Advanced Computing: An International Journal (ACIJ), Vol.3, No.3, May 2012

Data Security by Preprocessing the Text with Secret Hiding

Ajit Singh[1] and Upasana Jauhari[2]

Department of Computer Science & Engineering and Information Technology

BPS Mahila Vishwavidyalaya, Khanpur Kalan, Sonepat-131305 Haryana (India).

[1]ghanghas_ajit@rediffmail.com

[2]upasanajohari2006@gmail.com

ABSTRACT

With the advent of the Internet, an open forum, the massive increase in the data travel across network make an issue for secure transmission. Cryptography is the term that involves many encryption method to make data secure. But the transmission of the secure data is an intricate task. Steganography here comes with effect of transmission without revealing the secure data. The research paper provide the mechanism which enhance the security of data by using a crypto+stegano combination to increase the security level without knowing the fact that some secret data is sharing across networks. In the first phase data is encrypted by manipulating the text using the ASCII codes and some random generated strings for the codes by taking some parameters. Steganography related to cryptography forms the basis for many data hiding techniques. The data is encrypted using a proposed approach and then hide the message in random N images with the help of perfect hashing scheme which increase the security of the message before sending across the medium. Thus the sending and receiving of message will be safe and secure with an increased confidentiality.

KEYWORDS*--- Data encryption using ASCII codes, Perfect hashing, Steganography, secret hiding.*

1 .INTRODUCTION

Data communication is a field where the security issues have its own priority. While exchanging data electronically, the privacy and secrecy of data is primarily concern. Here encryption, transforming data (plain text) into cipher text and decryption, a reverse process, plays a vital role in concealing the confidentiality of the data. The data (message) before exchange or transmission is encrypted with a secret key [1] which provide another level of secure communication between the sender and receiver.

The approach is recommended since it combines the benefits of hiding the existence of a secret message with the security of encryption. In such keyed steganographic process a secret message is first encrypted, then embedded into the cover data using available steganographic techniques. Once received, the encrypted secret message is extracted, and then the message is decrypted with the appropriate key and algorithm. A general steganographic mechanism is shown as below:

DOI : 10.5121/acij.2012.3307

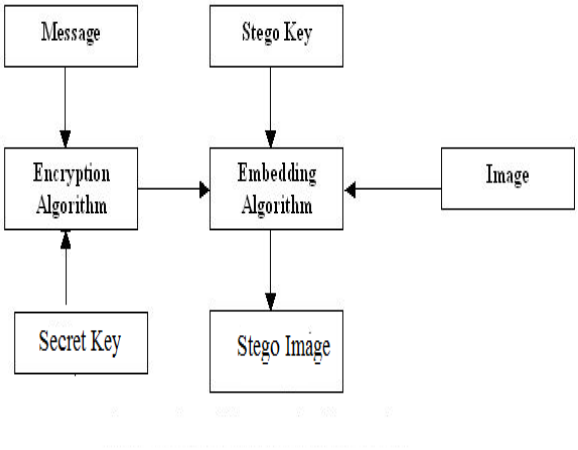

Fig 1 Steganography Process

Recently many steganographic schemes have been devised. Many researchers likewise attempting to detect the secret data that hide inside the different Medias presented to us. Without compromising this fact a solution or remedy has been proposed named secret hiding. In secret hiding, there are N images send by the sender and allows a recipient party to extract the message when it has images as the secret message is hidden randomly in the images. As many such schemes [14] have been developed but they require computational overhead.

We propose a scheme in which the message is hidden randomly in any of N images. To avoid much computational overhead like EX-OR operations for N images, among N images the stego images are especially marked with the symbol which is embed in LSB (Least Significant Bit) of the value of pixels so that the intended recipient need to extract the message from the unmarked images by checking the only the pixel LSB. Here the files need to be hide using perfect hashing algorithm[3,17]—is the one that contains the encrypted string of input text and second contains the random generated strings along with the secret key is sent through a secure channel, both the files- encrypted string file and random generated string file are produce during encryption process. Thus the receiving party need these three things to get the original message back. Thus this whole process provides better security.

The rest of the paper is divided as section 2 describe the encryption process on which the security of message based, section 3 is the description of perfect hashing scheme which forms the basis of hiding and LSB method for embedding the mark (symbol, which is `&` sign) in the images , section 4 is the description of proposed work, section 5 is about implementation and section 6 is the conclusion part.

2. DATA ENCRYPTION USING ASCII CODES

Data encryption involves the transformation of original data into some encoded form using a secret key [1]. Data encryption generally refers to some mathematical calculations and/or algorithmic strategies that transform the data into a form that is inaccessible by any unauthorized person/attackers. Many algorithms have been devised with the passage of time which is used enormously throughout the internet for security. On a whole data encryption is just the mechanism for making the information unintelligible by making it unreadable without knowing the strategies being applied to it. The basic operation is applied on the ASCII codes

which is generated and stored in file. The ASCII character encoding or a compatible extension is used on nearly all common computers, especially personal computers and workstations. ASCII codes [2] include definitions for 128 characters: 33 are non-printing, 94 are printable characters (the space is not printable). The representation of each and every character is with the seven bits (b7 to b1) e.g. the representation of A is (1000001) =65, similarly other characters are coded like this.
The encryption process applied on the input text is shown below:

Encryption Process:

During first phase of the algorithm

1. Generating the random strings and saving it into text file (in this case, creating 1000 random string)
2. Input the Text for the encryption.
3. Generating the key for the text which need different arrays to store the required data
   a) Getting all the lines from the entered text and inserting them into the string array arrLines.
   b) Lines will be separated by the "." operator.
   c) From the arrLines array, define an arrWord array that will be filled by all the words of the specific lines.
4. By working on specific lines and on all the words of the lines, start the encryption process.
   a) Getting the character of the word one by one for the processing.
   b) Create an array arrSeq and Filling the array that have the range for all the character(symbols, integers, alphabets)
   c) Convert the incoming character into its ASCII value.
   d) Find the range corresponding to the incoming char. Ascii value by using arrSeq array.
   e) By using the range, generate a random integer between that range and fetch thats no's position random string from the random string txt file. Ex – for the asciii value 65, 651 – 660 is the range. Let 653 is the random no. generated, then find the random string placed on the 653th place in the random string text file.
5. After getting the random ascii string of that character, add four parameter in that.
   Four parameters are:
   i. linoNo (line number from which that word relates),
   ii. posOfW (position of the word in that line),

iii. lengthOfW (length of that word) and

iv. posInWo (position of character in that word).

The output string contain the random string appended with these four parameters.

In 2nd phase, apply some mathematical operation in that encoded string.

1. Now key comes into play, store the entire individual integer in the arrKey array and all the individual ascii value into the arrCodes array.

2. In this case here using 5 length long integer value and performing respective mathematical operation on each value:

a. Ascii + first integer

b. Then multiplied by second integer

c. Then Ascii – third integer * 20

d. Then Ascii- fourth integer*5

e. Then Ascii-fifth integer

3. Convert into the symbol whose value is between 0 and 255 after applying operation and leave remain as it is.

4. The data retrieved after this is in corresponding encrypted form which needs to be store in separate file.

Decryption Process:

Following is the decryption process which involves the use of same key as generated during encryption process.

1. Using the key of length 5 (here used), apply the following mathematical operations which are performed.
   a. Ascii - fifth integer
   b. Then Ascii +fourth integer*5
   c. Then Ascii + third integer * 20
   d. Then Ascii- second integer

e. Then Ascii- first integer

2. Inserting the changed value of code in array

3. Inserting the decrypted code value

4. The random string generated is shown by considering and processing the four parameters taken.

5. From the random string generated, the particular character is retrieve

6. The outputted text is the original message.

3. General approaches used in proposed scheme

3.1LSB Technique

This technique has been introduced in several papers [7, 9, 11]. So staying on the subject, here present the procedure which is carried out for embedding the symbol using the LSB of pixels values. A 1,024 X 768 image has the potential to hide a total of 2,359,296 bits (294,912 bytes) of information. . Each pixel value contains the value of the color and is represented in bits (0 & 1).Similarly, text is also represented in bits (0&1).Therefore comparing bit values byte by byte result in hiding the bit values of Text in bit value of an Image. i.e. storing in LSB of a byte (pixel). For example, the letter A can be hidden in three pixels (assuming no compression). The original raster data for 3 pixels (9 bytes) may be:
(00100111 11101001 11001000)

(00100111 11001000 11101001)

(11001000 00100111 11101001)

The binary value for A is 10000011. Inserting the binary value for A in the three pixels would result in

(00100111 11101000 11001000)

(00100110 11001000 11101000)

(11001000 00100111 11101001)

The bold color bits are the only three actually changed in the 8 bytes used. On average, LSB requires that only half the bits in an image be changed. So you can hide data in the least and second least significant bits and still the human eye would not be able to discern it.

3.2 Perfect Hashing based approach

Here, the used perfect hashing based algorithm was originally presented in [17] and now is explained with respect to its usage with grey-scale images. However for data transmission on internet, the file formats such as Jpeg/jpeg and gif are popular due to their small size. Following is the algorithm which defines both hiding and extracting of data.

A. Hiding Data

Following steps of the algorithm were used to hide the target data/information in an image**.**

1. **–** Input a file containing encrypted string and input (.jpg, .gif, .bmp or .tiff) image.

2. **–** Read the file, tokenize the values by making chunks of the values of 3 characters each and storing them in an array-list (*la*). Total count of data chunks are represented as *n*.

3. – Generate a random number that is used as a hash key and hash-key is represented using *h*.

4. – The hash-function (H) [20] uses the hash-key (*h*) and total number of chunks (*n*) to generate a pattern i.e. sequence of numbers (hash-values) which represents the position of the pixels where data will be stored.

5. – The generated pattern (containing sequence of numbers) is stored in an array-list (*ls*).

6. – First chunk from *la* and *ls* are read. String stored in *lc* is read and tokenized. The ASCII value of each token *la[i]* is replaced with the *i* byte of the *ls[i]*.

This process is repeated until the last token of the encrypted text is stored.

7. – The output is the image containing stored data and a hash-key (*h*) that is used to retrieve data

Note: The random generated string file here is hiding using a tool hip21 [19] used to hide data which is password protected as to preserve the length of each random string and the key will send through a secure medium.

B. Extracting Data

Following steps of the algorithm were used to retrieve the hidden data/information from the image**.**

1. **–** Input the (.jpg, .gif, .bmp or .tiff) image that contains that stored information and the hash-key (*h*) that was actually used to store data.

2. – The input hash-key (*h*) is used with the hash function (H) for generating the sequence of numbers as an array-list (*ls*) and these numbers are actually position of the pixels where data was be stored.

The hash-function (H) here generates specifically same pattern of random numbers for a hash-key (H) those were generated at the time of encoding.

3. – Each value from the generated patterns represent index of a pixel where the data is saved. Values of the grey color byte at *ls[i]* are read. As each byte contains an ACSII value of a character, the read ASCII value is converted to a character and each character is written to a text file in sequence it is read from the image.

4. – The output is a text file that contains the retrieved data from image which now need to decrypt.

C.Perfect Hash Function Generator

A function for perfect hashing defined for a set S is a hash function that maps distinct N elements in S to a set of integers, with no collisions. A perfect hash function supports efficient lookups by placing hash-keys from N to a hash-table [17]. A few implementations for perfect hash functions are available. Here used GNU implementation of hash function called 'gperf' [18] that typically generates perfect hash- functions for a hash-key. A 'gperf' based hash function locates only one position in a domain using exactly 1 probe.

6. PROPOSED TECHNIQUE

In this proposed scheme, from the N cover images, randomly images are chosen for secret hiding. Similar approaches have used in [14]. As the message now is in encrypted form produced during encryption process describe in section 2 which makes itself difficult to decrypt unless one has both the random generated strings and the secret key. Before hiding the encrypted string and the random generated strings produced, apart from chosen images, the rest of the cover images are embedded with a mark (symbol) [9] at various pixels position to differentiate them from the stego image thus making the third party a wrong illusion of some secret data is hidden. Moreover the recipient party by checking this mark, will aware of non existing of any secret information in the image. For embedding, LSB (most commonly used) technique is used due to its simplicity and easy implementation. The change in the LSB of the pixel in image render the visual quality of image so on hiding the message does not much affect image and make unnoticeable to unintended person.

4.1 IMAGE STEGANOGRAPHY

As said that steganography [5,6,13] is all about hiding the data in any cover medium and then transmit it. The fact lies here is that cryptography is about making data in unreadable form while steganography is meant for hiding the data itself for more secrecy. Nowadays the most common used media is images. The digital images are widely used and accepted by steganography because of their practical usage across the communication. Moreover a high visual quality of image does not reveal the fact that some secret message is hiding into it. In this paper, for hiding the message produce after encryption method proposed here, the message is embed into the image [8,9,12] as the cover medium and the result produce a stego image which then transmit over the channel.

In this section, the gray-scale images are used as the cover medium [7] and the procedure based on perfect hashing described in section 3 is explained.

Following is the major steps involved in data hiding and extraction:

a) Input the cover images as the target image that is not marked.
b) Input the file containing the encrypted strings produce during encryption process.
c) Hide data from the file containing encrypted strings into one of the chosen image.
d) Similarly hide the data from the file containing random generated strings into another chosen image using the tool.
e) Retrieve back the data from both the images
f) Retrieved one of the data is in encrypted form.
g) Decrypt the message with the help of secret key produced during encryption.
h) After decryption, the original text will get retrieve.

A. Hiding Process

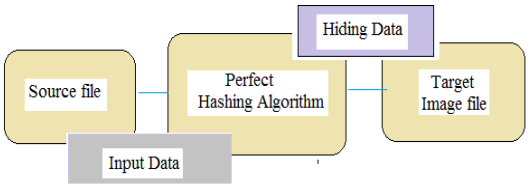

Figure 2 Hiding Information Process

B. Extracting Process

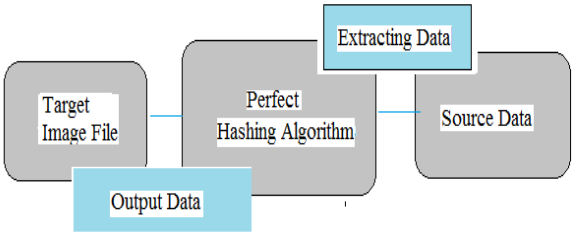

Figure 3 Retrieving Information Process

6. IMPLEMENTATION

The encryption and decryption process has been implemented in C#.NET and the result shown is the encryption and decryption of data. Here with experiment the input has taken as the word "banastahli", corresponding key has been generated and so the encoded string. During decryption a reverse process has been applied to get the original data.

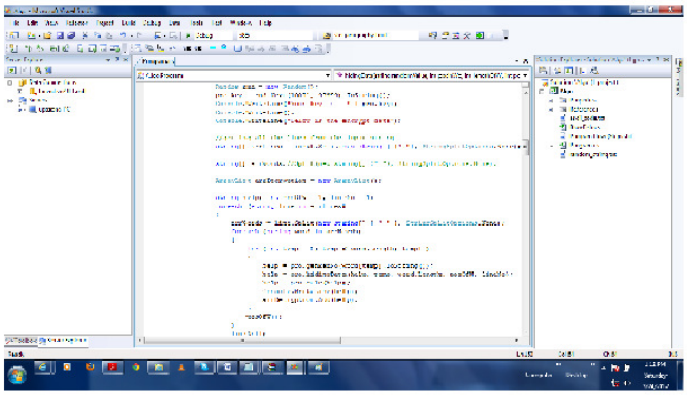

Figure 4 Coding in C#.NET

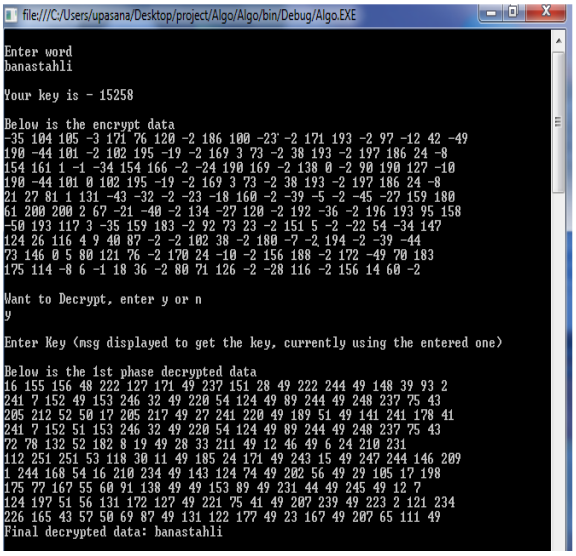

Figure 5 Results during Encryption & Decryption Process.

The prototype data hiding is experimented by utilizing a tool and result is shown:

Figure 7: Perfect Hashing based Hiding Process

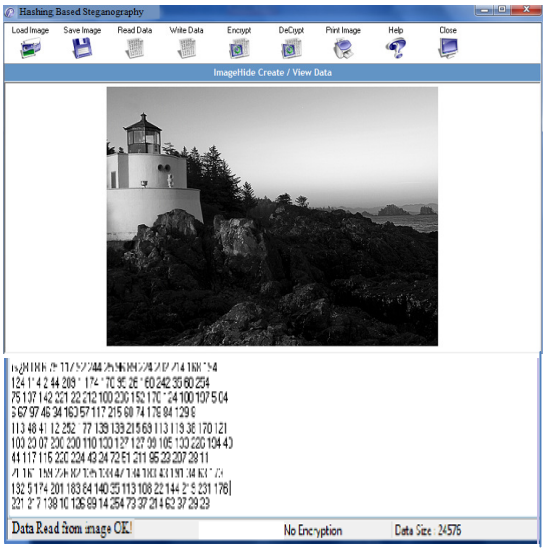

Figure 8 Perfect Hashing based Extracting Process

6. CONCLUSION

A novel approach have mentioned where preprocessing of message has been done by manipulating the ASCII values using various parameters which involve the process of encryption which is an add-on for simple data hiding. Random images are chosen and used for hiding data which are differentiated with image carry secret data using the mark. A perfect hashing based scheme is utilized to avoid the basic problems like collision in hashing

technique. The whole procedure is meant to provide better secure transmission of message across medium as the encryption process itself proves another layer of security. However, the used algorithms can be improved to get better results.

7. REFERENCES

[1] *Ajit singh and Rimple Gilhotra " Data Security Using Private Key Encryption system Based On Arithmetic Coding" International Journal of Network Security & Its Applications (IJNSA), Vol.3, No.3, May 2011*

[2] *Tarun Narayan Shankar and G. Sahoo "Cryptography by Karatsuba Multiplier with ASCII Codes" 2010 International Journal of Computer Applications (0975 – 8887) Volume 1 – No. 12*

[3] *Fabiano C. Botelho and Nivio Ziviani. "External perfect hashing for very large key sets". in proceedings of 16th ACM Conference on Information and Knowledge Management (CIKM07), Lisbon, Portugal,November 2007.*

[4] *Rubata Riasat , Imran Sarwar Bajwa, M. Zaman Ali, A Hash-Based Approach for Colour Image Steganography. International Conference on Computer, Networks and Information Technology, ICCNIT 2011, Peshawar, Pakistan, 2011.*

[5] *Ross J. Anderson, Fabien A.P. Petitcolas, "On the limits of steganography"*

[6] *N. Provos, and P. Honeyman, "Hide and Seek: An Introduction to Steganography," IEEE Security and Privacy, 1(3): 32-44,May 2003.*

[7] *C. Kim, E.-J. Yoon, Y.-S. Hong, and H. 1. Kim, "Secret Sharing Scheme Using Gray Code based on Steganography," Journal of the Institute of Electronics Engineers of korea, 46(1): 96-102, January 2009.[3] C. C. Thien, and J. C. Lin, "Secret Image Sharing," Computers and Graphics, 26(1):765-770, February 2002.*

[8] *Yang, Li "Digital Watermarking". Canada, Ontario. University of Windsor, November 13, 2003.*

[9] *F. Shih, Digital watermarking and stegnography, fundamentals and techniques. UsSA: CRC Press, 2008.*

[10] *Johnson, N.,Jajodia, S. "Steganalysis of Images Created Using Current Steganography Software".Virginia, Fairfax. George Mason University. Center for Secure Information Systems.*

[11] *Neil F. Johnson, Sushil Jajodia, George Mason University, "Exploring Steganography: Seeing the Unseen"*, IEEE Computers, February 1998, pp. 26-34.

[12] Donovan Artz " Digital Steganography: Hiding Data within Data", IEEE Internet computing, pp.75-80 May –June 2001.

[13] Kefa Rabah, "Steganography – The Art of Hiding Data",Information Technology Journal 3 (3): 2004, ISSN 1682-6027, *2004 Asian Network for Scientific Information.*

[14] *Jinsuk Baek, Cheonshik Kim, Paul S. Fisher, and Hongyang Cha "(N, 1) Secret Sharing Approach Based on Steganography with Gray Digital Images".*

[15] *Miroslav Dobscek "Modern Steganography" In 8th International Student Conference on Electrical Engineering. FEE CTU, 2004.*

[16] *Beenish Mehboob and Rashid Aziz Faruqui "A Steganography Implementation" IEEE 2008.*

[17] *Imran Sarwar Bajwa, Rubata Riasat "A New Perfect Hashing based Approach for Secure Steganograph" IEEE 2010*

[18] *GPERF: A Perfect Hash Function Generator DouglasC.Schmidt schmidt@cs.wustl.eduLinux Software Directory, "Gperf - Perfect hash function generator", Available at: http://linux.maruhn.com/sec/gperf.html,*

*For Windows: gnuwin32.sourceforge.net/packages/**gperf**.htm*

[19] *hip21_en:tool Available at: sourceforge.net/projects/hide-in-picture/files/*

[20] *J. Wen, M. Severa, and W. Zeng, "A format-compliant configurable encryption framework for access control of video," IEEE Trans. CircuitsSyst. Video Technol, 12(6)(2002)545-557.*

Author1

Dr. Ajit Singh is presently working as Chairperson of School of Engineering & Sciences in BPSMV, Khanpur Kalan (Sonepat). He is also having the additional charge as a Director of University Computer Center (UGC). He posses qualifications of B.Tech, M.Tech, Ph.D. He is a member of BOG (Board of Governors) ofHaryana State Counselling Society, Panchkulaand also member of academic council in the University. He published approximate 20 papers in National/ International journals and conferences and holds a teaching experience of approximate 10 years. He holds the membership of Internal Quality Assurance cell, UG-BOS &PG-BOS and the NSS advisory committee. He is also an associate member of CSI & IETE. His research interests are in Network Security, Computer Architecture and Data Structure.

Author2

Ms. Upasana Jauhari has completed her B.Tech degree in Computer Science from CCS University, Meerut in year 2008. She is pursuing M.Tech in Computer Science from Bansathali University Banasthali from June 2010. Currently she is doing Internship from B.P.S.M.V Khanpur Kalan, Sonipat. Her research interests are in Network Security, Soft Computing, Distributed Computing.