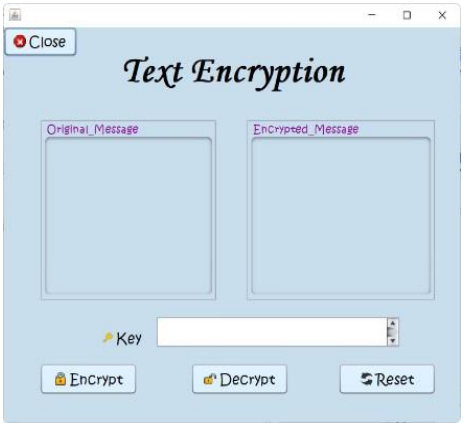

Figure 8: Encrypt Text Module

Figure 8 shows the Encrypt Text interface after selecting Encryption -> EncryptMessage from the Menu bar on the Encode Image page. User can encrypt the message by setting a key.

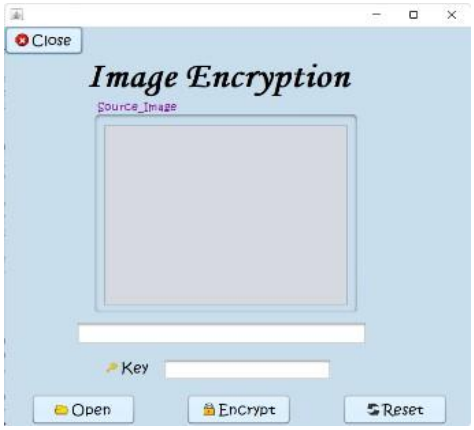

Figure 9: Encrypt Stego-Image Module

Figure 9 shows the encrypt stego-image interface after selecting Encryption -> EncryptMessage from the Menu bar on the Encode Image page. User can encrypt the stego-image by setting a key.

Figure 10 Decrypt Encrypted Stego-Image Module

Figure 10 shows the decrypt encrypted stego-image interface after selecting Decryption -> DecryptMessage from the Menu bar on the Decode Image page. User can Decrypt the encrypted stegoimage by setting a key.

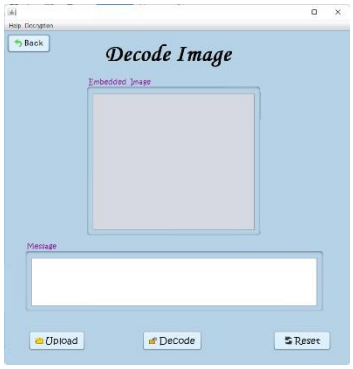

Figure 11: Decode Stego-Image Module

Figure 11 shows the decode stego-image interface. User should select an image file (PNG) from their desktop by clicking the Upload button. The encrypted message will be extract from the stego-image and then display at the textbox after clicking the Decode button.

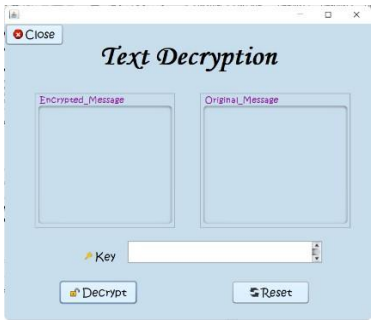

Figure 12: Decrypt Text Module

Figure 12 shows the Decrypt Text interface after selecting Decryption -> DecryptMessage from the Menu bar on the Decode Image page. User can decrypt the message by setting a key, and then original message will be displayed. If the key is incorrect, it is unable to decrypt the message.

4.2 Tool Testing

Tool testing is a process quality assurance to test the operation and function of LH StegTool can operation well and run as expected. Software is tested by giving it some related input and evaluating the output to determine how it complies, relates, or differs from its underlying requirements.

Table 7: Functional Testing Result for user login

Test Description	Expected Result	Actual Result
Login by entering user login and password	User can successfully login to the tools using valid username and password	As expected

Table 8: Functional Testing Result for Encrypt and Decrypt Text

Test Description	Expected Result	Actual Result
Input message and key before encrypt it	User able to enter the message and the key.	As expected.
Encrypt the message by clicking Encrypt button	The message can be encrypted and displayed successfully.	As expected.
Input encrypted message and key before decrypt it	User able to enter the encrypted message and the key.	As expected.
Decrypt the message by clicking Decrypt button	The encrypted message can be decrypted and displayed successfully.	As expected.

Table 9: Functional Testing Result for Encode and Decode Image

Test Description	Expected Result	Actual Result
Input message	The message will be entered successfully and displayed correctly.	As expected.
Select image by clicking Upload button	Allow the user to select JPG, JPEG, BMP and PNG image from their computer and have it displayed correctly.	As expected.
Input the number of characters of secret message exceed than the specified amount of characters	Prompt out error notification when the user input the number of characters of message exceed than the specified amount of characters.	As expected.
Encode image by - clicking Encode button	The message can successfully embed within an image.	As expected.
The stego-image will be created after the image encoding process.	The stego-image will be displayed, with the ability to save it to the user's PC.	As expected.
Decode image by - clicking Decode button	The message can successfully extract from the stego-image.	As expected.

Table 10: Functional Testing Result for Encrypt and Decrypt Stego-Image

Test Description	Expected Result	Actual Result
Select stego-image and key before encrypt it	User able to select the image and input the key.	As expected.
Encrypt the stego-image by clicking Encrypt button	The secret message can be encrypted and save it to the user's PC.	As expected.
Select encrypted stego-image and input key before decrypt it	User able to select encrypted stego-image and input the key.	As expected.
Decrypt the encrypted stego-image by clicking Decrypt button	The encrypted stegoimage can be decrypted and save it to the user's PC.	As expected.

Table 11: Test Plan for Input Validation

Test Description	Actual Result
Prompt out error message when user enter invalid username and password	As expected.
Prompt out error message when user enter a number of messages exceeds the given character limit	As expected.
Prompt out error message when user try to encrypt the message without input any key or message.	As expected.
Prompt out error message when user try to decrypt the encrypted message without input any key or message.	As expected.
Prompt out error message when user try to decrypt the encrypted message by input incorrect key.	As expected.
Prompt out error message when user try to encode the image without input any messages.	As expected.
Prompt out error message when user try to encode the image without select any image.	As expected.
Prompt out error message when user try to decode the image without select any image.	As expected.
Prompt out error message when user try to encrypt the stego-image without input any key.	As expected.
Prompt out error message when user try to encrypt the stego-image without select any image.	As expected.

Table 11: (cont)

Test Description	Actual Result
Prompt out error message when user try to decrypt the encrypted stego-image without input any key.	As expected.
Prompt out error message when user try to decrypt the encrypted stego-image without select any image.	As expected.

Table 12: Testing User Plan for Sender

Test Description	Actual Result
Tool can be executed from start to end.	As expected.
Sender is able to login to the tool.	As expected.
Sender is able to encrypt the secret message.	As expected.
Sender is able to decrypt the secret message.	As expected.
Sender is able to input secret message.	As expected.
Sender is able to choose image.	As expected.
The selected image can be previewed by the sender.	As expected.
Sender is able to encode the image.	As expected.
Sender is able to see the stego-image.	As expected.
Sender is able to save the stego-image.	As expected.
Sender is able to encrypt the stego-image.	As expected.

Table 13: Testing User Plan for Receiver

Test Description	Actual Result
Tool can be executed from start to end.	As expected.
Receiver is able to decrypt the secret message.	As expected.
Receiver is able to choose image.	As expected.
The selected image can be previewed by the receiver.	As expected.
Receiver is able to decode the stego-image.	As expected.
Receiver is able to see the encrypted message after decode.	As expected.
Receiver is able to decrypt the encrypted stego-image.	As expected.
Sender is able to encrypt the stego-image.	As expected.

4.3 Experiments and Result

In this section will go through the quality of the stego-image. The differences between the two images and the stego-image used to embed the company's secret messages are displayed in Figure 13. As can be seen, the original image and the stego-image are nearly identical.

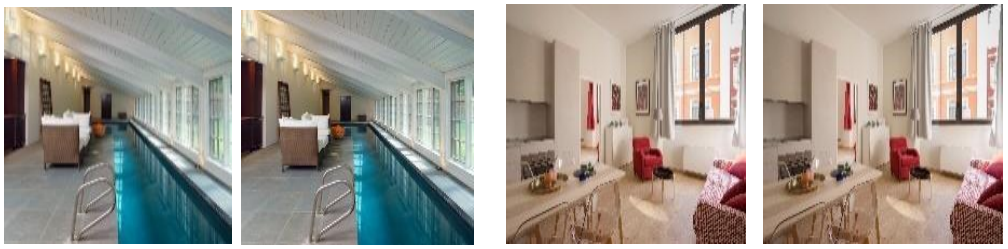

Figure 13: Two original image (left) and stego-image (right) and differences between them.

Table 14: The Quality of various type of Stego-Images

	Stego-Image	PSNR (dB)
JPG	JPG image with embedded 100 words	87.74
	JPG image with embedded 100 words (encrypted)	86.43
	JPG image with embedded 150 words	85.65
	JPG image with embedded 150 words (encrypted)	84.25
	JPG image with embedded 200 words	83.69
	JPG image with embedded 200 words (encrypted)	82.51
BMP	BMP image with embedded 100 words	96.51
	BMP image with embedded 100 words (encrypted)	95.18
	BMP image with embedded 150 words	94.90
	BMP image with embedded 150 words (encrypted)	93.54
	BMP image with embedded 200 words	92.01
	BMP image with embedded 200 words (encrypted)	91.55
PNG	PNG image with embedded 100 words	100.53
	PNG image with embedded 100 words (encrypted)	99.23
	PNG image with embedded 150 words	98.88
	PNG image with embedded 150 words (encrypted)	97.37
	PNG image with embedded 200 words	96.12
	PNG image with embedded 200 words (encrypted)	95.44

According to Table 14, when compared to JPG and BMP stego-image files, the PSNR of the PNG stego-image file is the greatest. The PSNR for every PNG stego-image file containing a variety of embedded messages is 95 dB or higher. The PSNR for JPG stego-image files is the lowest, at 82.51 dB. The PSNR for stego-image files in JPG, BMP, and PNG with embedded 100 words is 87.74 dB, 96.51 dB, and 100.53 dB, respectively. The PSNR for stego-image files in JPG, BMP, and PNG with embedded 200 words is 82.51dB, 91.55 dB, and 95.44 dB, respectively.

5. Conclusion

In conclusion, at the end of the project, LH StegTool has successfully implemented the features and requirements, satisfying the purpose of the tool. The main objective for this project is to design an image steganography tool which can help Lian Heng Management by hiding the secret information within an image to prevent the secret information disclosure to the third party. The requirements of Lian Heng Management are also met by the high quality of the stego-image and the capacity of the embedded data. Although the tool meets the user's requirements and functional requirements, it still has certain flaws, and there are ways to improve and upgrade it in the future. Finally, future stages for LH StegTool will be developed to increase efficiency and utility.

Acknowledgment

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support and encouragement throughout the process of conducting this study.

References

[1] M. S. Taha, M. S. Mohd Rahim, S. A. Lafta, M. M. Hashim, and H. M. Alzuabidi, “Combination of Steganography and Cryptography: A short Survey,” IOP Conf. Ser. Mater. Sci. Eng., vol. 518, p. 052003, 2019.

[2] Munjal, D. (2016). A Review Paper of Dual Steganography Technique Using Status LSB and DWT Algorithms. *International Journal of Computer Science & Engineering Technology*.

[3] Manjula, G. R., & Danti, A. (2015). A novel hash based least significant bit (2-3-3) image steganography in spatial domain. *arXiv preprint arXiv:1503.03674*.

[4] Surana, J., Sonsale, A., Joshi, B., Sharma, D., & Choudhary, N. (2017). Steganography Techniques. *IJEDR*, *5*, 989-992.

[5] Singh, S. K., Yadav, S., Raj, A., & Gupta, P. (2018, October). A Survey paper on different Steganography Techniques. In *Proceedings on international conference on Emerg* (Vol. 2, pp. 103-108).

[6] Srilakshmi, P., Himabindu, C., Chaitanya, N., Muralidhar, S. V., Sumanth, M. V., & Vinay, K. (2018). Text embedding using image steganography in spatial domain. *International Journal of Engineering & Technology*, *7*(3.6), 1-4.

[7] Raju, M. D. (2015). An Improved LSB based Image Steganography for Grayscale and Color Images. *International Journal of Current Engineering and Technology*, *5*(5), 3295-3297.

[8] Prajapati, H. A., & Chitaliya, N. G. (2015). Secured and robust dual image steganography: A survey. *International Journal of Innovative Research in Computer and Communication Engineering*, *3*(1), 30-37.

[9] Qin, J., Wang, J., Tan, Y., Huang, H., Xiang, X., & He, Z. (2020). Coverless image steganography based on generative adversarial network. *Mathematics*, *8*(9), 1394.

[10] Muhammad, K., Ahmad, J., Sajjad, M., & Zubair, M. (2015). Secure image steganography using cryptography and image transposition. *arXiv preprint arXiv:1510.04413*.

[11] Henriques, L., & Bernardino, J. (2018). Performance of memory deallocation in C++, C# and Java.

[12] E. Emad, A. Safey, A. Refaat, Z. Osama, E. Sayed and E. Mohamed, "A secure image steganography algorithm based on least significant bit and integer wavelet transform," in *Journal of Systems Engineering and Electronics*, vol. 29, no. 3, pp. 639-649, June 2018, doi: 10.21629/JSEE.2018.03.21

[13] Kessler, G. C. (2003). An overview of cryptography.

[14] Abdullah, A. M. (2017). Advanced encryption standard (AES) algorithm to encrypt and decrypt data. *Cryptography and Network Security*, *16*, 1-11.

[15] Sheth, U., & Saxena, S. (2016, April). Image steganography using AES encryption and least significant nibble. In *2016 International Conference on Communication and Signal Processing (ICCSP)* (pp. 0876-0879). IEEE.

[16] Rihan, S. D., Khalid, A., & Osman, S. E. F. (2015). A performance comparison of encryption algorithms AES and DES. *International Journal of Engineering Research & Technology (IJERT)*, *4*(12), 151-154.

[17] Al-Najjar, Y. A., & Soong, D. C. (2012). Comparison of image quality assessment: PSNR, HVS, SSIM, UIQI. *Int. J. Sci. Eng. Res*, *3*(8), 1-5.

[18] Kesarwani, S., Pal, N., Negi, M., Singh, D., & Aggarwal, A. (n.d.). LSB Steganography with PSNR and Data Integrity Check. Retrieved July 21, 2022, from Irjet.net website: https://www.irjet.net/archives/V6/i1/IRJET-V6I1230.pdf

[19] Hamid, N., Yahya, A., Ahmad, R. B., & Al-Qershi, O. M. (2012). Image steganography techniques: an overview. *International Journal of Computer Science and Security (IJCSS)*, *6*(3), 168-187.

[20] Zhang, K. A., Cuesta-Infante, A., Xu, L., & Veeramachaneni, K. (2019). SteganoGAN: High capacity image steganography with GANs. *arXiv preprint arXiv:1901.03892*.