International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 - 5161

Available at http://inpressco.com/category/ijcet

Research Article

Stegnalytic Method using Matching Keys

Savita I.Makond[Å*] and Manoj K.Nandani[Å]

[Å]E&C Dept.KLE CET Chikodi, India

Accepted 10 May 2014, Available online 01 June 2014, **Vol.4, No.3 (June 2014)**

Abstract

Steganography is the art of hiding information imperceptibly in a cover medium. The main aim in the steganography is to hide the very existence message in the covered medium. Today digital technology and internet provide for easy to use cover media for steganography. In order to improve the security for data by providing stego Image. The proposed method describes two steps for hiding the secret information by using the steganography based on matching method. The first step is to find the shared stego-key between the two communication parties over insecure networks by applying Diffie Hellman Key exchange protocol. The second step in the proposed method is that the sender uses the secret stego-key to select pixels that it will be used to hide. Each selected pixel is then used to hide 8 bits binary information depending on the matching method.

Keywords*: Diffie Hellman, LSB,DataHiding, CoverImage, StegoImage , Encryption, Decryption.*

1. Introduction

Steganography is the art and science of hiding communications, a steganographic system thus embeds hidden content in unremarkable cover media so as not to arouse a eavesdropper's suspicion. In the past, people used hidden tattoos or invisible ink to convey steganographic content. Today, computer and network technologies provide-easy-to-use communication channels for steganography. Essentially, the information-hiding process in a steganographic system starts by identifying a cover medium's redundant bits with data from hidden message (Neils P and Peter H *et al* 2003).Modern steganography's goal is to keep the presence of the message undetectable form an unauthorized access.

Steganography applications that hide data in images generally use a variation of least significant bit(LSB)embedding (R.Chandramouli N.Memon *et al* 2001). In LSB embedding the data is hidden in the least significant bit of each byte in the image. The size of each pixel depends on the format of the image and normally ranges from 1 byte to 3 bytes. Each unique numerical pixel value corresponds to a colour: thus an 8-bit pixel is capable of displaying 256 different colours (J.D.Foley *et al* 1996) . Given two identical images, if the least significant bits of the pixels in one image are chnaged,then the two images still look identical to the human eye (Diffie, W and , Hellman, M. E. *et al* ,1976) .This is because human eye is not sensitive enough to notice the difference in colour between pixels that are different by 1 unit (R.Chandramouli N.Memon *et al* 2001). Thus, steganography applications use LSB emedding because attackers do not notice anything odd or suspicious about an image if its pixel's least significant bits are modified(R.Chandramouli N.Memon *et al* 2001).

In 1976 Diffie and Hellman (Diffie W and Hellman, M. E. *et al* ,1976) introduced the first concept of public-key cryptography to solve the key exchange problem. Public-key cryptography is one of the greatest contributions in the history of cryptography. Nowadays, Public-key cryptography is practically utilized in everyday life to attain privacy, authenticity, integrity and non-repudiation (W.Stallings *et al* 2006; K.Y. Chen *et al* 2004; C.S. Laih, and K.Y. Chen *et al* 2004).One of the main branches and applications of the public-key cryptography is a public-key encryption scheme which allows two parties to communicate securely over an insecure channel without having prior knowledge of each other to establish a shared secret key.

2. Methodology

The proposed method describes two steps for hiding the secret information by using the public steganography based on matching method.

Step 1: The first step is to find the shared stego-key between the two communication parties (Alice and Bob) over insecure networks by applying Diffie Hellman Key exchange protocol. As shown by Fig, DH key exchange protocol shows the technique for the key exchange between two parts (Alice and Bob) to get shared Stego-key values. Alice must generate the keys (public and private keys) and use her private keys to give new public key and send it to Bob's side. Bob must obtain and issue new public keys. Then at the end the protocol, each side recovers his/her received public key to reach the shared

*Corresponding author: **Savita I.Makond**

values between them, that's mean Alice and Bob have arrived same sego-key value.

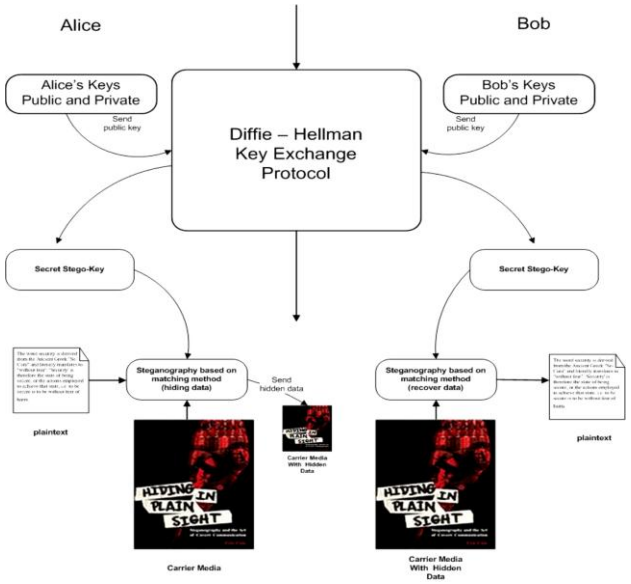


Figure 1 The public key steganography protocol

Step 2: Matching Method

The second step in this technique is that the sender uses the secret stego-key to select pixels that it will be used to hide. Each selected pixel is then used to hide 8 bits binary information depending on the matching method which is summarized in four cases as shown by Table .1. Since the 8 bits data will be compared with the selected pixel's bytes, red, green and blue values respectively to produce an array with 2 bit binary values as 00, 01, 10, and 11. Alice's side, starts comparing to search the equality, where, she takes data value and compare it with the value of the red colour (± 7 – decimal value). As shown by Table 2.6.1, Case no. 1, if they are equal, then the value zero (00 – binary value) is set to the array. Case no. 2, if the data value and the red value are not equivalent then the value will be compared with the green colour, if they are equals (± 7 – decimal value) then the array is set to be one (01- binary value). Case no. 3, if the data value and the green value are not equivalent then the value will be compared with the blue colour, if they are equals (± 7 – decimal value) then the value two (10 – binary value) is set to the array (refer to Figure 2.6). Finally Case no. 4, If in case the secret data didn't equal any of the previous three conditions then the LSBs method is used to embed the data inside the selected pixel, and the value three (11 – binary value) is set to the array. In this case, the data value will be distributed as follows:

- The first three bits of the data are replaced by the three least significant bits of the red byte.
- The second three data bits are replaced by the three least significant bits of the green byte.
- The last two data bits are replaced by the two least significant bits of the blue byte.

Table 1 The four main cases in the proposed public-key stego

Case no.1	If 8 bit data= red(8 bit)	Then red value= 8 bit data
Case no.2	If 8 bit data= green(8 bit)	Then green value= 8 bit data
Case no.3	If 8 bit data= blue(8 bit)	Then blue value= 8 bit data
Case no.4	Otherwise	Use LSB method

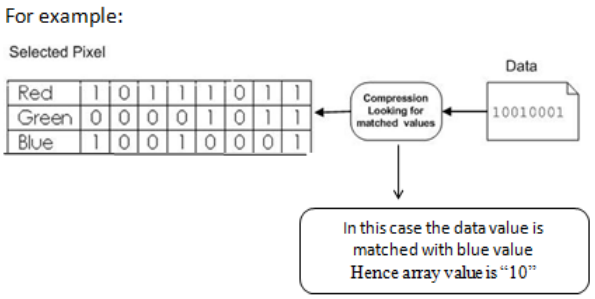


Figure 2 A working example in the proposed public-key stego

3. Data Flow Diagrams and Flowcharts

A. Data Flow Diagrams

Data flow diagrams are the basic building blocks that define the flow of data in a system to the particular destination and difference in the flow when any transformation happens. It makes whole procedure like a good document and makes simpler and easy to understand for both programmers and non-programmers by dividing into the sub process. The data flow diagrams are the simple blocks that reveals between various components of the system and provide high level overview, boundaries of particular system as well as provide detailed overview of system elements. The data flow diagrams start from source and ends at the destination level. i.e, it decomposes from high level to lower level .The important things to remember about data flow diagrams are: it indicates the data flow for one way but not for loop structures and it doesn't indicate the time factors.

i.) Level 0 Data Flow Diagram

'DFD level 0' is the highest level view of the system, contains only one process which represents whole function of the system.

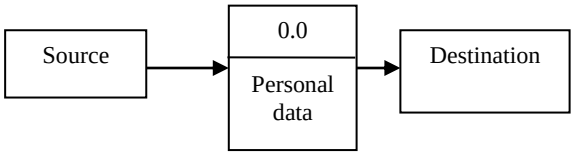


Figure 3 Level 0 Data Flow Diagram

It doesn't contain any data stores and the data is stored within the process. For constructing DFD level 0 diagram for this approach we need two sources one is for 'source' and another is for 'destination' and a 'process'. DFD level 0 is the basic data flow process, the main objective is to transfer the data from sender to receiver after encryption.

ii.) Level 1 Data Flow Diagram

In this data flow diagram, the secret data is sent to the encryption phase for embedding the data into the image for generating the carrier image. In the next phase the carrier image is sent to the decryption phase through the transmission phase. The final phase is the decryption phase where the data is extracted from the image and displays the original message.

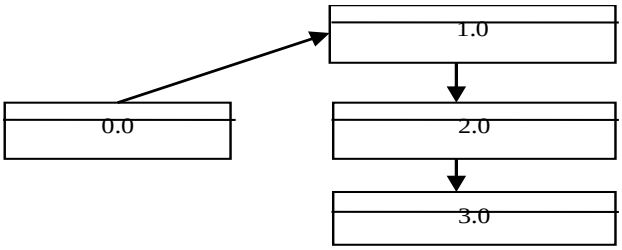


Figure 4 Level 1 Data Flow Diagram

iii.) Level 2 Data Flow Diagram

The image and the text document are given to the encryption phase. The encryption algorithm is used for embedding the data into the image. The resultant image acting as a carrier image is transmitted to the decryption phase using the transmission medium. For extracting the message from the carrier image, it is sent to the decryption section. The plain text is extracted from the carrier image using the decryption algorithm.

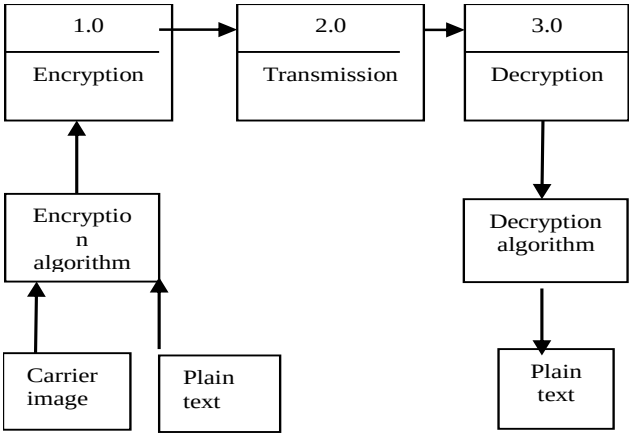
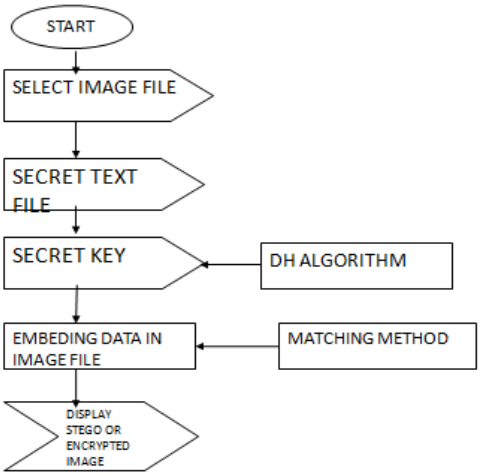


Figure 5 Level 2 Data Flow Diagram

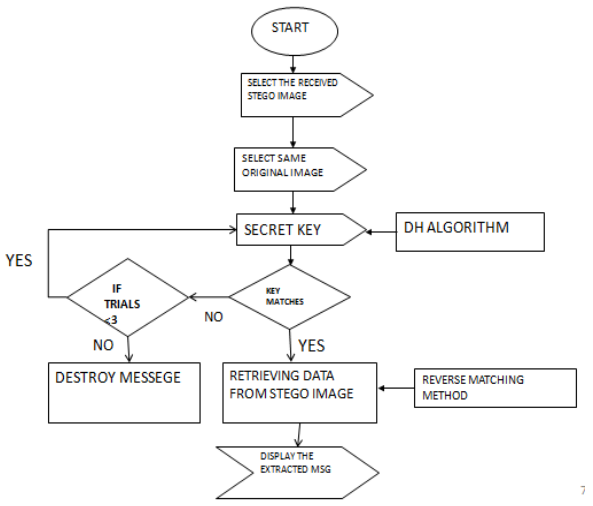
B. Activity Diagram (Flowcharts)

The sender sends the message to the receiver using three phases. Since we are using the steganographic approach for transferring the message to the destination, the sender sends text as well as image file to the primary phase i.e., to encryption phase. The encryption phase uses the encryption algorithm by which the carrier image is generated. The encryption phase generates the carrier image as output. The carrier image is given as input to the next phase i.e., to decryption phase. The decryption phase uses the decryption algorithm for decrypting the original text from the image so that the decryption phases generate plain text. The plain text is then sent to the receiver.

i) Encoding phase



ii) Decoding Phase

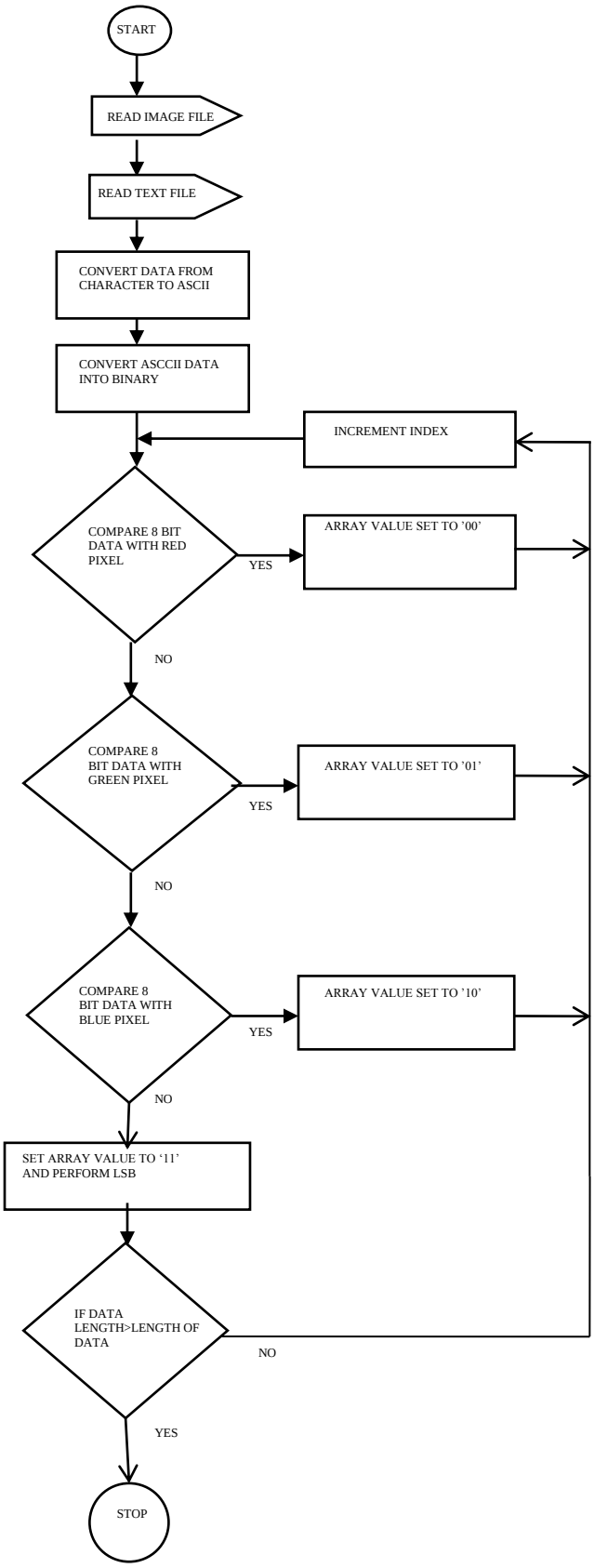


iii) Matching Method

It is shown in flow chart on page 1448

4. Results and Discussions

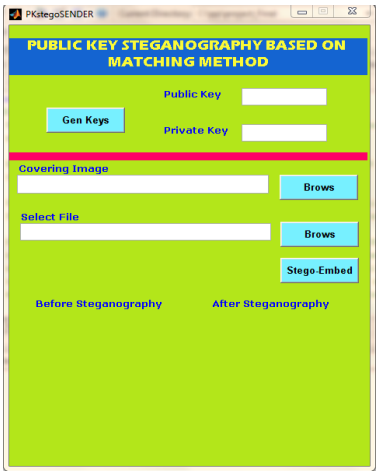
For designing the steganographic application, it consist of different phases like encryption, decryption and data transmission. An application for sending the personal data securely to the destination has been developed successfully. The design phase is the primary phase, which gives a brief idea about the different levels used for



developing an application with the help of block diagram. The most important phase in the paper is the execution phase. The execution phase is developed with the help of design phase. For executing the application, we have worked on two sections: one is encryption and another is decryption. In this paper we mainly concentrated on embedding the data into the image. We have designed the Steganography application which embedded the data into the image.

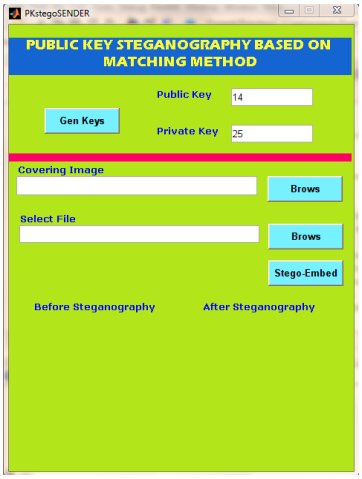
Normally, after embedding the data into the image, the image may lose its resolution. In this approach, the image remains unchanged in its resolution as well in size. The speed of embedding the data into the image is also high in this method such that the image is protected and the data to the destination is sent securely. For the decryption phase, for the purpose of designing. We have used security keys like personal password for protecting the image from unauthorized modifications, which improved the security level. We have choosen image stegenography because it is simple to use its user friendly application.

A. GUI for Sender



This is the gui for sender, it has 3 buttons Gen keys, Brows and Stego-Embed which are used to get input from user. Gen keys button will generate two types of keys they are public and private keys. Brows button is used to select cover image. Another Brows botton is used to select text file. Stego-embed button is used to embed the text file in selected image.

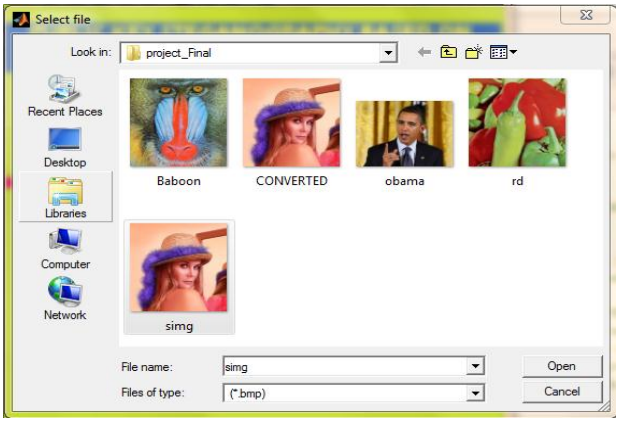
i) Generating key

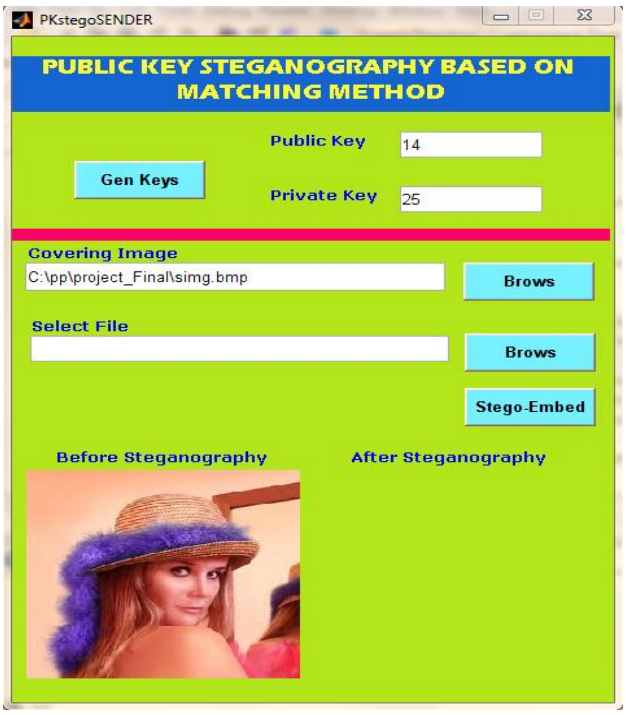


When Gen Key button is clicked, execution pointer will jump to Deffie Hellman procedure, this function will

generate a set of keys, they are private key and public key, and they are displayed back on GUI, User can also enter keys manually or can also select from generated keys.

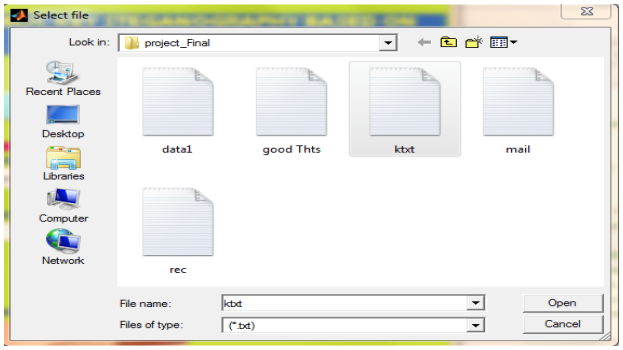
ii) Selecting image file

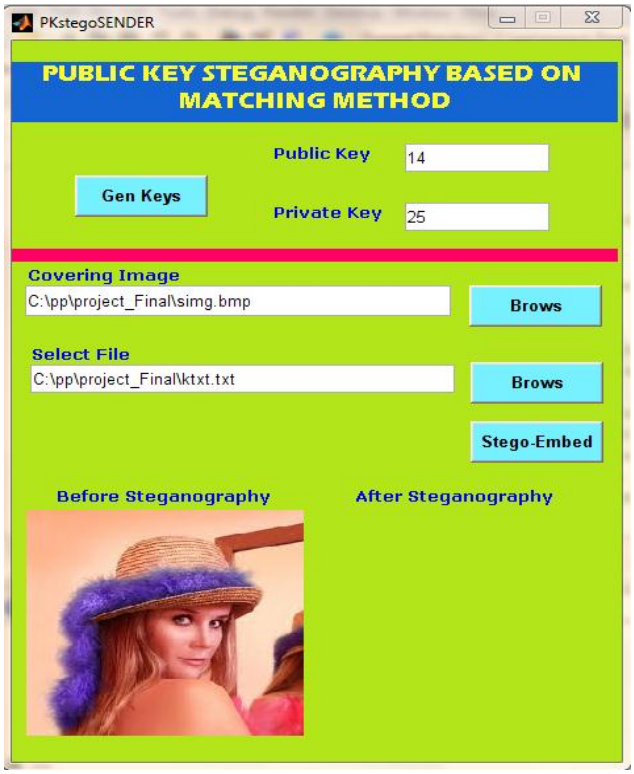




All image files stored in specified locations will display after clicking on Brows button of the GUI, user can select any image file to cover his secret message.

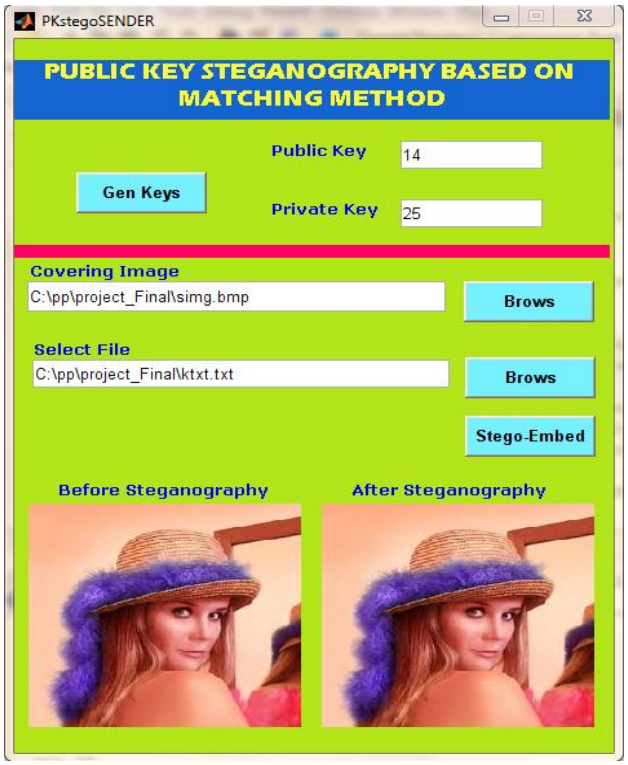
iii) Selecting text file





All text files stored in a specified locations will display after clicking Browse button of GUI, user can select any text file which he wants to hide in image file.

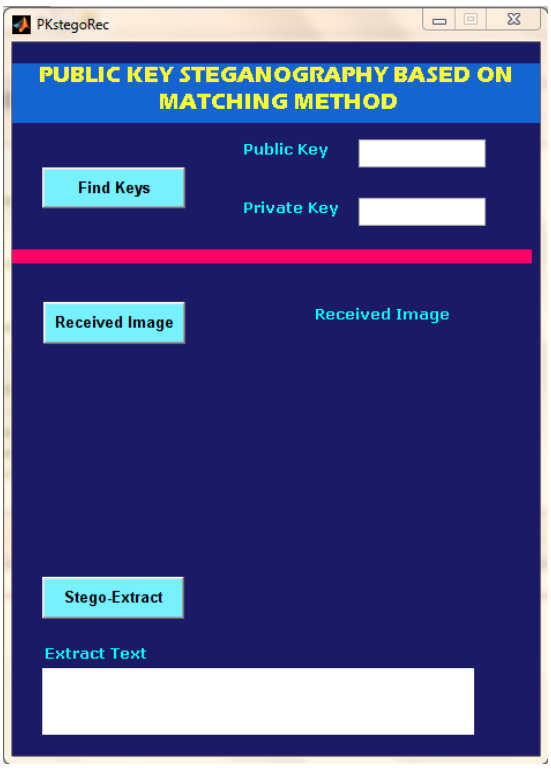
iv) Output of encoding



After clicking on stego-embed button, pointer moves to matching method procedure, after executing that function, a resulted stego image will display on the axis of GUI. Based on quality of embedded image user can send it to

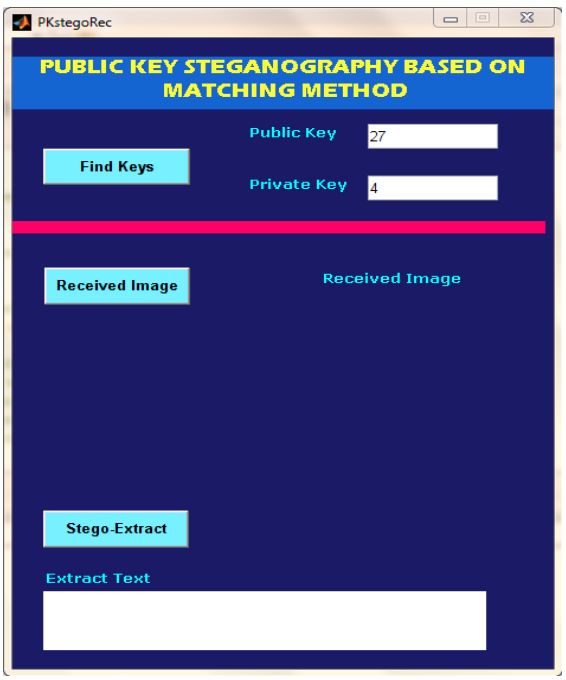
receiver or he may discard it and try for another cover image which may have more quality.

B. GUI for Receiver



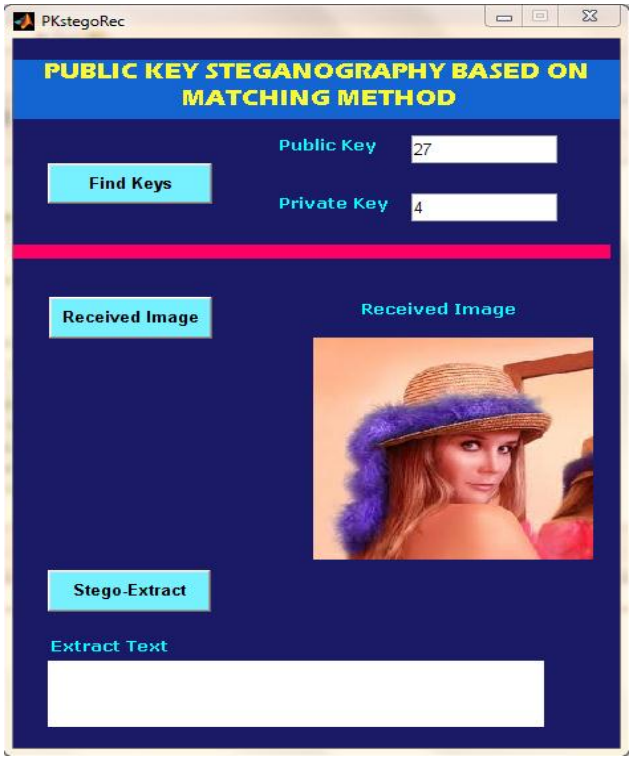
This is the GUI for receiver, it also has 3 buttons, Find Keys, Received Image, & Stego-Extract. Find keys button will display the public and private keys, Recieved image button will display the received stego-image, and Stego-Extract button will display the information that is extracted from the stego-image.

i) Finding keys

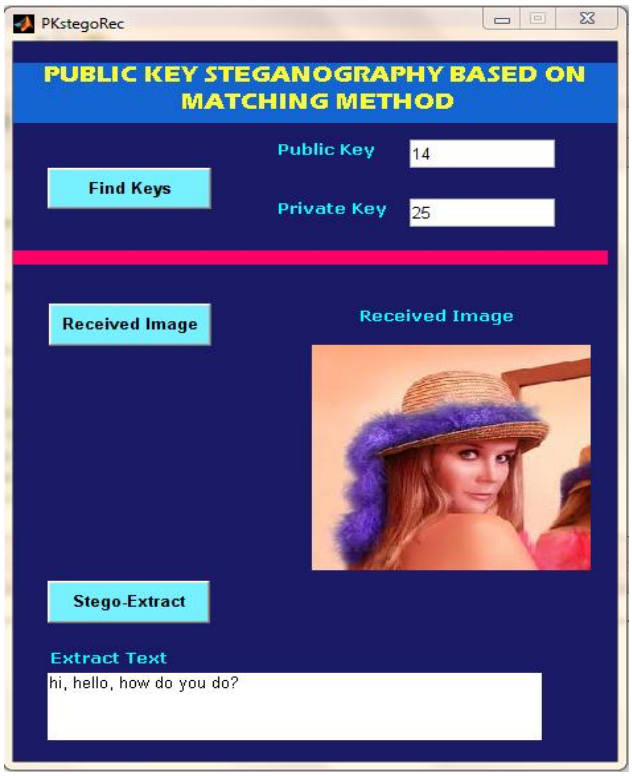


A set of private key and public key will display after clicking on Find Keys button, user can manually enter the keys or can also select from generated sets.

ii) Select received image

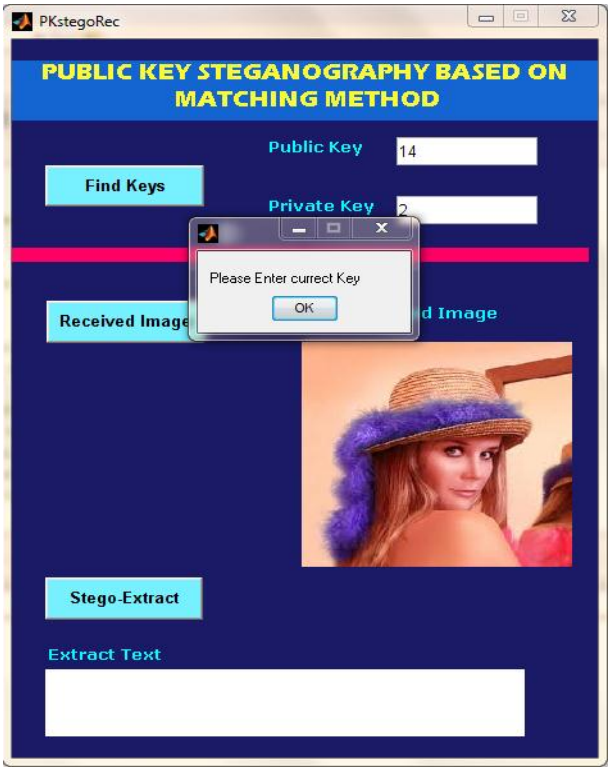


iii) Output after decoding

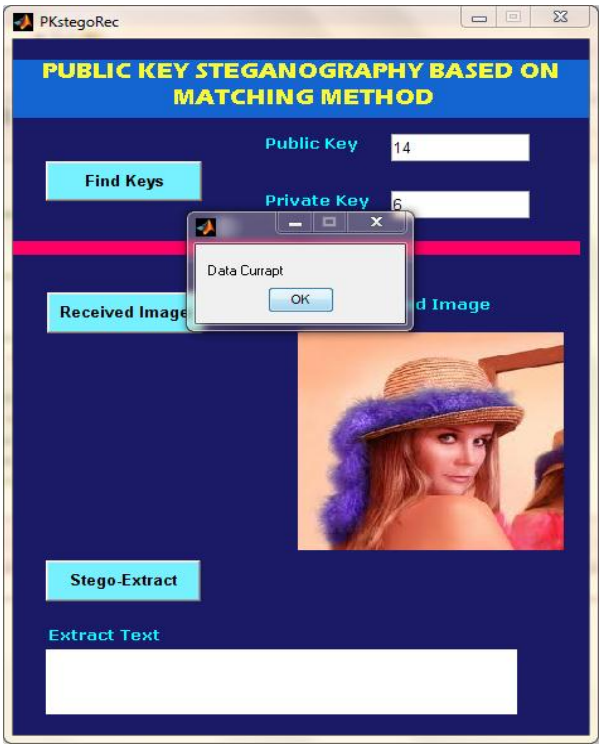


When receiver enters a key for extraction of secret data, tool will check entered key with sender secret key. If key

is wrong it will show a message box which contains a message please enter correct key.



If a hacker tries to hack the information by trial and error method, tool will show a message box containing message data corrupt after 3 trials.



Conclusion

Steganography has its place in security. Though it cannot replace cryptography totally, it is intended to supplement it. Steganography can be used along with cryptography to make a highly secure data . Formerly just an interest of the military, Steganography is now gaining popularity among the masses. Since we use least significant bits of pixel values to store data, the data may lost if any compression techniques are applied to stego-image, but after embedding the stego-image remains unchanged in its resolution as well in size. Tool worked fine with all type of images like .jpg, bmp, giff, tiff etc. In this paper it is shown the possibility of using matching method in public steganographic protocol. It produces matching between the data bit parts and the selected pixels. 3 pixels were needed to hide 8 bit of data in case of normal LSB method but in this method, only one pixel is enough to hide 8 bit of data. So by using matching method size of the data that can be hidden in a pixel is increased by 3 times more than normal LSB method. The security of the in this steganography depends on Deffie Hellman public key exchange protocol. Hackers may use trial and error method to match the keys, so tool is designed such that it should corrupt the secret data after three trials.

Future Work

Future work includes, embedding secret message in other type of files like audio, video. Further work could include developing a YASS(yet another steganographic scheme) and strong encryption algorithms like AES or DES. GUI can be refined and made more user friendly. Also a command line version can be developed for this application so that it will suit the more experienced users.

References

Neils, P. and Peter, H 2003, Hide and Seek: An Introduction to Steganography, *IEEE Computer Society IEEE Security and Privacy*, pp. 32-44.

R.Chandramouli N.Memon Analysis of LSB Based image steganography techniques, Image processing. Vol 3.pp 1019-1022, October 2001.

J.D.Foley, Computer Graphics: Principles and Practices, Cornell: Addition,wessley, 1996,pp.3

Diffie, W. and , Hellman, M. E., (1976) New Directions in Cryptography, *IEEE Transactions on Information Theory*, IT-22, pp. 644-654.

W.Stallings, Cryptography and Network Security- Principles and Practices,Prantice Hall,Inc,,th ED,(2006)

K.Y. Chen, The study and Implementations of Certificates in PKI, Ph D. Thesis ,Department of Electrical Enginnering, National Cheng Kung University,Taiwan,(2004)

C.-S. Laih, and K.Y. Chen,Generating Visible RSA public keys for PKI, *International Journal of InformationSecurity,Vol.2,No.2,Springerverlag,Berrlin*,(2004)

Nameer N. EL-Emam Hiding a large amount of data with high security using steganography algorithm, *Journal of Computer Science*. April 2007, Page(s): 223 – 23

Johnson, N. F. and Jajodia, S., (1998) Exploring Steganography: Seeing the Unseen, *IEEE Computer*, pp. 26-34.pp 103-109

Cryptography And Network Security book by Behrouz A. Forouzan and Debdeep Mukhopadhyay

Bender, W., Gruhl, D., Morimoto, N., and Lu, A., (1996) Techniques for Data Hiding, *IBM Systems Journal*, Vol 35, Nos 3&4, pp. 313-336.

Cox ,I. J., Kilian, J., Leighton, T. , and Shamoon, T., (1996) A Secure, Robust Watermark for Multimedia, Proc. First Int'l Workshop Information Hiding, Lecture Notes in Computer Science No. 1, 174, Springer- Verlag, Berlin, pp. 185-206.