https://afrjcict.net

Design and Implementation of a Steganographic System using the Least Significant Bit Algorithm

Christopher Ubaka Ebelogu[1], Idongesit Essien[2], M. B. Hammawa[3] and Emmanuel Chinanu Uwazie[4]
[1,2,3]Department of Computer Science,
University of Abuja,
Nigeria.
[4]Department of Computer Science,
Nasarawa State University, Keffi,
Nigeria.
Email: [1]Christopher@uniabuja.edu.ng, [2]idongessien@yahoo.co.nz, [3]mbhammawa@gmail.com, [4]uwazieemmanuel@yahoo.com

ABSTRACT

Secure data transmission over communication channels has been a critical issue since the beginning of the digital era. Due to advances in ICT, most of information is kept electronically. Consequently, the security of information has become a fundamental issue. Besides cryptography, steganography can be employed to secure information. Steganography is similar but adds another dimension to Cryptography. In cryptography, an intruder is normally aware that data is being communicated, because they can see the coded/scrambled message while in steganography, an unintended recipient or an intruder is unaware of the fact that observed data contains hidden information. In this method, people not only want to protect the secrecy of an information by concealing it, but they also want to make sure any unauthorized person gets no evidence that the information even exists. An example is invisible watermarking; although, this technique is not new, its application has been exploited in various information security technologies. This paper shows the design and implementation of a software system that uses the new image steganographic approach. This approach makes use of Least Significant Bit (LSB) algorithm and is implemented through the Microsoft.NET framework. This software is applied in embedding data into a bit map image (.bmp).

Keywords: *Steganography, Cryptography, Security, Stego-object, Stego-key, Encryption, Decryption, LSB (Least Significant Bit).*

African Journal of Computing & ICT Reference Format:

Christopher Ubaka Ebelogu, Idongesit Essien, M. B. Hammawa and Emmanuel Chinanu Uwazie (2019), Design and Implementation of a Steganographic System using the Least Significant Bit Algorithm,
Afr. J. Comp. & ICT, Vol.12, No. 4, pp. 86 - 100.

I. INTRODUCTION

Due to advances in ICT, most of information is kept electronically. Consequently, the security of information has become a fundamental issue. Information security is becoming more important as the number of data being exchanged on the internet increases. Therefore, the confidentiality and data integrity are required to protect against unauthorized access and users. This has resulted in an explosive growth of the field of information hiding.

Steganography in Greek means "covered writing" [5]. Steganography is the process of hiding one information into other sources of information like text, image or audio file, so that it is not visible to the natural view.

There has been a rapid growth of interest in steganography for two reasons: The publishing and broadcasting industries have become interested in techniques for hiding encrypted copyright marks and serial numbers in digital films, audio recordings, books and multimedia products. Moves by various governments to restrict the availability of encryption services have motivated people to study methods by which private messages can be embedded in seemingly innocuous cover messages. The basic model of steganography consists of Carrier, Message and password. Carrier is also known as cover-object, which the message is embedded and serves to hide the presence of the message.

Steganography can be employed to secure information. In steganography, the message or encrypted message is embedded in a digital host before passing it through the network, thus the existence of the message is unknown. Besides hiding data for confidentiality, this approach of information hiding can be extended to copyright protection for digital media: audio, video and images.

Steganography is also the practice of hiding private or sensitive information within something that appears to be nothing out to the usual. If a person or persons view the object that the information is hidden inside of he or she will have no idea that there is any hidden information, therefore the person will not attempt to decrypt the information.

What steganography essentially does is exploit human perception; human senses are not trained to look for files that have information inside of them. Software is available that can hide a file inside another file.

Steganography differs from cryptography and watermarking. Image steganography is the method of hiding highly sensitive information in a cover image and the resulting stego image is transferred securely through an unsecure channel. In steganography, sensitive information is invisible to the unauthorized person, while in cryptography the unauthorized person cannot able to understand the sensitive information though it is visible to all [6]. For Cryptography, in some cases sending encrypted data may draw attention while invisible data may not. Watermarking embed the information related to the digital data, while in steganography there is no relation between information and digital data. Among these methods, steganography provides a high level of security to the secret data.

There are different approaches in classifying the steganographic method. In the first approach, it is classified according to the type of cover used for communication and in the second approach; it is classified depending on the cover modification performed in embedding process. The second approach was followed in this work [23].

1.1 AIM/OBJECTIVES OF RESEARCH

The main goal of this project is to develop a system to communicate securely in a completely undetectable manner and to avoid drawing suspicion to the transmission of a hider data.

This will be achieved by:

-proposing a new framework of an image steganography system to hide a digital text of a secret message and information.

-Designing and implementing a software application to make the information hiding simpler and user friendly.

1.2 MOTIVATION FOR THE STUDY

Cryptography hides the contents of a secret message from malicious people, whereas steganography even conceal the existence of the message. In cryptography, the system is broken when the attacker can read the secret message. Breaking a steganography system need the attacker to

detect that steganography has been used. It is possible to combine the techniques by encrypting message using cryptography and then hiding the encrypted message using steganography. The resulting stego-image can be transmitted without revealing that secret information is being exchanged.

It is on this premise that this project seeks to implement an efficient steganographic system to make it difficult for illegitimate users to be aware of the existence a secret message.

II. REVIEW OF LITERATURE

2.1 STEGANOGRAPHY

There are varieties of steganographic techniques available to hide the data depending upon the carriers we use. Steganography and cryptography both are used for the purpose of sending the data securely. The same approach is followed in Steganography as in cryptography like encryption, decryption and secret key. In steganography, the message is kept secret without any changes but in cryptography the original content of the message is differed in different stages like encryption and decryption. Steganography supports different types of digital formats that are used for hiding the data. These files are known as carriers. The main file formats that are used for steganography are text, images, audio, video, protocol [1].

The different types of steganographic techniques available are:
1. Pure steganography
2. Public key steganography
3. Secret key steganography

Pure Steganography:
Pure steganography is the process of embedding the data into the object without using any private keys.
This type of steganography can't provide better security because it is easy to extract the message if the unauthorized person knows the embedding method. It has one advantage that it reduces the difficulty in key sharing [4].

Public Key Steganography:
Public key steganography uses two types of keys: one for encryption and another for decryption. The key used for encryption is a private key and for decryption, it is a „public key" and is stored in a public database [4].

Secret Key Steganography:
Secret key steganography is another process of steganography which uses the same procedure other than using secure keys. It uses the individual key for embedding the data into the object which is similar to symmetric key. For decryption it uses the same key which is used for encryption.
The main problem of using this type of steganographic system is sharing the secret key. If the attacker knows the key it will be easier to decrypt and access original information [4].

Model of Steganography
Basically, the model for steganography is shown in Figure 1

The embedded message is the data that the sender wishes it remains confidential. It can be plain text, ciphertext, other image, or anything that can be embedded in a bit stream such as a copyright mark, a covert communication, or a serial number. Password is known as *stego-key*, which ensures that only a recipient who knows the corresponding decoding key will be able to extract the message from a *cover-object*. The *cover-object* with the secretly embedded message is then called the *Stego-object*.

Recovering the message from a *stego-object* requires the decoding key; that is the *stego-key* used during the encoding process. There are several suitable carriers below that can be the *cover-object:*

- Network protocols such as TCP, IP and UDP
- Audio in digital audio formats such as wav, midi, avi, mpeg, mpi and voc
- File and Disk that can hide and append files by using the slack space
- Text such as null characters, just like most code including html and java
- Image file such as bmp, gif and jpg, where they can be both color and gray-scale.

The process of hiding consists of two steps:

- Identification of redundant bits in a *cover-object* that can be modified without corrupting the quality or destroying the integrity of the *cover-object.*
- Embedding process then selects the subset of the redundant bits to be replaced with data from a secret message. The *stego-object* is

created by replacing the selected redundant bits with message bits

2.1.1 STEGANOGRAPHIC ALGORITHMS

For encryption and decryption of text messages using the secret key, steganographic systems use steganographic algorithms. The mostly used algorithms for embedding data into images are:

1. LSB (Least Significant Bit) Algorithm
2. JSteg Algorithm
3. F5 Algorithm

LSB Algorithm

LSB (Least Significant Bit) substitution is the process of adjusting the least significant bit pixels of the carrier image. It is an approach in which the least significant bits of some or all of the bytes in an image are replaced with the secret message bits [7]. The Least Significant Bit insertion varies according to number of bits in an image. For an 8 bit image, the least significant bit i.e., the 8^{th} bit of each byte of the image is changed to the bit of secret message. For 24 bit image, the colours of each component like RGB (red, green and blue) are changed. LSB is effective in using BMP (Bitmap) images since the compression in BMP is lossless. But for hiding the secret message inside an image of BMP file using LSB algorithm it requires a large image which is used as a cover.

LSB substitution is also possible for GIF formats, but the problem with the GIF image is whenever the least significant bit is changed the whole colour palette will be changed. The problem can be avoided by only using the gray scale GIF images since the gray scale image contains 256 shades and the changes will be done gradually so that it will be very hard to detect. For JPEG, the direct substitution of steganographic techniques is not possible since it will use lossy compression. There are many approaches available for hiding the data within an image: one of the simple least significant bit submission approaches is "Optimum Pixel Adjustment Procedure". This method of substitution is simple and easy to retrieve the data and the image quality better so that it provides good security.

[10] and [11] have proposed the LSB replacement or LSB substitution method. The concept of LSB replacement is, the LSB plane of the cover image is replaced or overwritten or embedded directly with the secret bit according to the pseudo random number generator. In this method, the embedded secret data can easily be detected by the unauthorized person even at a low embedding capacity. The embedding capacity of this method is 1 bit per pixel. A new method of hiding the data into binary images based on approaches of changing low-level and high-level features of cover image was proposed by [16]. This method introduces a visual artifact.

In [12], method data is embedded into the red plane of the image and the pixel is selected using a random number generator. It is almost impossible to notice the changes in the image. A stego key is used to seed the PRNG (Pseudo Random Number Generator) to select pixel locations. This paper focuses on increasing the security of the message and reducing distortion rate.

In [13] article the authors propose a novel method based on LSB. Data embedding is performed using a pair of pixels as a unit, where LSB of the first pixel carries one bit of information and a function to two pixel values carries another bit of information. The proposed method shows better performance in terms of distortion and resistance against existing steganalysis. Embedding is done in the sharper edge regions using a threshold. PSNR value is compared for adaptive and non-adaptive techniques of data hiding in gray scale & color images.

In [14] paper, the authors present a study of a new method for insertion of message in an image. The last two bits of pixel value are used for insertion and retrieval of message. If the last two bits of pixel value are 00 or 10, we can insert 0, else by adding/subtracting 1 at that pixel value we can insert 0. Similarly 1 is inserted if last two bits are 01 or 11. For increased security, message is embedded at pseudo random locations. The message is retrieved similarly based on the pixel values of the last two bits.

[15] Has proposed a new idea of inverted pattern (IP) LSB substitution approach to improve the quality of stego-image. [15] and [17] also brought a new idea of adaptive least-significant bit steganographic method based on pixel value differencing (PVD) to achieve high embedding capacity. But it failed to achieve high PSNR. This problem is solved by [18] Reversible data hiding in encrypted images is a challenging one was proposed by [19] The concept is that, the secret message owner can encrypt the cover image with an encryption key. Then

LSB of that compressed image is encrypted to accommodate the additional data using the data hiding key. Hence, encryption key and data hiding key are necessary to retrieve the secret message and additional data at the receiver side separately. If the additional data is too large, it affects the quality of stego image. [20] Handled the lossless image data hiding based on modulo-256 addition. But this method generates a salt and pepper noise which affects the image quality [18] mainly focused on the distortion tolerance during the data hiding. Hiding the data directly in the spatial domain reduces the security. [21] Improved the embedding capacity by LSB substitution method of data hiding. This method produces distortion in the stego-image when the embedding rate becomes high.

JSteg Algorithm

JSteg algorithm is one of the steganographic techniques for embedding data into JPEG images. JSteg algorithm replaces LSBs of quantized Discrete Courier Transform (DCT) coefficients. In this process the hiding mechanism skips all coefficients with the values of 0 or 1. This algorithm is resistant to visual attacks and offers an admirable capacity for steganographic messages. Normally, JSteg embeds only in JPEG images. In these JPEG images, the content of the image is transformed into 'frequency coefficients' so as to achieve storage in a very compressed format. There is no visual attack in the sense presented here, due to the influence of one steganographic bit up to 256 pixels [3] and it is less immune for statistical attacks

F5 Algorithm

F5 algorithm was introduced by German researchers Pfitzmann and Westfeld in order to avoid the security problem when embedding the data into the JPEG images. The F5 algorithm embeds the message into randomly chosen Discrete Courier Transform (DCT) coefficients. It utilizes matrix embedding which minimizes the changes to be made to the length of certain message. The F5 Algorithm provides high steganographic capacity, and can prevent visual attacks. F5 algorithm is also resistant to statistical attacks. This algorithm uses matrix encoding such that it reduces the number of changes needed to embed a message of certain length. This algorithm avoids the chi-square attack since it doesn't replace or exchange the bits. The resistance is high for both visual and statistical attacks. It has high embedding capacity that is greater than 13%. This algorithm supports TIFF, BMP, JPEG and GIF formats [2].

EVALUATION OF THE STEGANOGRAPHIC ALGORITHM

The most important requirement is that a Steganographic algorithm has to be imperceptible. Below criteria has been proposed for imperceptibility of an algorithm:

1) ***Invisibility***- The invisibility of a steganographic algorithm is most important requirement. Strength of steganography lies in its ability to be unnoticed by human eye. The moment that one can see that an image has been tampered with, the algorithm is compromised.

2) ***Payload capacity***- Watermarking, needs to embed only a small amount of copyright information; in other words, Steganography requires sufficient embedding capacity.

3) ***Robustness against statistical attacks*** – Statistical Steganalysis is the practice of detecting hidden information by applying statistical tests on image data. Many Steganographic algorithms leave a "signature" when embedding information that can be easily detected through statistical analysis.

4) ***Robustness against image manipulation*** –While being transmitted the image may undergo changes by an active attacker in an attempt to remove hidden Information. Image manipulation, such as cropping or rotating, can be performed on the image. This may destroy the hidden message. It is required for Steganographic algorithms to be robust against malicious changes to the image.

5) ***Independent of file format*** – The most powerful Steganographic algorithms thus possesses the ability to embed information in any type of file.

6) ***Unsuspicious files*** – This requirement includes all characteristics of a Steganographic algorithm that may result in images that are not used normally and may cause suspicion. Abnormal file size, for example, is one property of an image that can result in further investigation of the image by a warden.

Application of Steganography

Steganography is applicable to, but not limited to, the following areas.

1) Confidential communication and secret data storing

2) Protection of data alteration

3) Access control system for digital content distribution

4) Media Database systems

III. SYSTEMS ANALYSIS AND DESIGN

3.1 METHODOLOGY

Because the requirements of the new system must be well understood before the system is designed and implemented, this paper adopts the Waterfall System Development Life Cycle model.
At the requirement stage, interactions were made with selected stakeholders and other literature were consulted to know their position on common security systems and their expectations from a new system [8] and [9] spectrum statistics and mel-ceptrum coefficients of audio signal for steganography.

With this, three levels of requirements were obtained:

1. Business processes/procedures: for example, use the system and a key to hide a given file in a selected image such that the addition of the file to the image is not noticeable to human vision, deliver the new object to the unauthorized user. Let the recipient use the system with the same key to extract the original message from the image without any alteration of the text.

2. Users or functional requirements: it involves the inputs and outputs e.g. digital file, key, image and stego-object.

3. Non-functional requirements: defining the expected behavior of system e.g. Usability, Data integrity, Reliability, Maintainability, etc.

3.2 SYSTEM ANALYSIS

The algorithm used for Encryption and Decryption in this application provides using several layers in lieu of using only LSB layer of image. Writing data starts from last layer (LSB layer); because the significance of this layer is least and every upper layer has double significance than its down layer. So every step we go to upper layer, the image quality decreases and image retouching transpires.
This project has two methods – Encrypt and Decrypt. In encryption the secret information is hidden in any type of image file. Decryption is getting the secret information from image file.

The data hiding patterns using the steganographic technique in this project can be explained using three phases:
1. Encryption phase
2. Transmission phase
3. Decryption phase

ENCRYPTION PHASE:
The "Encryption phase" uses two types of files for encryption purpose. One is the secret file which is to be transmitted securely, and the other is a carrier file such as image. In the encryption phase the data is embedded into the image using "Least Significant Bit algorithm" (LSB) by which the least significant bits of the secret document is arranged with the bits of carrier file such as image, Such that the message bits will merge with the bits of carrier file.
In this research, the symmetric key encryption with a single key is used.

TRANSMISSION PHASE
The transmission phase is one of the important sections for sending the data to destination securely. Usually e-mail or web is used for transferring the data. If the person hacks the e-mail or web and obtains the image, the secret key helps to protect it from unauthorized modification.

DECRYPTION PHASE:
The Decryption phase is the reverse of encryption phase. In decryption phase, the carrier image in which the data is hidden is given as an input file. The decryption phase uses the same password which was used for the encryption. After giving the correct password the decryption section uses the "Least Significant bit Algorithm" (LSB) by which the encoded bits in the image is decoded and turns to its original state thereby giving the output of a text document as well as an image.

3.3 SYSTEM DESIGN

3.3.1 DATA FLOW DIAGRAMS:

3.3.1.1 Data Flow Diagram Level 0

See Figure 2

3.3.1.2 Data Flow Diagram Level 1

See Figure 3

3.3.1.3 Data Flow Diagram Level 2

See Figure 4

IV. SYSTEM IMPLEMENTATION

4.1 HARDWARE/SOFTWARE REQUIREMENT

HUBs, LAN and Routers are needed for building the communication media from receiver to the sender. Other

hardware that can be used include: Processor 1.0MHZ or higher, Hard disk 150GB or higher, RAM 1GB or higher. Minimum software requirements for the project: Windows 7 or higher, Microsoft Visual Studio 2010 or higher (**C#**).

4.2 PROGRAM CODES/SCRIPTS IMPLEMENTATION

In this paper the application was developed on the visual studio platform using C#, the interfaces (front end) were designed to be simple with the form designer feature of visual studio; a fixed picture is used to hide the text for encryption.

The **"using System.Security.Cryptography library:"** is used in implementing the back end. It provides cryptographic services including secure encoding and decoding of data as well as many other operations, such as hashing, random number generation, etc.

The application is compiled by the command-line compiler provided by the Microsoft.NET framework which also executes the compiled C# application.

4.3 SYSTEM INTERFACES

Login Page:

Fig 5 shows the login page which is used to authenticate the user identity. The system check both the user name and password of the user. These two data are used to validate the user's input to acertain that it is not an inruder attempting to break through the system. After succcesful validation of the user name and password, the system proceeds to the main menu where user can carry out encryption and decryption, otherwise the system alerts the user for invalid login details. This module is the first and an additional security measure incoporated into the new system.

Activity Page:

Fig 6 shows the main menu of the new system where user can carry out both encryption and decryption. This can be done by entering plaintext into the provided textbox after which the checkbox must be checked to enter an encrypted password and then a stego image must be chosen where the message will be hiding. Finally, the user clicks on the Encrypt Message button and save the stego image by clicking on File => Save => Image.

The reverse of this process is the decryption phase where the user selects the stego image by clicking File => Open => Image, enter the encryption if required and click on the button Decrypt Message to get the content of the message which is inside the image.

Help Page:

This section (Figure 7) gives the user guide on how to use the new system from the encryption section to the decryption of the image.

Pseudocode for Least Significant Bit (LSB)

The Least Significant Bit which is often found to be 8th bit of some or all of the bytes inside an image is usually changed to a bit of the secret message. Digital images are mainly of two types

(i) 24 bit images
(ii) 8 bit images

In 24 bit images, three bits of information in each pixel can be embedded, one in each LSB position of the three eight bit values. Increasing or decreasing the value by changing the LSB does not change the appearance of the image; much so the resultant stego image looks almost same as the cover image. While in 8 bit images, one bit of information can be hidden. If the LSB of the pixel value of cover image C(i,j) is equal to the message bit m of secret massage to be embedded, C(i,j) remain unchanged; if not, set the LSB of C(i, j) to m. The message embedding procedure is given below:

$S(i,j) = C(i,j) - 1$, if $LSB(C(i,j)) = 1$ and $m = 0$
$S(i.j) = C(i,j)$, if $LSB(C(i,j)) = m$
$S(i,j) = C(i,j) + 1$, if $LSB(C(i,j)) = 0$ and $m = 1$
where LSB (C(i, j)) stands for the LSB of cover image C(i,j) and m is the next message bit to be embedded. S(i,j) is the stego image

Each pixel is made up of three bytes consisting of either a 1 or a 0
The hidden image is extracted from the stego-image by applying the reverse process

Data Embedding

The embedding process is as follows.
Inputs: Cover image, stego-key and the text
Output: stego image

Procedure

Step 1: Extract the pixels of the cover image

Step 2: Extract the characters of the text.
Step 3: Extract the characters from the Stego key
Step 4: Choose first pixel and pick characters of the Stego key and place it in first component of pixel.
Step 5: Place some terminating symbol to indicate end of the key. 0 has been used as a terminating symbol in this algorithm.
Step 6: Insert characters of text in each first component of next pixels by replacing it.
Step 7: Repeat step 6 till all the characters has been embedded.

Data Extraction
The extraction process is as follows.
Inputs: Stego-image file, stego-key
Output: Secret text message.
Procedure:
Step 1: Extract the pixels of the stego image.
Step 2: Start from first pixel and extract stego key characters from first component of the pixels. Follow Step3 up to terminating symbol, otherwise follow step 4.
Step 4: If this extracted key matches with the key entered by the receiver, then follow Step 5, otherwise terminate the program.
Step 5: If the key is correct, then go to next pixels and extract secret message characters from first component of next pixels. Follow Step 5 till up to terminating symbol, otherwise follow step 6. Step 6: Extract secret message

Pseudocode for the System

```
-Login (validate users' login credentials using their user name and password)
Begin
Valid_input_password := false
Valid_input_username := false
Login := false
While(login == false) do
Begin
Print("please enter Username: ") //input username
Readin(input_username)
Print("please enter password: ") //input password
Readin(input_password)
//Begin matching process

If input_username and input_password is in login_directory then
Valid_input_password := true
Valid_input_username := true
Login := true
steg.Show()
this.hide();
Else
Print("Invalid username or password please try again ")
End if

 // Input secret message to generate the key from the shared secret and the salt
Let key = new Rfc2898DeriveBytes()
// Create a RijndaelManaged object
   let  Bind = new RijndaelManaged()
    Bind.Key = key.GetBytes(aesAlg.KeySize / 8)
// Create a decryptor to perform the stream transform.
            ICryptoTransform encryptor =
aesAlg.CreateEncryptor(aesAlg.Key, aesAlg.IV);

Let save_dialog = new SaveFileDialog()
     let    save_dialog.Filter = "Png Image|*.png|Bitmap Image|*.bmp"
if (save_dialog.ShowDialog() == DialogResult.OK) then
bmp.Save(save_dialog.FileName, ImageFormat.Png)

plaintext = "plain text"
if (string.IsNullOrEmpty(plainText)) then
 throw new ArgumentNullException("plainText")
end if
 if (string.IsNullOrEmpty(sharedSecret)) then
            throw new
ArgumentNullException("sharedSecret")
end if
let bmp = (Bitmap)imagePictureBox.Image;
let extractedText =
SteganographyHelper.extractText(bmp)
if (encryptCheckBox.Checked) then
extractedText = Crypto.DecryptStringAES(extractedText, passwordTextBox.Text);
end if
```

Test Data used for the implementation is:
Good Day Sir,
This is to inform you that login detail to the server is
Username: Server_@School
Password: P@ssword
Kindly note the capital letter Sir.
Thanks

V. DISCUSSION

Since ancient times, man has found a desire in the ability to communicate covertly. The recent explosion of research in watermarking to protect intellectual property is evident that steganography is not just limited to military or espionage applications. Steganography, like cryptography, will play an increasing role in the future of secure communication in the "digital world".

Steganography is a really interesting subject which is outside of the mainstream cryptography and system administration that most of us deal with day after day.

Steganography can be used for hidden communication. This paper shows the implementation of the image steganography system using LSB approach to provide a means of secure communication. A stego-key has been applied to the system during embedment of the message into the cover image. That same stego-key is applied in extracting the message from the cover image.

This steganography application software is provided for the purpose of using image of any format to hide any type of file.

VI CONCLUSION

The design and implementation of a steganographic application software provided for the purpose of using images to hide any type of file will make up for the problem of message identification inherent in cryptography. This will ensure secure information delivery is increased with the system. This software will be useful to both individuals and organizations/business in protecting their privacy, earning them trust while boosting their profitability.

Future research work on more efficient and advanced techniques in steganalysis can be done that will help law enforcement agencies to better detect illicit materials transmitted through the Internet.

ACKNOWLEDGEMENTS
The authors would like to thank and appreciate the reviewers and everyone else who gave support and ideas in bringing the paper to its present level.

REFERENCES

[1] T. Morkel, J. H.P. Eloff, M. S.Oliver, "An overview of image steganography,"
ISSA, pp. 1-11, 2005.

[2] J. Fridrich, M. Goljan, D. Hogea, "Steganalysis of JPEG Images: Breaking the F5 Algorithm," Information Hiding, 5th International Workshop, IH, 2002, pp.310-323.

[3] A. S. Sakr, M. Amin, H. M. Abdullkader and H. M. Ibrahem, "A steganographic method based on DCT and new quantization technique," International Journal of Network Security, Vol.16, No.4, pp. 265-270, July 2014.

[4] K.A. Zaidoon, A.A. Zaidan, B.B Zaidan, H.O. Alanazi, "Overview: main fundamentals for steganography," Journal of Computing, 2(3), pp. 40-43, 2010.

[5] 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," Journal of Systems Engineering and Electronics, Vol. 29, Issue 3, pp. 639-649, 2018;

[6] S. N. Gowda and S. Sulakhe, "Block Based Least Significant Bit Algorithm for Image Steganography," in conf. Annual International Conference on Intelligent Computing, Computer Science & Information Systems (ICCSIS-16), Pattaya, Thailand, April 2016, pp. 16-19;

[7] A. T. Al-Tamimi and A. A. Alqobaty, 'Image Steganography using Least Significant Bits (LSBs): A Novel Algorithm', International Journal of Computer Science and Information Security, Vol. 13, No. 1, pp. 1-5, 2015; http://sites.google.com/site/ijcsis

[8] N.N.H. Al Dawla, M.M. Kazi and K.V. Kale, "Steganography enhancement by combining text and image through wavelet technique", International Journal of Computer Applications (0975–8887), Vol. 51, No. 21, August 2012.

[9] N. Liu, P. Amin and K.P. Subbalakshmi, "Security and robustness enhancement for image data hiding", IEEE

Transactions on Multimedia, Vol. 9, No. 3, pp. 466–474, 2007.

[10] C.C. Chang, N.H. Lin and Hu, Y.C. "A fast and secure image hiding scheme based on LSB substitution", International Journal of Pattern Recognition and Artificial Intelligence, Vol. 16, No. 4, pp. 399–416, 2002.

[11] Lin, G.S., Chang, Y.T. and Nunglie, W. "A framework of enhancing image steganography with picture quality optimization and anti-stegnalysis based on simulated annealing algorithm", IEEE Transactions on Multimedia, Vol. 12, No. 5, pp. 345–357, 2010.

[12] S. A. Laskar and K. Hemachandran, "Steganography Based On Random Pixel Selection For Efficient Data Hiding", International Journal of Computer Engineering and Technology, Vol.4, Issue 2, pp. 31-44, 2013.

[13] S.S. Priya, K.Mahesh and Dr.K.Kuppusamy, "Efficient Steganography Method To Implement Selected Least Significant Bits in Spatial Domain", International Journal of Engineering Research and Applications, Vol. 2, Issue 3, pp. 2632-2637, 2012.

[14] R. Yadav, "A Novel Approach For Image Steganography In Spatial Domain Using Last Two Bits of Pixel Values", International Journal of Security, Vol. 5 Issue 2, pp. 51-61, 2011.

[15] C.H. Yang, C.Y. Weng, S.J. Wang and H.M. Sun, "Adaptive data hiding in edge areas of images with spatial LSB domain systems", IEEE Transactions on Information Forensics and Security, Vol. 3, No. 3, pp. 488–497, 2008.

[16] M. Wu and B. Liu, "Data hiding in binary image for authentication and annotation", IEEE Transactions on Multimedia, Vol. 6, No. 4, pp. 528–538, 2004.

[17] X. Liao, Q.Y. Wen and Zhang, J. "A steganographic method for digital images with four-pixel differencing and modified LSB substitution", J. Vis. Commun. Image R, Vol. 22, No. 1, pp. 1–8, 2011.

[18] L, Fillatre, "Adaptive steganalysis of least significant bit replacement in grayscale natural images", IEEE Transactions on Signal Processing, Vol. 60, No. 2, pp. 556–569, 2012.

[19] X. Zhang, "Separable reversible data hiding in encrypted image", IEEE Transactions on Information Forensics and Security, Vol. 7, No. 2, pp. 826–832, 2012.

[20] Z. Ni, Y.Q. Shi, N. Ansari, W. Su, Q. Sun and X. Lin, "Robust lossless image data hiding designed for semi-fragile image authentication", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 18, No. 4, pp. 497–509, 2008.

[21] C.C. Lin, W.L. Tai and C.C. Chang, "Multilevel reversible data hiding based on histogram modification of difference images", Pattern Recognition, Vol. 41, No. 12, pp.3582–3591, 2008.

[22] S. U. Maheswari and D. J. Hemanth, "Different methodology for image steganography-based data hiding: Review paper" International Journal of Information and Communication Technology, Volume 7, Issue 4/5, pp. 521-536, July 2015.

[23] F.M. Shelke, A. A. Dongre, P.D. Soni, "Comparison of different techniques for Steganography in images," International Journal of Application or Innovation in Engineering & Management (IJAIEM) Vol. 3, Issue 2, pp. 175, Feb 2014.

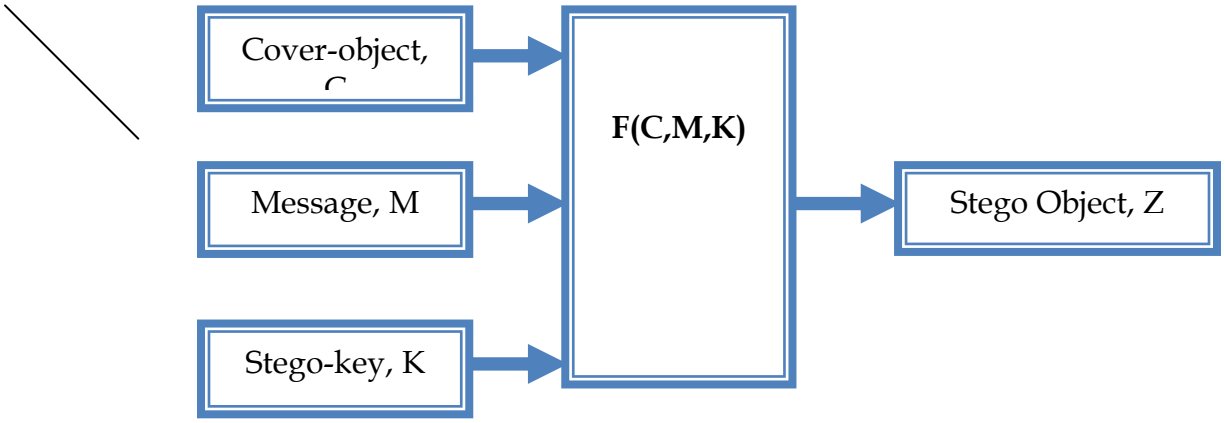


Figure 1: Model of Steganography

Table 1: Comparison of image Steganography techniques

	Invisibility	Payload capacity	Robustness against statistical attack	Robustness against image manipulation	Independent of file format	Unsuspicious files
LSB in BMP	High	High	Low	Low	Low	Low
LSB in GIF	Medium	Medium	Low	Low	Low	Low
JPEG	High	Medium	Medium	Medium	Low	High
Spread Spectrum	High	Medium	High	Medium	High	High

(Falesh. et.al., 2014) [24]

Figure 2: Data Flow Diagram Level 0

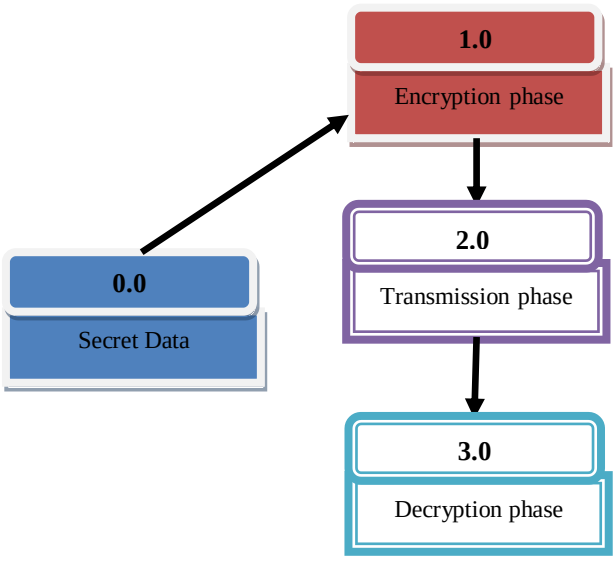


Figure 3: Data Flow Diagram Level 1

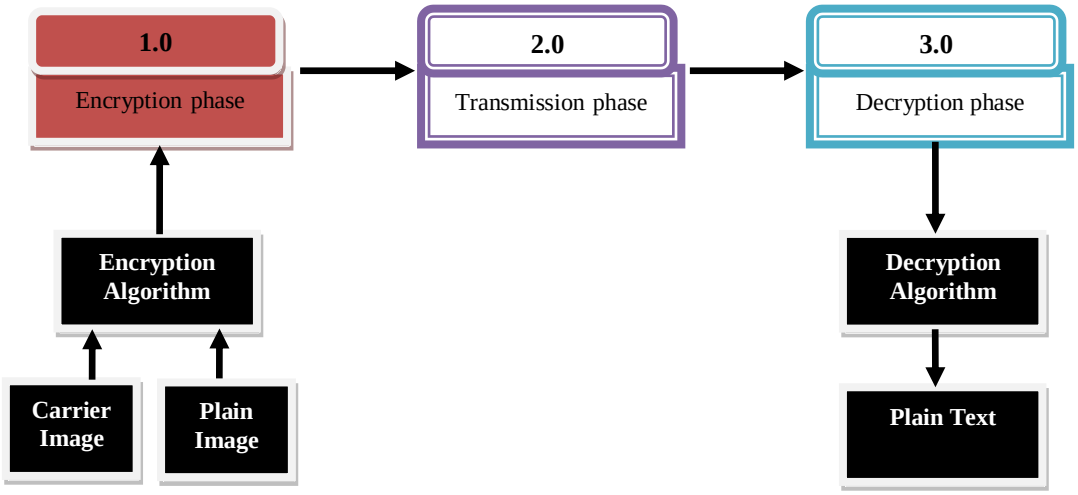
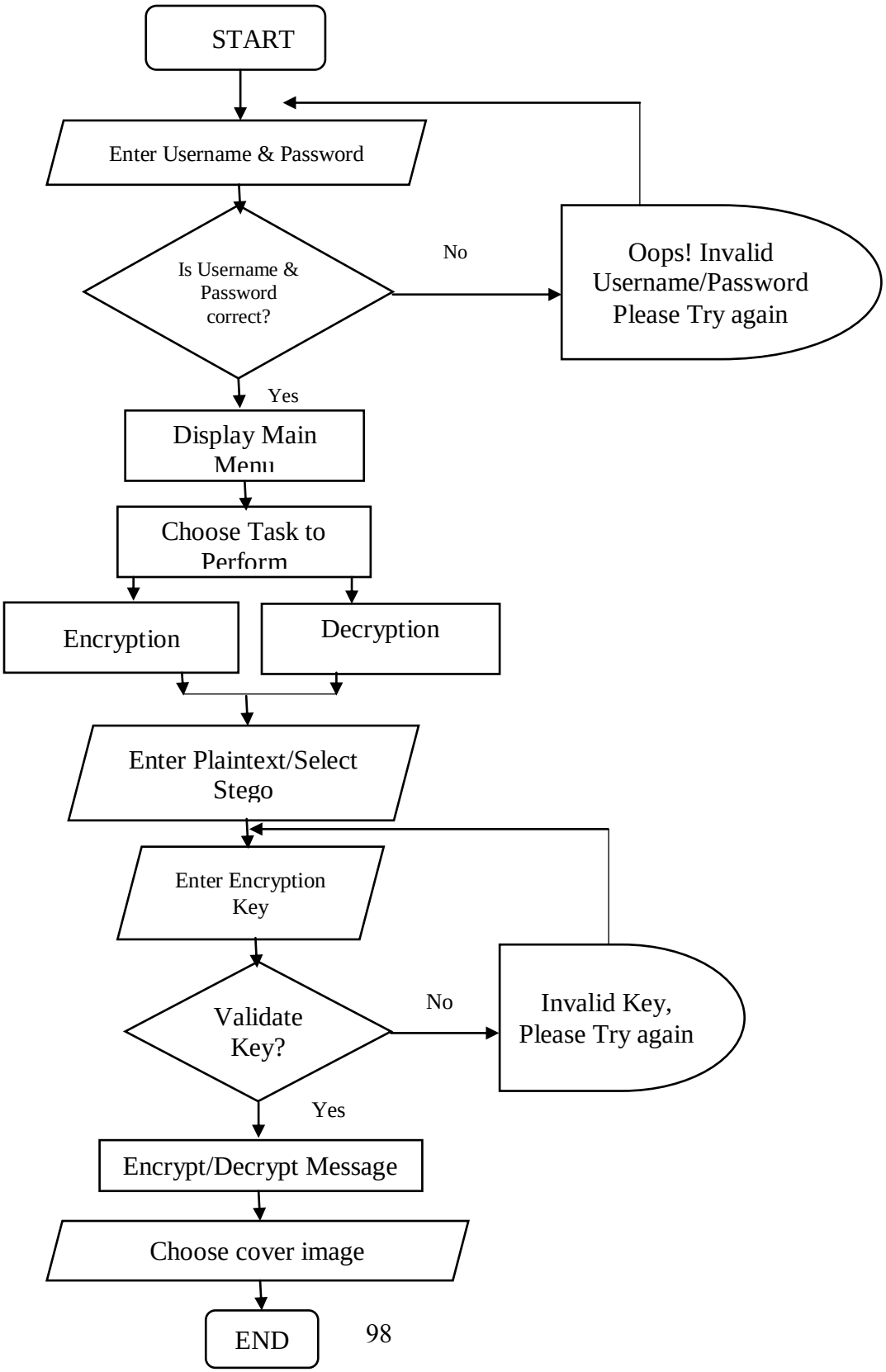


Figure 4: Data Flow Diagram Level 2

System Flowchart



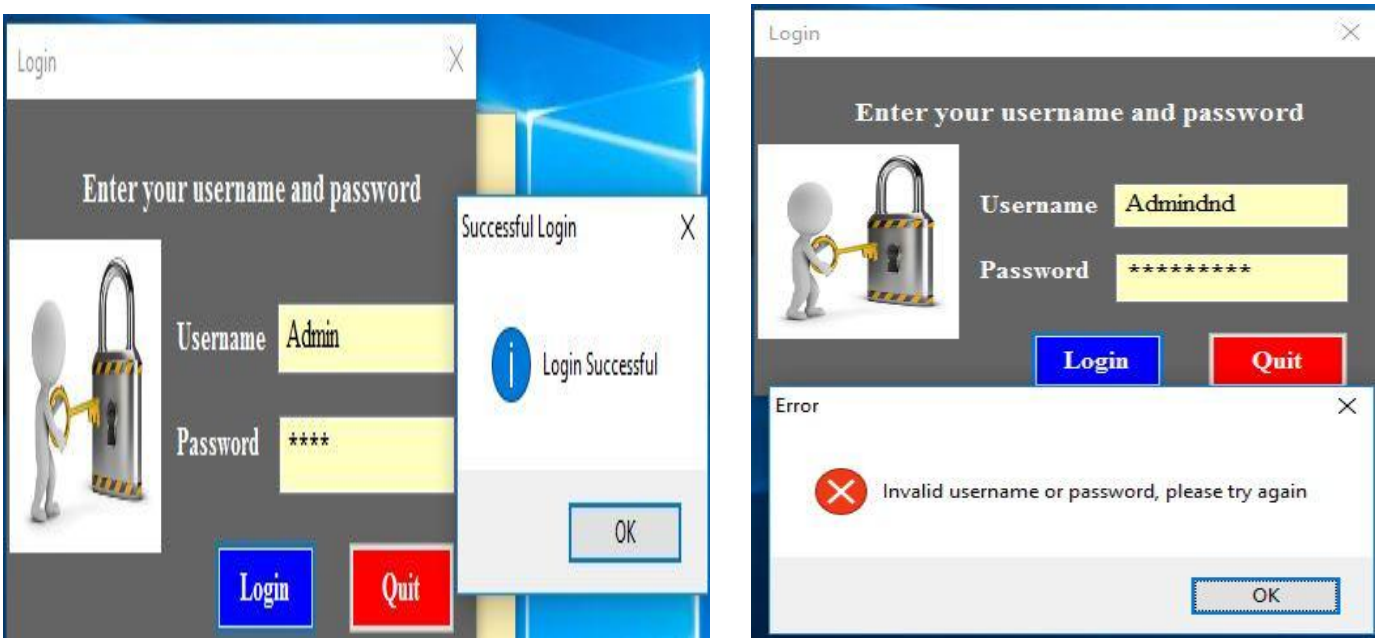


Figure 5: Login Page

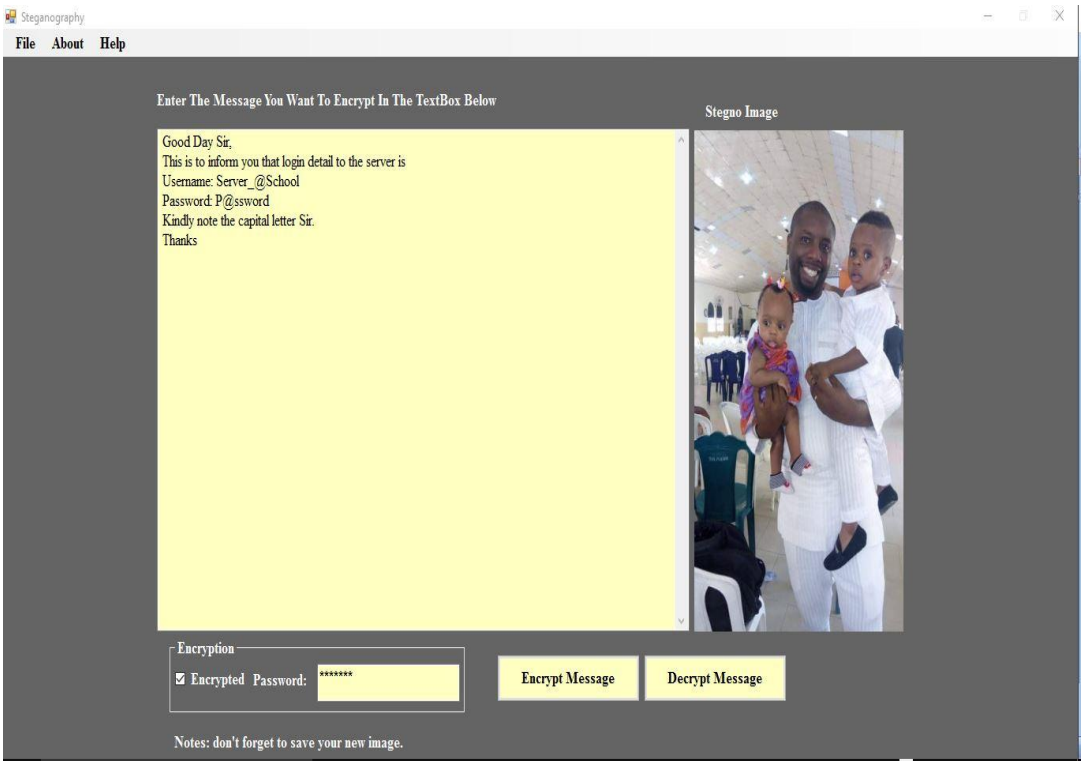


Figure 6: Activity Page

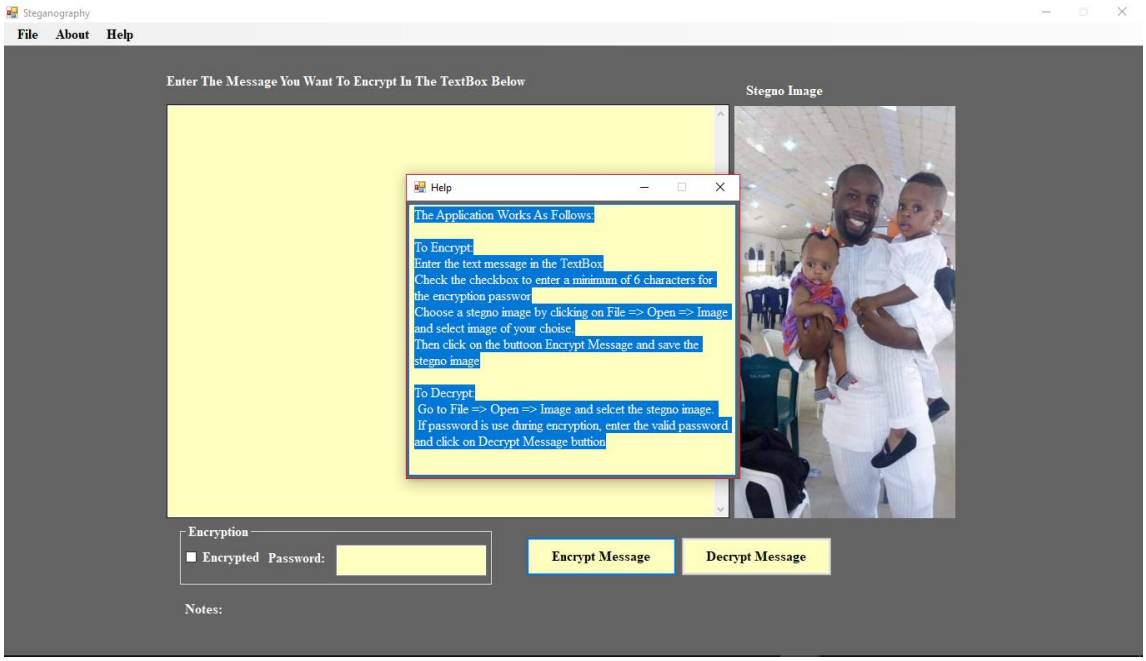


Figure 7: Help Page