LSB-Based Random Byte Data Embedding

Wai Wai Zin
University of Computer Studies, Mandalay
waiwaizin.ucsmmdy@gmail.com

Abstract

Due to advances in technologies, most of the information is kept electronically. Steganography and Cryptography are two popular ways of sending vital information in a secret way. In this paper, an image steganography technique is proposed by combining cryptography and steganography. The proposed technique uses LSB-based data embedding technique and RC4-liked keyed state vector to hide the encrypted message. Before embedding the secret message, RC4 encryption algorithm is also used for message encryption. After encrypting, the encrypted messages can be embedded according to state vector size. The secret messages can be randomly hidden in BMP image file by using LSB technique. This system can protect the privacy of information. Moreover, this technique supports for data integrity and data confidentiality.

1. Introduction

Steganography become more important as more people join the cyberspace revolution. The goal of steganography is to avoid drawing suspicion to the existence of a hidden message [6]. By exploiting the human visual system (HVS), image steganography is very popular nowadays. The large amount of data can be embedded into an image while guaranteeing that the hidden data are perceptually invisible.

Steganography is often combined with cryptography to provide an additional layer of security. Using cryptography, data is encrypted and then transmitted. In steganography, the data is embedded in an image file and then the image is transmitted. It is possible to combine the techniques by encrypting message using cryptography and then hiding the encrypted message using steganography. This paper mainly focuses on to develop system with extra security features where a meaningful piece of text message can be hidden by combining security techniques like Cryptography and Steganography.

There are many encryption algorithms but RC4 encryption algorithm is used for data confidentiality in this system. In RC4 algorithm, encryption is about 10 times faster than DES and a particular RC4 key can be used only once. After encrypting the plaintext (original message), these encrypted messages are embedded in BMP image file by using LSB method. Least Significant Bits (LSB) insertion is a simple approach to embed secret information in image file. Altering the LSB will only cause minor changes in color, and thus is not usually noticeable to the human eye. This system improves the security of the data by embedding the encrypted text (ciphertext) and not the plaintext in an image. It also uses RIPEMD-160 hash function to check integrity of message.

This paper is organized as follows. Section 2 contains the related works. Section 3 gives the background theory about cryptography and steganography. Section 4 describes the steganographic method by using LSB insertion. BMP image format is illustrated in section 5.

Section 6 presents the details of the proposed system and conclusion remarks are given in section 7.

2. Related Works

Encryption and steganography are the preferred techniques for protecting the transmitted data. In [4], Mamta Juneja et al. presented a technique that combined steganography and encryption technique. The goal of this application was to help users to maintain their data's confidentiality. They described steganography tools based on LSB algorithms. Although they intended to support BMP, GIF, PNG images and WAV audio files as the carriers, their application only supported hiding data in BMP images. In [7], Neha Sharma et al. proposed a system that combines the effect of two methods such as cryptography and steganography to enhance the security of data. The authors also used MD5 hashing algorithm to provide the integrity of message contents. They can't compare their system with steganographic tools.

In [13], Yoendra Kumar Jai and R.R.Ahirwal presented a novel image steganography method. This method used LSB method and private stego-key. In this paper, they embedded binary bit stream in 24 bits color image or in 8 bits gray scale image. According to their results, their proposed system is better than the existing methods and better security. Experimental results verify that the proposed model is effective and efficient. W.W. Zin et al presented a Scattered LSB steganographic technique based on pseudo random number generator in [11]. This system uses LSB technique to insert the encrypted message in the BMP image file. In this system, PRNG repeatedly generated random sequence to select random block for data hiding.

3. Steganography and Cryptography

Steganography is the art and science of hiding communication; a steganographic system thus embeds hidden content in unremarkable cover media so as not to arouse an eavesdropper's suspicion [8]. The purpose of steganography is to hide the very presence of communication by embedding messages into innocuous-looking cover objects, such as digital images. To accommodate a secret message, the original cover image is slightly modified by the embedding algorithm to obtain the stego image. The embedding process usually incorporates a secret stego-key that governs the embedding process and it is also needed for the extraction of the hidden message [3].

In cryptography, the structure of a message is scrambled to make it meaningless and unintelligible unless the decryption key is available. It makes no attempt to disguise or hide the encoded message. Basically, cryptography offers the ability of transmitting information between persons in a way that prevents a third party from reading it. Cryptography can also provide authentication for verifying the identity of someone or something [11].

4. Steganographic Method Using LSB Insertion

The different types of steganographic techniques are Least Significant insertion (LSB), Masking and Filtering, and Transform Techniques.

The least significant bit insertion method is probably the most well known image steganography technique. It is a common, simple approach to embed information in a graphical image file. In digital, images are represented with the numerical values of each pixel where the value represents the color and intensity of the

pixel. To hide an image in the LSBs of each byte of the 24- bit image, one can store 3 bits in each pixel. A 1024 ×768 image has the potential to hide a total of 2,359,296 bits of information [7].

In this system, a BMP image file is used as a carrier to hide message. Least Significant Bit (LSB) insertion [6] is a simple approach for embedding information in image file. The simplest steganographic techniques embed the bits of the message directly into least significant bit plane of the cover image in a deterministic sequence.

The LSB insertion method also has the advantage of providing high capacity to embed data into images, particularly in captioning applications, when lots of information related to the image need to be embedded into the image. For example, in medical images, personal data and diagnosis of the patient are embedded into the image. Due to this advantage, LSB insertion is still the most common algorithm used in steganographic software today [10].

However, there are few weaknesses of using LSB. It is very sensitive to any kind of filtering or manipulation of the stego-image. Scaling, rotation, cropping, addition of noise, or lossy compression to the stego-image will destroy the message.

On the other hand, for the hiding capacity, the size of information to be hidden relatively depends on the size of the cover-image. The message size must be smaller than the image. A large capacity allows the use of the smaller cover-image for the message of fixed size, and thus decreases the bandwidth required to transmit the stego-image [1].

Another weakness is an attacker can easily destruct the message by removing or zeroing the entire LSB plane with very little change in the perceptual quality of the modified stego-image. Therefore, if this method causes someone to suspect something hidden in the stego-image, then the method is not successful.

5. BMP Image File Structure

Images are the most popular cover media for steganography and can be stored in a straight-forward bitmap format (such as BMP) or in a compressed format (such as JPEG). There are many applications for digital steganography of images, including copyright protection, feature tagging, and secret communications.

In this system, BMP image file is used as a container file. So if we were to modify the least significant bit (LSB) for every byte specifying color intensity, a human won't see the difference when the modified image is displayed. This is a very good opportunity for hiding information in the bitmap image. This is, however, an advantage for hiding data without raising suspicion.

The BMP file format is an image file format used to store bitmap digital images. In uncompressed bmp files and many other bitmap file formats, image pixels are stored with a color depth of 1,4,8,16,24 or 32 bits per pixel. An alpha channel for transparency may be stored in a separate file or in fourth channel that converts 24 bit images to 32 bits per pixel. Uncompressed bitmap files such as BMP files are typically much larger than compressed image file formats for the same image. For example an image of 1058 * 1058 pixels in png format occupies about 287.65 KB while in 24 bit BMP file it occupies about 3358KB [9].

A bitmap file has two main parts, the header and the data. The header, consists of 54 bytes, has two sub blocks: Bitmap Header, and Bitmap Information. Bitmap Header, 14 byte, is used to identify the file as a valid bitmap image and Bitmap Information is composed of the next forty bytes of the file.

6. LSB-Based Random Byte Data Hiding Technique

In random byte data hiding technique, the bytes of the message can hide into the least significant bits (LSBs) of the pixels within a carrier image, called the *cover image.* Random byte hiding is intended to make it harder for an attacker to detect the embedded secret message with attacks such as the visual attacks and statistical attacks.

6.1. RC4-Liked Keyed State Vector

The proposed method also uses the RC4-liked keyed state vector for data embedding. In this method, state vector size depends on message length. According to the encrypted message length and image size, state vector size can define. After accepting the user input key (K), RC4-liked keyed state vector is generated. To generate state vector, do the following: [12].

Initialization

1. L = message length
2. for i = 0 to 2^L do
3. SV [i] = i;
4. T [i] = K [i mod keylen] ;

Initial Permutation of S

5. j = 0;
6. for i = 0 to 2^L do
7. j = (j +SV [i] + T [i]) mod 2^L;
8. swap (SV [i], SV [j]) ;

After creating RC4-liked keyed state vector, the encrypted message can be embedded into LSB of BMP image file. Figure-1 shows the example of data embedding process. In this process, state vector size can be assumed by 2^8. In BMP image file, pixel data are divided into n-rounds. Each round has 256 bytes. According to this figure, the encrypted message is embedded to the LSB of each round by applying random byte position of RC4-liked keyed state vector.

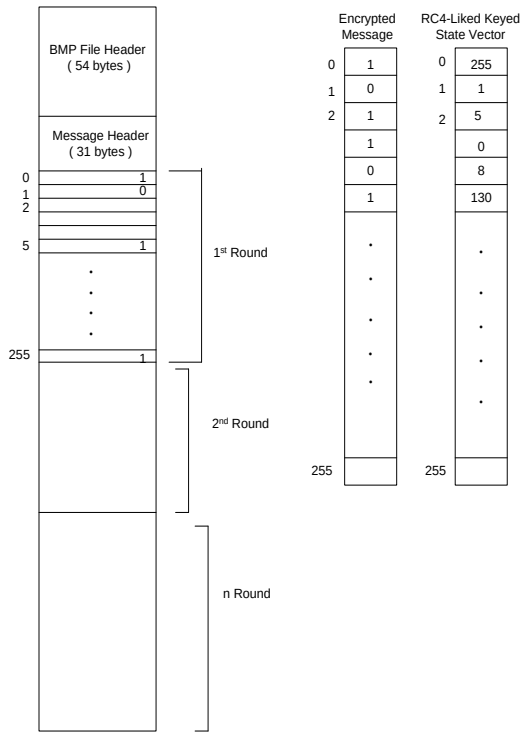

Figure 1. LSB-based data embedding process using RC4-liked keyed state vector

6.2. Preprocessing For Modified Message

In this method, message must be encrypted before inserting the secret data to the LSB of container image. RC4 encryption algorithm is used for data encryption in this system. Before hiding the encrypted data, header field is added to the encrypted message. Figure-2 shows the preprocessing for modified message. Hash code of the encrypted message and file extension are

included with these modified messages together.

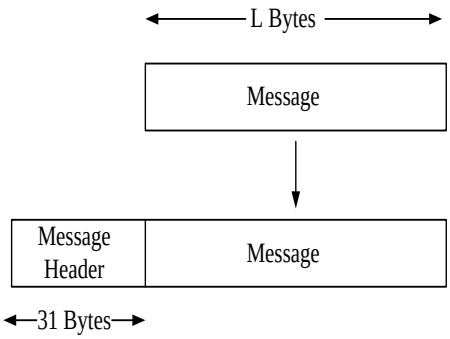

Figure 2. Preprocessing for modified message

6.3. RIPEMD-160 Hash Function

Hash algorithms are important components in many cryptographic applications and security protocol suites [2]. Hah functions, also called message digests and one way encryption, use no key. They are also employed by many operating systems to encrypt passwords. Therefore, it provides a measure of the integrity of a file.

In this paper, RIPEMD-160 hash algorithm is used to provide higher protection. RIPEMD-160 has been designed by Hans Dobbertin, Antoon Bosselaers and Bart Preneel and produces a 160 bit output after performing five independent rounds. Each round is composed of 16 iterations resulting in 80 iterations in total. RIPEMD-160 operates on 512-bit message blocks which are composed of sixteen 32-bit words. The compression function consists of two parallel data paths as shown in Fig. 3. F_i and F_i' are non-linear functions and K_i and K_i'are fixed constants. Temporary variables A, B, C, D and E for the left and A', B', C', D' and E' for the right data path, are initialized with the five 32-bit chaining variables, h_0, h_1, h_2, h_3 and h_4 respectively. Chaining variables are either initialized with the fixed values to hash the first 512-bit message block or updated with the intermediate hash values for the following message blocks. Each step of the algorithm uses a different message word Xi for the left and X_i' for the right data path. All the 16 message words are reused for each round but in a different order [5].

RIPEMD-160 Algorithm

1. $T = rol_s(A \oplus F_i(B, C, D) \oplus X_s \oplus K_i) \oplus E$
2. $E = D$
3. $D = rol_{10}(C)$
4. $C = B$
5. $B = T$
6. $A = E$
7. $T' = rol_{s'}A' \oplus F_s'(B', C', D') \oplus X_i' \oplus K_i' \oplus E'$
8. $E' = D'$
9. $D' = rol_{10}(C'')$
10. $C'' = B'$
11. $B' = T'$
12. $A' = E'$

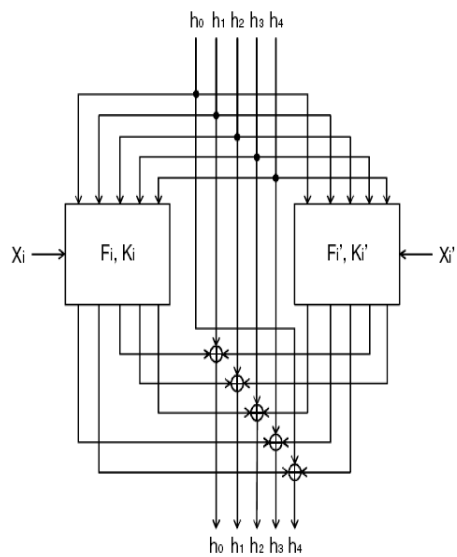

Figure 3. Comparison function of RIPEMD-160 algorithm

6.4. Embedding Process

In this system, we proposed the LSB-based random byte data embedding technique. According to figure-4, the plaintext or original message is encrypted with RC4 encryption algorithm. In this case, message digest is used as encryption key to generate ciphertext. The ciphertext or encrypted message is embedded in the BMP image file. In embedding process, LSB based random byte data embedding technique is applied. This technique is based on RC4-liked keyed state vector. By using random byte position, the encrypted message can be embedded to the LSB of container image. In this system, BMP image is used as the container image before embedding the secret messages. In this case, BMP file size is large enough to cover the entire messages. And then, we embed the messages to LSB of BMP file depending on state vector size. After embedding, the stego image, which is identical to original image, is generated.

Embedding Process

1. Load message file.
2. Load BMP container image.
(BMP image file size is large enough to hide the messages.)
3. Create state vector size according to encrypted message length and image size.
4. Accept user input key to create T vector.
5. Generate state vector depending on T vector as RC4-liked keyed state vector.
6. Select random number (Byte Position) depending on state vector, SV.
7. Calculate the number of rounds to embed the message as the following:

$$NumberOfRounds = \frac{CipherTextSize}{StateVectorSize} + 1$$

8. Embed the message into the container image according to the byte position of SV.
9. Save the stego image.
10. End.

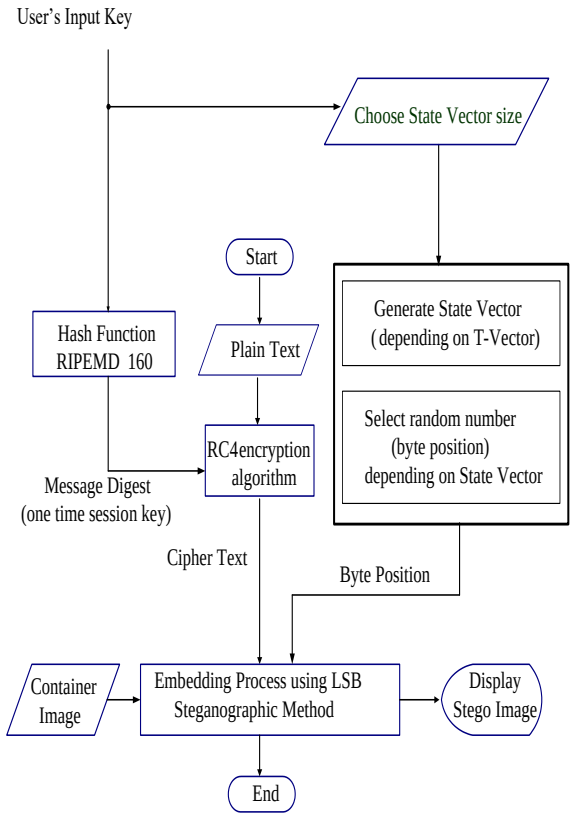

Figure 4. Overview process for data embedding

6.5. Extraction Process

In the extraction process, stego image is loaded and then corrected user's input key is accepted. Then the encrypted message is extracted according to byte position of state vector. Finally, we can generate the original message.

Extraction Process

1. Accept stego image and corrected user's input key.
2. Extract the encrypted message according to byte position.
3. Generate the original message from the secret encrypted message.

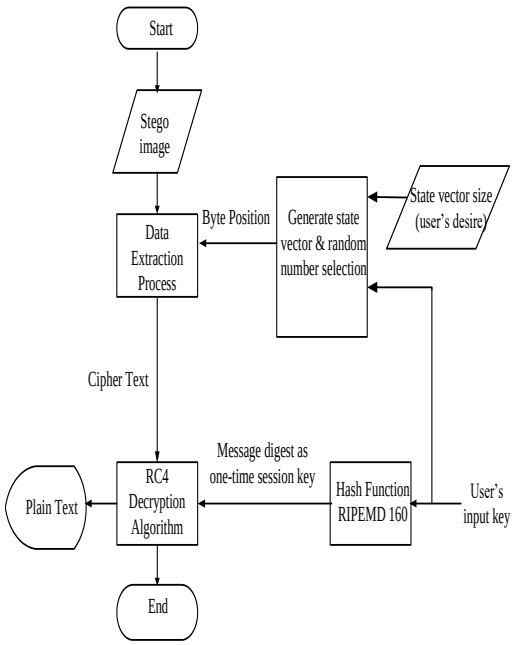

Figure 5. Overview process for data extraction

7. Conclusion

In this paper, a LSB-based random byte data hiding technique that enhances the message hiding security is proposed integrating cryptography and steganography. If message is encrypted, it would still need the decoding key to get the original message. In this proposed technique, RC4-like keyed state vector is used to embed messages. Besides, data byte is also inserted randomly according to the state vector size. Any attacker can't easily detect the embedded secret message. The proposed method can provide acceptable image quality with very little distortion in the image by using LSB insertion. Besides, this proposed method uses random byte data embedding technique. This LSB-based random byte hiding purposes to make it harder for detecting the original message. Since a slight change in color scheme is not detectable by the human eye, it can be used to hide information. Our system can be supported for data integrity and confidentiality.

References

[1] C.Cachin, "An Information-Theoretic Model for Steganography", in proceeding 2nd Information Hiding Workshop, vol-1525, pp-306-318, 1998

[2] Chiu-Wah Ng, Tung-Sang Ng and Kun-Wah Yip, "A Unified Architecture of MD5 and Ripemd-160 Hash Algorithms", 2004, IEEE.

[3] Jessica Fridrich and Miroslav Goljan, "Digital image steganography using stochastic modulation", Department of Electrical and Computer Engineering, SUNY Binghamton, Binghamton, NY, 13902-6000, USA.

[4] Mamta Juneja, Parvinder Singh Sandhu, "Designing of Robust Image Steganography Technique Based on LSB Insertion and Encryption", 2009 IEEE

[5] M.Knezevic, K.Sakiyama, Y.K.Lee, I.Verbauwhede, "On the High-Throughput Implementation of RIPEMD-160 Hash Algorithm", 2006.

[6] Muhalim Mohamed Amin, Subariah Ibrahim, Mazleena Salleh, Mohd Rozi Katmin, " Information Hiding Using Steganography", Department of Computer System & Communication Faculty of Computer Science and Information System, University Technology Malaysia, 2003

[7] Neha Sharma, Mr.J.S.Bhatia, Dr (Mrs) Neena Gupta, "An Encrypto Setgo Technique based secure

data transmission system", PEC, Chandigarh, May, 2005

[8] Niels Provos, Peter Honeyman, "Hide and Seek: Introduction to Steganography" , IEEE Security and Privacy, Volume 1, Issue 3 (May 2003), Pages: 32 – 44

[9] Rajanikanth Reddy Koppola, "A High Capacity Data Hiding Scheme in LSB Based Image", Thesis: The Graduate Faculty of the University of Akron, May 2009.

[10] Raphael C.W.Phan and H.C. Ling, "steganalysis of random lsb insertion using discrete logarithms", 2004, MMU International Symposium on Information and Communication Technologies 2003(M2USI 2003), Petaling Jaya, Malaysia, pp.56-59, 2003

[11] Wai Wai Zin, Than Naing Soe, "Scattered LSB Steganographic Technique Based on Pseudo Random Number Generator", The Ninth International Conference on Computer Application (ICCA 2011), May 5-6, Yangon, Myanmar.

[12] William Stallings, "Cryptography and Network Security", Principles and Practices, Fourth Edition, 2007.

[13] Yogendra Kumar Jain, R.R.Ahirwal, " A Novel Image Steganography Method With Adaptive Number of Least Significant Bits Modification Based on Private Stego-Keys ", International Journal of Computer Science and Security (IJCSS), Volume 4, Issue 1, pages 40-49.