ISSN 2320 -2610
Volume 6, No.4, July - August 2017
International Journal of Multidisciplinary in Cryptology and Information Security
Available Online at http://www.warse.org/IJMCIS/static/pdf/file/ijmcis02642017.pdf

Encryption of Text in Image over a Network

Olatunji K. A.[1], Afolalu C. O.[2], Oguntimilehin A.[3]
[1]Afe Babalola University Ado Ekiti, Ekiti State, Nigeria
olatunjika@abuad.edu.ng
[2]Afe Babalola University Ado Ekiti, Ekiti State, Nigeria
catherineea@abuad.edu.ng
[3]Afe Babalola University Ado Ekiti, Ekiti State, Nigeria
ebenabiodun2@yahoo.com

Abstract – An increase in number of attack recorded during electronic exchange of information between the sender and the recipient in the recent years has called for a way to protect data in a robust manner. Most data transmitted over a network is sent in clear text making it easy for intruders to capture and read sensitive information. Encryption algorithms protect data from unwanted person and make sure that only the intended recipient can decipher and read the information.

In this project a three tier design architecture process consisting of Blowfish algorithm for cryptography, least significant bit algorithm for steganography and handshaking a client server TCP/IP protocol, was used to develop a model for an encryption of text in image over a network system "a method for securing messages" which encrypts plain text into images before sending them to recipient.

Cryptography is simply the translation of data into a secret code, and it is considered the most effective way to ensure data security. To read an encrypted file, you must have access to a secret key or password that enables you to decrypt it. Modern encryption is achieved using algorithms with a "key" to encrypt text or other data into digital nonsense and then decrypting it by restoring it to its original form.

Steganography hides the existence of a secret message and in the best case nobody can see that both parties are communicating in secret. This makes steganography suitable for some tasks for which cryptography is not. The model was implemented using socket programming (Visual C#.net) to design the user interface. Every part of the systems was tested and found working.

Keywords: Encryption, Cryptography, Steganography, Cypher, Network.

1. INTRODUCTION

One of the reasons that intruders can be successful is that most of the information they acquire from a system is in a form that they can read and comprehend. Intruders may reveal the information to others, modify it to misrepresent an individual or organization, or use it to launch an attack. Solutions to this problem may involve the use of steganography or the use of cryptography or even combining the two of them together. Steganography and cryptography are techniques used for hiding information in digital media. Steganography hides the existence of the message from others while cryptography scrambles the message so you know it is there but cannot access it without a key [1].

Cryptography and steganography are well known and widely used techniques that manipulate information in order to cipher or hide their existence respectively. Cryptography scrambles a message so it cannot be understood. It involves converting a message into an unreadable cipher. A large number of cryptography algorithms have been created till date with the primary objective of converting information into unreadable ciphers [2].

Cryptography systems can be broadly classified into symmetric-key systems and public key systems. The symmetric key systems make use of a common key for encryption and decryption of the message. This key is shared privately by the sender and the receiver. The sender encrypts the data using the joint key and then sends it to the receiver who decrypts the data using the same key to retrieve the original message. The public-key systems use a different key for encryption as the one used for decryption. Public key systems require each user to have two keys – a public key and a private key (secret key). The sender of the data encrypts the message using the receiver's public key. The receiver then decrypts this message using his private key. Even though both Steganography and cryptography provide security, studies are made to combine both cryptography and Steganography methods into one system for better confidentiality and security [3].

2. REVIEW OF RELATED LITERATURE

The reviews of the following researches were carried out as follows:

1. Mamta and Parvinder implemented an Improved Steganographic Technique for 24-bit Bitmap Images in Communication [4]. They discovered that terrorist groups abuse technology by hiding maps and photographs of terrorist targets and so on; that is the reason they study steganography so as to allow innocent messages to be placed in digital media as well as intercept abuse of this technology. Another thing that they realized in the course of reviewing is that Least Significant Bit technique works well for 24-bit color image files, while Steganography has not been as successful when using an 8-bit color image file, due to limitations in color variations and the use of a colormap.

The method used was Least Significant Bits in which the least significant bit of every byte is altered to form the bit-string representing the embedded file. A colormap consisting of (240, 24 bits) colors instead of the normal (256, 24 bits) was created so as to reduce the noise of the picture, the color-map was then optimized to provide the best 240 colors which makes use of Linde-Buzo-Gray optimization algorithm methodology. Then sorting of the color map was done in order to reduce noise so that similar colors are next to each-other before the pixels are assigned to color-map colors. Beginning with the first color in the colormap array, the pixel that is the closest in color to the starting pixel is found using the mean absolute error measure. If the best match to a color results in an error level greater than 100 meaning that there really was not a very good match to the color), a new color is created in the first open slot (using the sixteen extra spaces in the colormap) and this new color is used as the pair.

After sorting the colormap, an 8-bit image was created by assigning pixels to the color map color. This was achieved by choosing one pixel from the original 24-bit image pixels and its RGB values are compared to the RGB values of every color in the colormap. For each comparison an error level is calculated using the mean absolute error of the red, green, and blue color components. The colormap color that produces the smallest amount of error is the colormap color that gets assigned to this pixel. Finally, encoding the data was done by first changing the text to its binary equivalent. In order to do this, each character of the text message is converted to its ordinal number (example: 'a' = 97). The ordinal number is then converted to binary using the following method called the division-remainder routine. An ordinal number is divided by two using the *mod ()* function. This function returns either a one or a zero, which is then placed in a remainder array. This is continued until the dividend is zero. The ones and zeros in the remainder array is the binary equivalent of the ASCII ordinal number. Then, once all characters have been converted, the binary data is embedded in the image by sequentially altering the least significant bit of the image data as necessary.

Normally spies and criminals wanting to pass on secrets in inconspicuous data over public networks can abuse the same applications that are being developed to allow copyrights to be placed in digital media. His work employs use of only one bit in the least significant bit insertion instead of using variations such as two or more of the least significant bit to embed data. Since this research only allow text to be embedded, there is need to also embed pictures in JPEG and GIF or MS documents.

2. Using Steganography to Hide Messages inside PDF Files was developed by Alizadeh et al., [5]. Techniques used by previous researchers do not increase the file size but the amount of data that can be embedded is very limited by the number of white spaces in the text [6]. Others embeds data by altering text in a visible way (change the value of some text state variables), then writes an incremental update containing the original PDF data, so the altered text is not actually displayed. Afterward data was embedded by writing incremental updates for objects that do not exist in the original data, so that the update has no effect. The data is embedded in the value of the stream objects used in the update. And finally, data was embedded by writing incremental updates with a given length for several objects; then the data can be retrieved by reading the cross-reference section of the update, for it includes the start address of each updated object [7]. Another researcher stated that justifying a text (so that it is aligned both with the left and right margin) using a PDF writer would produce random values for the TJ operators that are used to position the characters. It would then be possible to hide data in the least significant bits of some of these TJ operator values. However this works only when the TJ operator values are random and do not contain any pattern [8].

The improved and recommended algorithm to hide data in PDF documents is a combination of the original TJ algorithm and the improvements. PDF created from LATEX source file was used as a basis, and then used chunks of 4 bits to hide the input data in TJ values. The input data was encrypted before it was embedded in the stego-file to keep the distribution of TJ values as close as possible to the original distribution. Two ranges of TJ values ([-447, -337] and [-320, -257]) were selected as possible sources to hide the input data. This was done to avoid changing TJ values that have a very low or very high frequency. This also means that most TJ values would be used to hide data instead of only the values between [-16, 16]. To make it impossible to notice the difference in the PDF output and to counter an attack that calculates and compares the line widths, some TJ values were used to compensate for the changes in the line widths that are introduced. At last, the randomization and redundancy features that are part of the original algorithm were discarded in favour of extra capacity.

According to his work multiple improvements to the steganographic security have been incorporated in the new algorithm to protect it against statistical analysis but this does not mean that it is secure against other methods that are not yet researched during the project. One method described here could be to look at the TJ value distribution of specific

character pairs. Although several improvements to the embedding capacity have been incorporated in the new algorithm, it is not yet proven how much capacity gain has been obtained.

3. Amin et.al researched on Information Hiding Using Steganography [1]. They took their inspiration from the fact that the number of data being exchanged on the Internet is increasing daily. Therefore, the confidentiality and data integrity are required to protect against unauthorized access and use.

They proposed the Secure Information Hiding System (SIHS) based on the Least Significant Bit (LSB) technique in hiding messages in an image using discrete logarithm. The system enhanced the LSB technique by randomly dispersing the bits of the message in the image and thus making it harder for unauthorized people to extract the original message. While using LSB, the message is embedded with sequence-mapping technique in the pixels of a cover-image. Although LSB hides the message in such way that the humans do not perceive it, it is still possible for the opponent to retrieve the message due to the simplicity of the technique. SIHS overcome the sequence-mapping problem by embedding the message into a set of random pixels, which are scattered on the cover-image. The bits of the secret message are embedded in pixels of the cover-image that are generated by discrete logarithm calculation. Discrete logarithm calculation can be used to solve sequence-mapping problem. The main idea here is to generate random numbers without any repetition. With this set of random numbers, a random-mapping can be done.

The stego process starts with the selection of a cover-image to hide a message. The user will then select a key k, which will depend on the size of the message, m and the image, I. The value of k lies in the range, $m < k < I$. after that a prime number, p is obtained by searching for the first prime number that exceeds the key, k. Then a primitive root, *a*, is derived by using $a, a^2, .., a^{p-1}$ equation. The primitive root, *a*, is then used to generate a set of random numbers, y_i. This set of random numbers will determine the position of the pixel to embed the bits from the message. this derivation and calculations are done using discrete logarithm. The message bits are then mapped onto the cover-image by the stego system encoder in the following manner: M_i -----Iy_i, M_i refers to the i^{th} bit of the message, while Iy_i is the i^{th} random number generated. After this their encoding/embedding process was done. Then to decode their stego image and recover the hidden message the stego key, k used in encoding the message will be required, both the sender and receiver must share the stego key during the communication. The key is then used for selecting the positions of the pixel where the secret bits had been embedded.

In their work they made use of random pixels from an entire image and were not able to utilize the entire image for embedding the message, therefore the processing time it takes to generate the random numbers takes a relative amount. It was also found out that the quality of the original object used would be degraded if some pixels are modified directly or indirectly.

4. Vyas and Pal proposed a Method in Image Steganography to Improve Image Quality with LSB Technique. LSB hiding technique works as it hides the secret message directly in the least two significant bits in the image pixels, which affects the image resolution, due to this it reduces the image quality and make the image easy to attack, it is clear that LSB changes the image resolution when the least significant bits add in the binary image format, so that image quality become burst and there become so much difference in the original image and encoded image in the respect of image quality. These two problems listed above are the main reason [9] proposed this method in image steganography.

They overcame the LSB technique problem of reduced image quality or blurred image by modifying the LSB. The basic idea of this proposed method is to choose one pixel of the image randomly; then selected this pixel as the centre of the image and divides the image into three parts: Red, Green and Blue separately according this Centre pixel. Now two by two bits of the secret message in each part of the pixel is hid by searching about the identical, if the identical is found or satisfied then the image is set with new values, if the identical is not found, they proposed hiding in the two least significant bits and set the image with new values. Then save the location of the hiding bits in binary table. This modification gave the same image quality as original image after the encoding. But there is need to improve the compression ratio of the image to the text.

5. [10] proposed Data Hiding Technique Text Image inside Image (TIII). The internet and the World Wide Web have revolutionized the way in which digital data is distributed. The Number of spies is increasing year after year and now it is time to hit the target to defend a secret message during transmission.

The technique (TIII) deals with a text as an image (text image) which will be hidden inside another image (cover image) to generate an image have the same details of the (cover image) but also have the hidden (text image) inside it and will be called (stego image). Text image in an image consist of two colors black and white, which has the message that will be hidden. This image consists of black and white pixels, which represent the visual representation of hidden message, because of the visual representation of the image pixels in black and white they then converted the image of the black and white pixels to image of zeros and ones (0, 1). Now to hide this text image of numbers (0,1) they brought a digital image (cover image) whose pixels have the values between (0 and 255) and has the same size like the text image. Then they modified their (cover image) by removing one value from a pixel having 255 hence every

pixel with a value of 255 will now have 254 instead, this modification is necessary to avoid generating any pixels that have values more than 255 and to keep them in the range of the grey scale of the images which is (0 -> 255). The stego image (new1) is now generated which has the appearance of the cover image and has the hidden text inside, and it will be exactly twice bigger than the cover image because, TIII will generate two pixels for the stego image where one pixel is meant to be, the first pixel of the stego image (new1) will be equal to the corresponding pixel in the cover image with respect of the increase in position. To generate the value of the second pixel of new1, TIII take the value of the first pixel of cover image plus the value of the first pixel at the text image with respect to the increase in position.

This methodology assumes that the text you want to hide already exists in an image which has constraint in colors (white and black), their work does not allow the text image in RGB colors. TIII assumes that the cover image must match the text image in the dimensions and size e.g. if the (text image) has dimensions of (7*7) then the (cover image) must also have dimensions of (7*7) Their proposed technique TIII did not addressed the difference in size between the cover image and the stego image.

6. Multilevel Network Security Combining Cryptography and Steganography on ARM Platform is designed and implemented by Pallavi et.al [11]. The fact that modern embedded systems need data security more than ever before since many embedded systems depend on obscurity to achieve e-mail from being read by someone other than the intended recipient birth this research. PDAs store personal e-mail and contact lists; GPS receivers and cell phones keep logs of user's movements and automobiles record peoples driving habits. With that users still demand products that can be reprogrammed during normal use enabling them to eliminate bugs and add few features as firmware upgrades becomes available.

The methodology here made used of Generated key from iris image because iris is considered to be most trusted and unique part of an individual. Key length used was 128 bits. They used Blowfish algorithm for encryption of the confidential information. Then this encrypted text then hides into every pixel of iris image. The Iris image is transmitted to receiver, at the receiver side, the hidden data is then removed from the encrypted image using same encrypted key, the original data recovered from encrypted text.

There is some weakness in hiding information in images; that is adversary could easily detect the confidential message, by noticing the noise and clarity of the image's pixels, also by observing the difference between the embedded image and the original one if it is known to him.

7. Channalli and Jadhav proposed to hide information on the output image of the instrument in real time, such as image displayed by an electronic advertising billboard – "Steganography an Art of Hiding Data"[12]. They discovered that due to lack of covertness on network channels, an eavesdropper can identify encrypted streams through statistical tests and capture them for further cryptanalysis even though encryption provides secure channels for communicating entities

An image was read from the source (billboard image to be displayed); it was broken down to smaller [R x C] blocks where R & C are the first & second bytes of the key respectively, each smaller block is a combination of many pixels of different values. Then the LSBs of the pixel are changed depending on the pattern bits and the secret message bits. The pattern bits are considered in sequence from its Most Significant Bit. If the pattern bit is 0, then the first LSB of the pixel is changed, if data bit is 1 and pixel bit is 0, then pixel bit is changed to 1 or else it is retained as it is. If the pattern bit is 1, then the second LSB of the pixel is changed accordingly. A single bit of the secret message is distributed throughout the block. This is done to have enough information so that correct information can be retrieved after decoding; similarly the other bits are inserted in the remaining blocks. If the length of the secret message is large, then it can be divided and stored in two or three frames. To extract the information, operations contrary to the ones carried out in embedding are performed.

The key plays a very important in embedding the message. The larger the key size, the more difficult to suspect the secrecy and also the smaller the key the easier it is to detect the secrecy. The key for embedding is smaller in this research which can allow third party to detect the hidden message easily.

8. Security using Image Processing was developed by [13]. This is to increase the security of transmitted data on the internet to a much needed higher level by using AES algorithm to encrypt the text message and embedded it in a part of the image thus making the text message difficult to find. The increase in cybercrime providing only network security is not sufficient. Security provided to images like blue print of company projects, secret images of concern to the army or of company's interest, using image steganography and stitching is beneficial.

First off they started by using blkproc function in mat lab to break an image of size w * h into n sub-images of size x * y, the message to be sent is then encrypted using AES algorithm; thus generating the cipher-text. After encryption they carried out the embedding phase, in this phase the encrypted message is embedded on to a part of the secret image, the cipher text that is given as input in the text editor is actually hidden in the cipher. Then LSB steganographic algorithm is used for hiding the cipher inside the image, here each bit of the cipher text (that has been converted into its binary equivalent) is exchanged with the last bit of each pixel value. Similarly, for each pixel the last bit is replaced with the consecutive bits of the cipher text,

next was the hiding phase, in this phase image steganography is performed. The technique used for image steganography is Kekre's Median Codebook Generation Algorithm (KNCG), finally comes the stitching phase which makes use of K-Nearest Neighbour or KNN algorithm; it is also a non-parametric technique, which means that no assumption is made about the parameters in this algorithm. The working is based on finding the minimum distance from the query instance to the training samples to determine the K-nearest neighbours to the query instance. After they found the k nearest neighbours, simple majority of these K-nearest neighbours is taken to be the prediction of the query instance.

The hidden message is limited to only 140 characters.

9. Sarmah and Bajpai proposed System for Data Hiding using Cryptography and Steganography[14]. Their main motivation was the inability of existing systems to properly show the efficiency of steganography and cryptography in the sense that a system that was intercepted and decrypted by a third party would instantly expose the hidden message therefore defeating the aim of encryption.

The research consisted of three modules: crypto module, security module and stego module. In crypto module text was inserted for encryption, AES algorithm was applied using 128-bit key (Key 1) and then cypher text was generated in hexadecimal form. In the security module, alphabets and digits in the cipher text were separated with the help of Separator 1and the original position of the alphabet and the digits in the form of a secret key (key 3) was kept tract of. Then the first seven alphabets retrieved from first step were separated and the remaining alphabets were added at the end of the separated digits as in the first step. This generated the second key (key 4). Stego module/process involves hiding the generated Cipher text by taken the first seven alphabets generated from the Security Module; scramble the alphabets using a 64-bit key (Key 2), took a Grey Scale Image, found the DCT of the Image, hid the Cipher by altering the DCT and then applied the Inverse DCT, the process which created the stego image.

To retrieve the cypher text, the process is repeated in the reverse manner. The research makes use of too many keys for both decryption and encryption in which without proper documentation it is easy to forget the keys.

3. THE DESIGN OF ENCRYPTION OF TEXT IN IMAGE OVER A NETWORK

The Design of Steganography and Cryptography for Sending Pictures over a Network is in three phases; firstly the messages from the sender is encrypted using blowfish algorithm, next the encrypted message is hid inside the picture using least significant bit algorithm and lastly the picture is sent over the network to the receiver. System architecture is presented in figure 1.

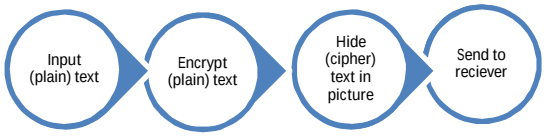

Figure1: System Architecture.

3.1 Blowfish Algorithm for Cryptography

Messages to be sent through the picture from the sender A to the receiver B is encrypted using the blowfish algorithm as explained below:

Blowfish provides a good encryption. Blowfish has a 64-bit block size and a variable key length from 32 bits up to 448 bits. It is a 16-round Feistel cipher and uses large key-dependent S-boxes. In structure it resembles CAST-128, which uses fixed S-boxes [15].

The figure 2 shows Blowfish's encryption routine. Each line represents 32 bits. There are five sub key-arrays: one 18-entry P-array (denoted as K in the diagram, to avoid confusion with the Plaintext) and four 256-entry S-boxes (S0, S1, S2 and S3). Every round r consists of 4 actions: First, XOR the left half (L) of the data with the r^{th} P-array entry, second, use the XORed data as input for Blowfish's F-function, third, XOR the F-function's output with the right half (R) of the data, and last, swap L and R.

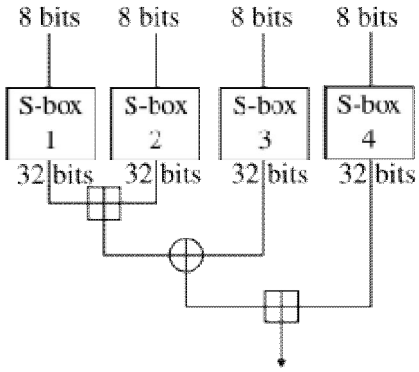

Figure 2: Blowfish Routine

The F-function splits the 32-bit input into four eight-bit quarters, and uses the quarters as input to the S-boxes. The S-boxes accept 8-bit input and produce 32-bit output. The outputs are added modulo 2^{32} and XORed to produce the final 32-bit output. After the 16th round, undo the last swap, and XOR L with K18 and R with K17 (output whitening). Decryption is exactly the same as encryption, except that P1, P2..., P18 are used in the reverse order. This is not so obvious because xor is commutative and associative. A common misconception is to use inverse order of encryption as decryption algorithm (i.e. first XORing P17

and P18 to the cipher text block, then using the P-entries in reverse order).

Blowfish's key schedule starts by initializing the P-array and S-boxes with values derived from the hexadecimal digits of pi, which contain no obvious pattern. The secret key is then, byte by byte, cycling the key if necessary, XORed with all the P-entries in order. A 64-bit all-zero block is then encrypted with the algorithm as it stands. The resultant cipher text replaces P_1 and P_2. The same cipher text is then encrypted again with the new sub keys, and the new cipher text replaces P_3 and P_4. This continues, replacing the entire P-array and all the S-box entries. In all, the Blowfish encryption algorithm will run 521 times to generate all the sub keys - about 4KB of data is processed. Because the P-array is 576 bits long, and the key bytes are XORed through all these 576 bits during the initialization, many implementations support key sizes up to 576 bits. While this is certainly possible, the 448 bits' limit is to ensure that every bit of every sub key depends on every bit of the key, as the last four values of the P-array does not affect every bit of the cipher text.

3.2 Least Significant Bit for Steganography

Least significant bit (LSB) insertion is a common and simple approach to embed information in an image file. In this method the LSB of a byte is replaced with an M‟s bit. This technique works well for image steganography. To the human eye the stego image looks identical to the carrier image. For hiding information inside the images, the LSB (Least Significant Byte) method is usually used. To a computer an image file is simply a file that shows different colors and intensities of light on different areas of an image. The best type of image file to hide information inside is a 24 Bit BMP (Bitmap) image. When an image is of high quality and resolution it is an easier to hide information inside image. Although 24 Bit images are best for hiding information due to their size. Some people may choose 8 Bit BMP‟s or possibly another image format such as GIF. The reason being is that posting of large images on the internet may arouse suspicion. The least significant bit i.e. the eighth bit is used to change to a bit of the secret message. When using a 24-bit image, one can store 3 bits in each pixel by changing a bit of each of the red, green and blue color components.

Least Significant Bit Algorithm

a. Select a cover image of size M*N as an input.
b. The message to be hidden is embedded in RGB component only of an image.
c. Use a pixel selection filter to obtain the best areas to hide information in the cover image to obtain a better rate. The filter is applied to Least Significant Bit (LSB) of every pixel to hide information, leaving most significant bits (MSB).
d. After that Message is hidden using Bit Replacement method.

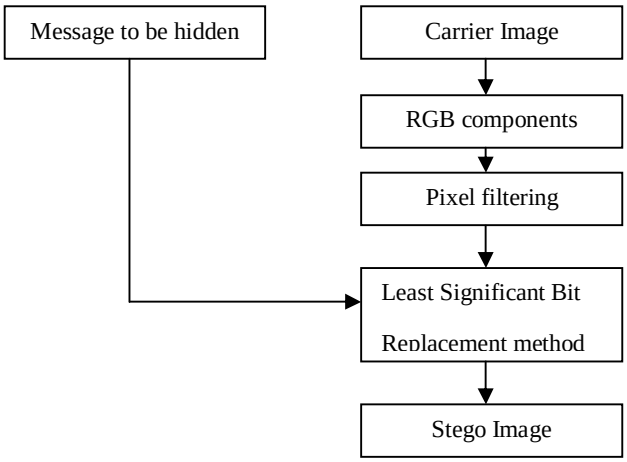

Figure 3: Least Significant Bit flowchart

3.3 Handshaking

Handshaking a client server TCP/IP protocol establishes connection between the two communication channels to enable them send messages to each other.

3.4 Steps for the Designed System

The whole process of hiding data inside picture using steganography and cryptography method is captured as follows:

❖ **Encryption module**

1. User prompted for username and password
2. Confirm whether the username and password is correct
3. Open the encryption module
4. Open picture
5. Input plain text (message)
6. Input password
7. Encrypt and encode message in picture
8. Save encrypted message
9. Send the encrypted message

❖ **Decryption module**

1. Receive encrypted message
2. Save encrypted message
3. Open encrypted module
4. Open picture carrying hidden text
5. Input password to decrypt the encrypted message
6. Confirm if the password is correct
7. Decode and decrypt message from picture
8. Read message.

Figure 4-10 presents the implemented interfaces of the designed system

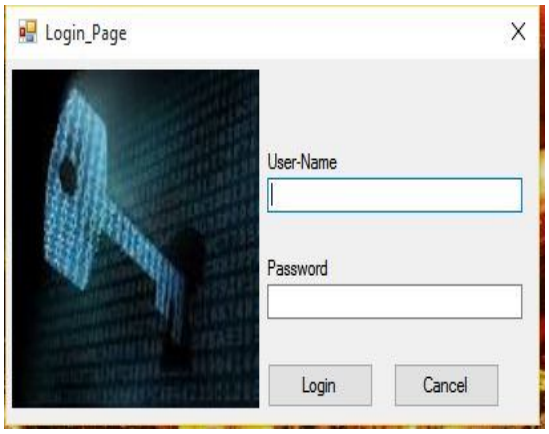

Figure 4: Login page

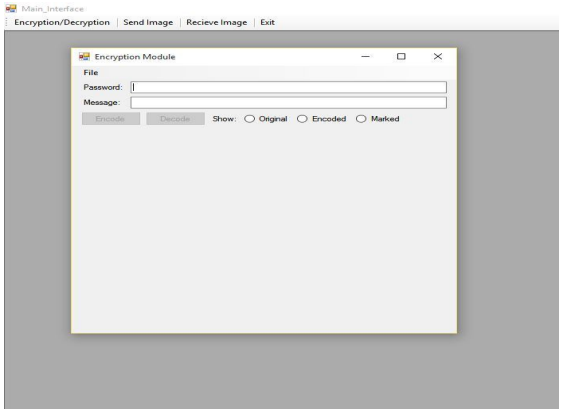

Figure 5: Encryption Module interface

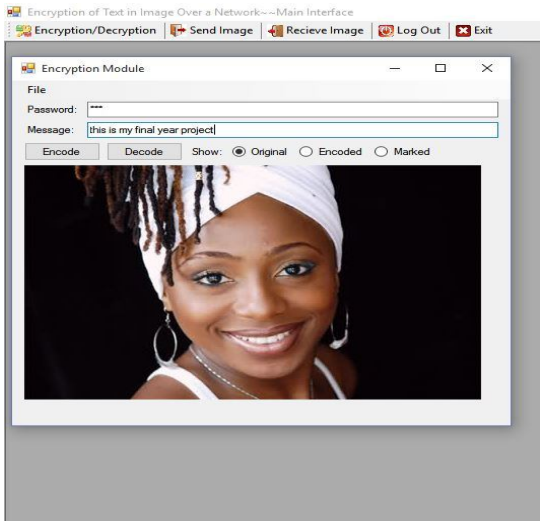

Figure 6: Text and image before encryption

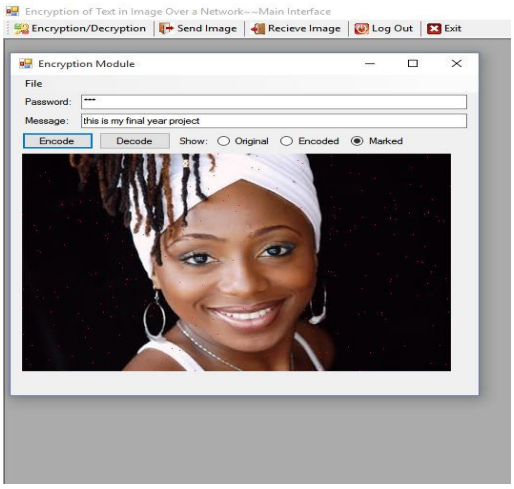

Figure 7: After Encryption

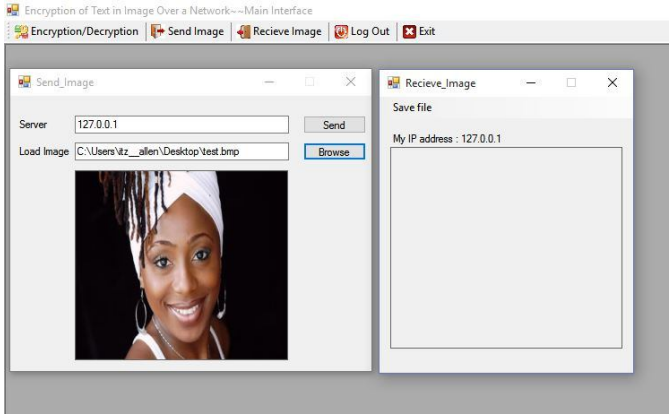

Figure 8: Message to be sent

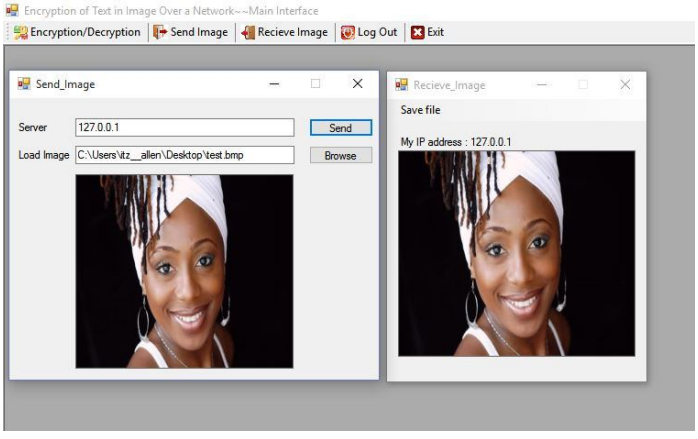

Figure 9: Sent message

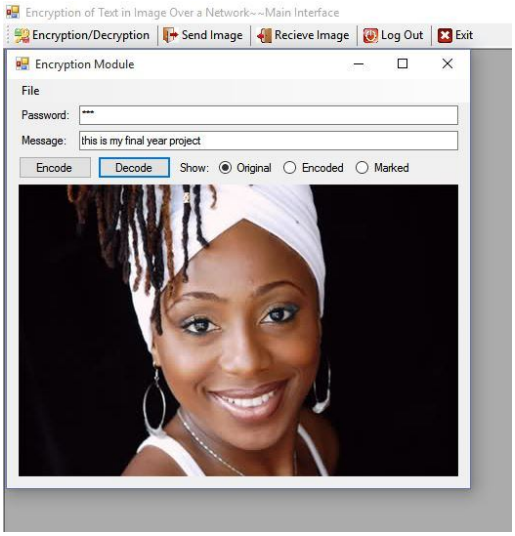

Figure 10: Decrypted message

4. CONCLUSION

This research has been able to hide text inside an image from sender A to recipient B over a network using blowfish algorithm for the encryption, Least Significant Bit method for the steganography and handshaking a client server TCP/IP protocol to establish connection between two communication medium. The system was implemented using socket programming (Visual C#.net). It is hereby recommended that other format of data could be included such as audio, PDF and so on. Also the research could be extended to include database so as to share the information offline.

REFERENCES

[1] Amin M. M., Salleh M., Ibrahim S., Katmin M. R., and Shamsuddin M. (2003), **"Information hiding using steganography,"** in Telecommunication Technology, 2003. NCTT 2003 Proceedings. 4th National Conference on, 2003, pp. 21-25.

[2] Abikoye Oluwakemi C, Adewole Kayode S and Oladipupo Ayotunde J. (2012). **"Efficient Data Hiding System using Cryptography and Steganography".** International Journal of Applied Information Systems (IJAIS) – ISSN: 2249-0868, Volume 4– No.11.

[3] www.webopedia.com/TERM/C/cryptography.html (accessed on: 4th may 2016).

[4] Mamta Juneja and Parvinder Sandhu. (2009). **"Implementation of Improved Steganographic Technique for 24-bit Bitmap Images in Communication".** Marsland Press Journal of American Science 2009:5(2) 36-42.

[5] Fahimeh Alizadeh, Nicolas Canceill, Sebastian Dabkiewicz and Diederik Vandevenne. (2012). **"using steganography to hide messages inside PDF files"**. SSN Project Report December 30, 2012

[6] I-Shi Lee and Wen-Hsiang Tsa (2010); **"A new approach to covert communication via PDF files".** Journal of Signal Processing, ISSN: 0165-1684, Vol (90):Issue (2) pp 557-565.

[7] Liu Hongmei, Li Lei, Jian Li, and Huang Jiwu (2007**). "Three novel algorithms for hiding data in pdf files based on incremental updates"**. Conference paper in International workshop on Digital Watermarking-Digital Forensic and watermarking Technical: part of the lecture notes in computer science book series (LNCS, volume 7128).

[8] Zhong Shangping, Cheng Xueqi, and Chen Tierui (2007)**," Data Hiding in a Kind of PDF Texts for Secret Communication"**. International Journal of Network Security, Vol.4, No.1, PP.17–26, Jan. 2007 17

[9] Krati Vyas and Pal.B.L. (2014). **"A proposed method in image steganography to improve image quality with LSB technique".** International Journal of Advanced Research in Computer and Communication Engineering Vol. 3, Issue 1, January 2014.

[10] Hebah H. O. Nasereddin and Murad Saleh Al Farzaeai. 2009**. "Proposed data hiding technique text image inside image (tiii)"**. Middle East University, Amman, Jordan, P.O. Box: 144378, Code 11814 Amman-Jordan, Amman Arab University for Graduate Studies.

[11] Pallavi H. Dixit, Kamalesh B. Waskar and Uttam L. Bombale. (2015). **"Multilevel Network Security Combining Cryptography and Steganography on ARM Platform".** Journal of Embedded Systems, vol. 3, no. 1 (2015): 11-15. doi: 10.12691/jes-3-1-2.

[12] Shashikala Channalli and Ajay Jadha. (2009). **"Steganography an Art of Hiding Data".** International Journal on Computer Science and Engineering, Vol.1(3), 2009, 137-141.

[13] Jyotika Kapur and Akshay J Baregar. (2013). **"Security using image processing".** International Journal of Managing Information Technology (IJMIT) Vol.5, No.2, May 2013.

[14] Dipti Kapoor Sarmah and Neha Bajpai (2010), **"Proposed System for Data Hiding using Cryptography and Steganography"**. Article in International Journal of Computer Applications · September 2010, vol(4) issue (5) pp7-10.

[15] Bruce Schneier (1993), "**Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish)**". Fast Software Encryption, Cambridge Security Workshop Proceedings (Springer-Verlag): 191–204.