Research Journal of Information Technology

ISSN 1815-7432

Academic Journals Inc.

www.academicjournals.com

Research Journal of Information Technology 6 (4): 399-405, 2014
ISSN 1815-7432 / DOI: 10.3923/rjit.2014.399.405

Multilevel Encryption System Using RC4 and Steganography

[1]Shinjit Kamal Borah and [2]Surajit Borah
[1]Department of Computer Science and Engineering, Tezpur University, Tezpur, 784028, India
[2]Department of Computer Science and Engineering, Girijananda Chowdhury Institute of Management and Technology, Guwahati-17, India

Corresponding Author: Surajit Borah, Department of Computer Science and Engineering, Girijananda Chowdhury Institute of Management and Technology, Guwahati-17, India

ABSTRACT

In this digital era, everybody keep their important and private data in their computers. At the same time, it becomes challenge to the computer professionals to protect their user's data from unauthentic third parties. There are lots of techniques to provide security to digital data and encryption is one of them. Encryption is a process of data encoding. In this study, an encryption method and its equivalent decryption method is explained. This encryption method is of two layers and capable to encrypt any type of file. In the first layer, data files are encoded using RC4 algorithm and in the next layer these encrypted data files are embedded inside a cover image. This study explains the details about the encryption and steganographic technique to hide the data file inside an image. Output image containing the data file remains same in naked eye view and can be easily transferred through the network. Our technique of embedding data inside the cover image is unique because it uses two metadata structures namely header and map array to randomly populate the data file inside the image. The header and map array are also attached to the image. Least Significant Bit (LSB) of red color is selected to embed the secret data file and metadata structures. In encryption process, a password is given and the same password is needed to decrypt the image. So, it can also be called as password protected steganography. This method provides security as well as privacy to the data files by hiding it.

Key words: Encryption, decryption, RC4 steganography, security

INTRODUCTION

Security is an important concern form the early ages. It plays its important role outside as well as inside the computer world. Encryption is a method to provide security to computer data. In computer science, encryption is a process of encoding information in such a way that a third party cannot access to it and only authorized party can access to the cipher information (Stinson, 1995). The encrypted information is known as cipher information. In this study, method is explained to combine two encryption methods namely RC4 and steganography algorithm and introduce some extra techniques to provide more security to data files. The RC4 algorithm is an encryption algorithm, which is the most widely used stream cipher algorithm in software applications (Dawson *et al*., 2002). It is also used in popular protocols such as Secure Sockets Layer (SSL), WPA (Wi-Fi Protected Access) etc. As RC4 stands for 'Rivest Cipher 4'. It generates a pseudorandom stream of bits; these are known as a keystream. This keystream is used to encrypt and decrypt the data (Pardeep and Pateriya, 2012; Juneja and Sandhu, 2013). Similarly, steganography is a

type of hidden writing that literally means covered writing (Judge, 2001). The goal of steganography is to hide an information message inside harmless cover medium in such a way that it is not possible even to detect that there is a secret message. Steganography was used in history, as it is found in some literature (Al-Ataby and Al-Naima, 2010). In digital steganography one file can be hidden inside another file. In current technology, the cover medium can be any type of file for example audio, video or image file. The method presented in this paper uses image as cover medium. This input image file can be of any type like Joint Photographic Experts Group (JPEG), Portable Network Graphics (PNG), Bitmap format (BMP) and Graphics Interchange Format (GIF). To hide the information, the bits of the information can be embedded anywhere in the cover medium or selectively in noisy areas. There are numbers of way to hide the data in a digital medium (Cheddad *et al*., 2010). Different steganographic techniques are classified as substitution systems, transform domain techniques, spread spectrum techniques, statistical methods, distortion techniques and cover generation methods (Juneja and Sandhu, 2013). Again steganographic methods can be classified as spatial domain, frequency domain and adaptive methods; and Least Significant Bit (LSB) replacement technique is used in spatial domain techniques (Cheddad *et al*., 2010). Least significant bit is used in the present encryption method. Least Significant Bit (LSB) is a common method to embed the data inside image. This can be done by changing one or more of the bits of the byte or bytes that make up the pixels of an image. In this method, LSB of red color of RGB is changed. The image color remains same because changing the LSB changes the integer value of byte by one only (Johnson and Katzenbeisser, 2000; Curran and Bailey, 2003). This multilevel encryption method is capable to hide a data file of any type. When an input file (which to be encrypt) and an image file is provided. At first the file is encrypted using RC4 and then steganographic techniques are used to hide the cipher file inside the image. Thus, users are able to keep their important files secretly and securely. Even, this can be transferred from one place to another without any threat. Moreover, the data are placed inside the image randomly which makes it more secure. To randomly populate the bits of data file a map array is used and a metadata, header is used to store the metadata of the data file and map array. This information is very important in the process of decryption. The header and map array are also encrypted and embedded inside the image. Decryption is the reverse process of encryption to get back the original file from the stego image (the image containing the cipher file as hidden).

In RC4 algorithm, with the help of a password given by the user a pseudorandom stream of bits (keystream) is generated. Then, bit-wise exclusive-or is applied between the input stream and keystream to get the cipher stream. Similarly, in decryption same keystream is combined with the cipher stream using exclusive-or to get the original stream. Since exclusive-or is a symmetric operation.

To generate the keystream, the algorithm makes use of a secret internal state which consists of two parts: A permutation of all 256 possible bytes (denoted as 'S' below) and two 8-bit index-pointers (denoted as 'i' and 'j'). The permutation is initialized with a variable length key, typically between 40 and 256 bits, using the key-scheduling algorithm (KSA). Once this has been completed, the stream of bits is generated using the Pseudo-Random Generation Algorithm (PRGA) (Noman *et al*., 2008; Pardeep and Pateriya, 2012).

The following steps are involved in encryption using RC4:

- Choose a secret key
- Run the KSA and PRGA using the key to generate a keystream
- XOR keystream with the data to generated encrypted stream

Key scheduling algorithm

1. The initialized array S[256] and use the input secret key to scramble the array
2. Take two integer say i,j and assign to i = 0 and j = 0
3. Assign the value of j as j = (j+S[i]+key [i mod key_length]) mod 256
4. Interchange S[i] and S[j] and increment the value of i by 1
5. Repeat step 3 and 4 till i<256

Pseudo-random generation algorithm

1. The KSA scrambled S [256] array is taken as input
2. Take two integer say i, j and assign to i = 0 and j = 0
3. Assign i = (i+1) mod 256
4. Assign j = (j+S[i]) mod 256
5. Interchange S[i] and S[j]
6. Output = S[(S[i]+S[j]) mod 256]
7. Repeat step 3 to 6 as an output is required

OUR APPROACH

In this method, RC4 and steganography are used to encrypt the input data file. At first, the input data file is encrypted using RC4 algorithm. While encrypting with RC4, a password is required and that password would be mutually known to both the sender and the receiver. Byte stream of the data file is encrypted using RC4 as mentioned earlier. The output of this module is a sequence of bytes. Then, these bytes are converted to a bit array. After that, these bits are embedded in the pixels of the image. LSB technique is applied here (Fridrich *et al*., 2001). We convert the randomly selected LSB of the red color of each pixel according to the bits of the bit array. Unlike other steganography techniques studied before, we tried to modify the last significant bit of the red color component only, keeping that of green and blue intact. To randomly select the pixels we generate a required number of random numbers and these numbers are stored in an array which is termed as map array. It is also important to check that one random number is not selected more than once. After the completion of map array pixels are selected according to random numbers stored there. In the decryption process, we read the random numbers from the array serially and select the corresponding pixel to get the bit. Thus, the original sequence of bits is obtained. A header is maintained to hold some of the most important information like File name, File length, File starting point, Map length, Map starting point and File extension. The information is extracted from the input data file and stored in the header before the encryption process starts. Decryption of the stego image is only possible because of these data. Header is also encrypted using RC4 algorithm. Header and map array are also hidden inside the image file. The algorithms required for encryption and decryption are as follows:

Encryption algorithm:

- Select the particular data file and use RC4 algorithm to encrypt it
- Convert the cipher file to bit array
- Generate a map array and populate it with random values
- Generate the header
- After the previous step, the data prepared in the above mentioned steps are written into the cover image according the map array. The LSB technique is used here

Decryption algorithm:

- Select the particular Stego image and extract the header file
- Decrypt the header file. Extract the information like starting point and size of the map array from it
- Extract the bits from the corresponding pixels using map array
- Finally, convert the bits to byte and apply decryption to get the original file

The size of header is always fixed. The size of the header is 1000 bits. That means 125 characters can be written in the header. It is always embedded from the first pixel to the 1000th pixel of the image. Therefore, the first step of decryption process is to extract the LSB of red color of these pixels and then decodes the bits to the actual characters. Rest of the work is done depending on that information. The size of the cover image should greater than the size of the data file. The minimum required size can be obtain using the Eq. 1:

$$X = A*(32+1)+1000 \quad (1)$$

where, X is minimum required size of cover image in pixels, A is size of cipher file after converting to bit i.e., number of bits of the cipher file, 1000 bits is reserved for the header and A*(32+1) bits is required for the map array and cipher file.

And the number of pixels of the cover image is always greater than this minimum value:

$$Y>X \quad (2)$$

where, Y is the number of pixels of the cover image and is calculated as:

$$Y = image\_Height*imgage\_Width \quad (3)$$

We calculate the number of pixels because only one bit of the pixels is to hold the information. Map array is an array of size A and it is an integer array because it holds the randomly generated integers. To populate this array in the cover image we have to convert the array to sequence of bits. The number of bits needed (B) is:

$$B = A*32 \quad (4)$$

The constant is 32, if the integers stored in the array are 32 bits integer. The array size is A because 'A' numbers of pixel positions are needed to embedded 'A' numbers of bits. In case of random number generation, a range is provided. The floor of the range is next to the 'A+1000' and ceiling is 'Y'. As mentioned earlier,1000 bits are reserved for the header and it is capable of holding 125 characters but sometimes all bits are not required and some bits may left out, in that case right padding with blank bit is needed to apply.

Decryption process is just reverse process of encryption. In this process first of all the header is extracted from 1st to 1000th pixel of the stego image and blank bits are simply removed from the header. After getting the all information like map length, map starting point, it identifies the map array bits from the stego image and coverts this bit sequence to the map array of 32 bit integer. The

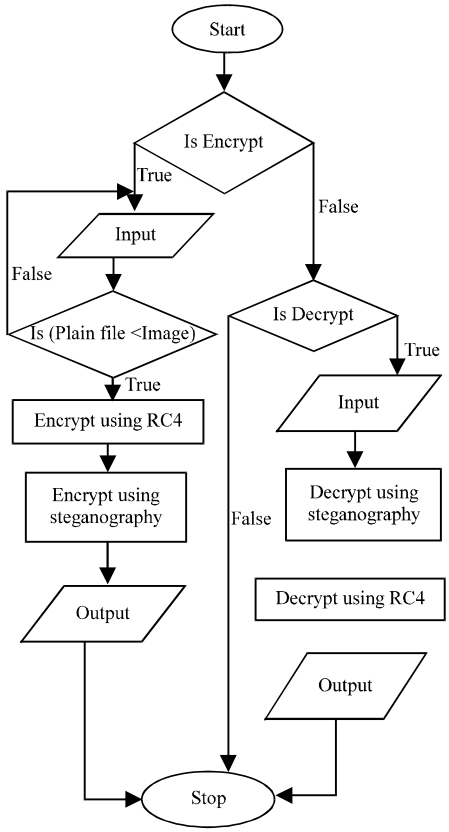

Fig. 1: Control flow chart

size of array will be 'map length/32'. Now, the pixels are selected according to integers found in the map array to get the required bit and generate the sequence of bits. After that, the sequence of bits is converted to a sequence of bytes. This is a sequence of encrypted bytes and then, decryption is applied to get the original file. File name and file extension are extracted from the header.

The control flow diagram of "Multilevel encryption system using RC4 and steganography" is shown in the Fig. 1. In the figure "Is Encrypt" is to check whether to encrypt or decrypt. Input of the encryption module is a data file, an image file and a password and the output of this module is an image file. That is a stego image containing the data file as hidden. Similarly 'Is Decrypt' is to check whether to decrypt or not. Input of decryption is the stego image and the password and the output is the original data file.

FEASIBILITY

This method can be easily implemented using currently available technology. Most of the currently available high level programming language has a rich library to provide build in functions or methods to handle with image and byte stream. The programming languages suitable to implement this method are Java, C# etc. We have implemented this method using net technology. We have used C#.net programming language. In C#, there are lots of library classes to handle data file and image file. We read the data file as byte stream. Data can be easily converted from byte to bit and vice versa using some build in function. Pixel can also be written easily (Stephens, 2008; Schildt, 2010). That makes our implementation somewhat easier. From the

implementation we conclude that the system is very rapid in performing encryption and decryption. The size of the data file does not affect the speed of the system very much.

Users can use our system very easily. Users can easily browse the data file and image file in encryption process. And output image can be easily saved in required directory. Output image of the encryption module is saved in bitmap format (bmp). In decryption process users can select the stego image to get the original data file. Users can save the output data file in any directory. The extension of file is provided by the system.

ADVANTAGE

When data files are encoded using currently available systems or methods. It becomes easily recognizable because data are not found in normal form. For example data of encrypted text file are found in unreadable format. If the encrypted file is easily detected, then crackers need to work with this specific file only to crack it. Thus it becomes somewhat easier for crackers. Similarly, steganography is a well known technique. If somehow image containing the information is identified it is easy to extract the information. But in case of this method, it will not be so easy because this method uses two level encryptions. Moreover, steganography is applied using randomly selected pixels. Thus, the advantages of this method can be pointed out as follows:

- Two level encryption method
- Secure steganograpic approach
- Password protected
- Covered medium can be easily stored and transferred

LIMITATION AND FUTURE ENHANCEMENT

Although, we have tried to add all the related features but there are also some limitations. These are described below:

- In this method, the plain file is embedded in an image file. Since each bit of the plain file is populated in the pixels of the image file. Therefore the size of image must be larger than that of plain file. As a result we cannot encrypt a large file (if there is no image file larger than that)
- Another limitation is that, in this paper we have not described any technique to embed a larger file

In future, a data compressing algorithm can be implemented to compress the data file. Such that a larger data file can be attached to an image file. Another method to encrypt a large size of file may be addition of a technique to split the large file into small files. And then these files are embedded to more than one image file. One of these two techniques will be sufficient to overcome from above mentioned limitations. Similarly, an automatic password generation method can also be implemented. Since, a password provided by user may be a weak password or may be matched with his some personal details and thus the possibility of cracking is increased. But when the password is generated by machine this possibility becomes less. Even in machine generated password, we can ensure the complicity using an efficient algorithm.

CONCLUSION

The method explained in this study is combination of steganography and cryptography. And it can be concluded that steganography is not intended to replace cryptography but rather to

supplement it. If a message is encrypted and hidden with a steganographic method it provides an additional layer of protection and reduces the chance of the hidden message being detected. This method can be defined as a secret key steganography since it needs a secret key to encrypt and decrypt. We have explained all the techniques required to implement a multilevel encryption system. Even, the reader will be able to implement some other encryption technique (other than RC4) with steganography to develop a multilevel encryption system after reading this paper. Some other techniques for example data compression and data splitting are needed to implement to provide more functionality to this method.

REFERENCES

Al-Ataby, A. and F. Al-Naima, 2010. A modified high capacity image steganography, technique based on wavelet transform. Int. Arab J. Inform. Technol., 7: 358-364.

Cheddad, A., J. Condell, K. Curran and P. Mc Kevitt, 2010. Digital image steganography: Survey and analysis of current methods. Signal Process., 90: 727-752.

Curran, K. and K. Bailey, 2003. An evaluation of image based steganography methods. Int. J. Digital Evid., 2: 1-40.

Dawson, E., H. Gustafson, M. Henricksen and B. Millan, 2002. Evaluation of RC4 stream cipher. Manuscript from Information Security Research Centre, Queensland University of Technology.

Fridrich, J., M. Goljan and R. Du, 2001. Reliable detection of LSB steganography in color and grayscale images. Proceedings of the Workshop on Multimedia and Security: New Challenges, October 1-5, 2001, Ottawa, Ontario, Canada, pp: 27-30.

Johnson, N.F. and S.C. Katzenbeisser, 2000. A Survey of Steganographic Techniques. In: Information Hiding Techniques for Steganography and Digital Watermarking, Katzenbeisser, S. and F.A. Petitcolas (Eds.). Artech House, Inc., Norwood, MA., USA.

Judge, J.C., 2001. Steganography: Past, present, future. Technical report No. 0414, Technical Information Center Oak Ridge Tennessee. http://www.ntis.gov/search/product.aspx?ABBR=DE200415006450

Juneja, M. and P.S. Sandhu, 2013. An improved LSB based Steganography with enhanced security and embedding/extraction. Proceedings of the 3rd International Conference on Intelligent Computational Systems, January 26-27, 2013, Hong Kong, China, pp: 29-34.

Noman, A.A., R.B. Sidek, M. Ramli and L. Ali, 2008. RC4A stream cipher for WLAN security: A hardware approach. Proceedings of the International Conference on Electrical and Computer Engineering, December 20-22, 2008, Dhaka, pp: 624-627.

Pardeep and P.K. Pateriya, 2012. PC1-RC4 and PC2-RC4 algorithms: Pragmatic enrichment algorithms to enhance RC4 stream cipher algorithm. Int. J. Comput. Sci. Network, 1: 98-108.

Schildt, H., 2010. The Complete Reference C. 7th Edn., The McGraw-Hill Companies, New York, NY., USA.

Stephens, R.C., 2008. Graphics Programming. Wiley Publishing, Inc., Hoboken, New Jersey, USA.

Stinson, D., 1995. Cryptography: Theory and Practice. 2nd Edn., CRC Press, Boca Raton, FL.