Secure Data Hiding in Digital Image using LSB Technique

Shwe Sin Myat Than, Zin May Aye
Computer University, Mawlamyine
shwesinmyatthan@gmail.com, zinmay110@gmail.com

Abstract

Nowadays, digital communication has become an essential part of infrastructure, a lot of applications are Internet-based and in some cases it is desired that communication be made secret. In a digital world, steganography is intended to protect information by hiding a message in a way detected from unwanted parties. This system uses a secret key stego graphic technique for hiding messages. They are embedded by using digital images by LSB Least Significant Bit method. In steganography, LSB techniques are commonly used to embed the information in the cover media. In this system, Stego1bit LSB technique is applied to have the quality of stego image close enough to original image. Linear Congruential Generator (LCG) is used to generate pseudo random numbers for secret key. The digital images are applied as cover media in this system. In typical, the size of cover image need to be large enough for hidden messages. Receiver can extract the original messages from the stego image with the secret key. This system can provide security for hiding information with stego graphic technique.

Keywords: steganography, secret key stego, LSB method, stego image, LCG

1. Introduction

Data hiding become an important field as the use of the Internet became popular and growing in an exponential rate. Steganography is the art and science of communicating in a way, which hides the existence of the communication. 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. One solution to this problem is, through the use of steganography. In contrast to cryptography, it is not only to keep others from knowing the hidden information but it is to keep others from thinking that the information even exists. **Least Significant Bit (LSB)** insertion technique is an approach for embedding information in a cover image. The goal of Steganography is to hide messages inside other harmless messages in a way that does not allow any enemy to even detect that there is a second message present. In this paper, a **pseudorandom number generator** (**PRNG**) is applied for stego key, it is an algorithm for generating a sequence of numbers that approximates the properties of random numbers. The sequence is not truly random in that it is completely determined by a relatively small set of initial values, called the PRNG's **state (seed).** Although sequences that are closer to truly random can be generated using, hardware random number generator, *pseudorandom* numbers are important in practice for simulations, and are central in the practice of cryptography and procedural generation.

In this paper, Section 1 introduces the introduction of the system. Section 2 describes related works of the system. In Section 3 describes theory background. Section 4 describes design and implementation of the system and the last section delineates the conclusions.

2. Related work

Kevin Curran and Karen Bailey[3] analyzed an evaluation of image based Steganography methods. Muhalim, Mohamed Amin,Subariah Ibrahim, Mazleena Salleh and Mohd Rozi Katmin[5] descrided detailed implementation of information hiding using Steganography. They also explained about the application of steganography.

L.Y. Por, W.K. Lai, Z. Alireza and B. Delina defined and analyzed the system of StegCure: An Amalgamation of Different Steganographic Methods in GIF Images. They also described the use of image's bit depth and number of colors in usage [4].

3. Theory Background

In this section, brief discussion about steganographic techniques and LCG pseudo random number generator is presented.

3.1 Steganographic Techniques

Digital images, video, sound files, and other computer files that contain perceptually irrelvant or redundant information can be used as "covers" or carriers to hide secret messages. The carrier can then be sent to a receiver without anyone else knowing that it contains a hidden message. Combining the embedded message into the cover makes a stegomedia . After embedding a secret message into the cover-image so called stego-image is obtained. Also, Steganography can be described using the following formula: [5]

Cover media + embedded message + stegokey = Stegomedia.

The basic model of steganography consists of *Carrier*, *Message* and *Password*. Carrier is also known as *cover-object*, in which the message is embedded and serves to hide the presence of the message. Message is the data that the sender wishes to remain it confidential. It can be plain text, ciphertext, other image, or anything else that can be embedded in a bit stream such as a copyright mark, a covert communication, or a serial number. Password is known as *stego-key*, which ensures that only recipient who know the corresponding decoding key will be able to extract the message from a *cover-object*. The *cover-object* with the secretly embedded message is then called the *stego-object.*

There are basically three types of steganographic protocols:(1) Pure Steganography,(2) Secret Key Steganography and (3)Public Key Steganography. **Pure Steganography** does not require the exchange of a cipher such as a stego-key. **Secret Key Steganography** that requires the exchange of a secret key (stego-key) prior to communication. Secret Key Steganography takes a cover message and then secret message is encrypted and this encrypted message is embedded inside of cover one by using a secret key (stego-key). Only the parties who know the secret key can reverse the process and read the secret message. The benefit to Secret Key Steganography is even if it is intercepted, only parties who know the secret key can extract the secret message. **Public Key Steganography** that uses a public key and a private key to secure the communication between the parties wanting to communicate secretly. The sender will use the public key during the encoding process and only the private key, which has a direct mathematical relationship with the public key, can decipher the secret message[1]. There have been many techniques for hiding information or messages in images in such a manner that the alterations made to the image are perceptually indiscernible.

Least Significant Bit (LSB) insertion

The popular techniques in LSB of hiding data inside images are: Stego1Bit, Stego2Bits, .Stego3Bits, Stego4Bits, StegoColourCycle, Stego1BitPRNG and StegoFridrich[2].

3.2 Stego1Bit

This technique also known as LSB 1Bit. When images are used as the carrier in steganography they are generally manipulated by changing one or more of the bits of the byte or bytes that make up the pixels of an image. The message can be stored in the LSB of one colour of the RGB value or in the parity bit of the entire RGB value. One of the methods involves changing the least significant bit of each of the three colours in a pixel in a 24-bit BMP or JPEG image. Changing the LSB will only change the integer value of the byte by one. This will not noticeable alter the visual appearance of a colour and hence the image itself. In other words, one can store 3 bits in each pixel. Changing a more significant bit would cause a proportionately greater change in the visual appearance of a colour.The use of Stego1Bit example is as follow: If the size of the cover image(24bits) is 1024×768 pixel high resolution, the total pixels are 786432; and the total bits are 18874368. The total bytes is 2359296. One character exists 8 bits and 8 bytes or (8 LSBs) of color pixels are needed (for RGB) to hide one character in Stego1Bit. Therefore, the above image size can hide up to 294912 characters messages including message size, key length and key itself.

Advantages of LSB Insertion

Major advantages of LSB algorithm is , it is quick and easy. LSB insertion also works well with gray-scale images. A slight variation of this technique allows for embedding the message in one or more of the Least Significant Bits per byte. This increases the hidden information capacity.

3.3 LSB in GIF, 24 bit and JPEG

Graphics Interchange Formal (GIF) are 8 bit images and by using this image in stego system, only 1 bit can hide in one pixel with LSB 1 bit insertion technique. It allows for a smaller storage file size and also minimizes the transfer time over the network[6]. GIF images only have a bit depth of 8, and the amount of information that can be hidden is relatively less than 24 bit and JPEG images. An image size of 640 by 480 pixels, utilizing 256 colors (8 bit per pixels) would contain 300 K bits of data whereas 24 bit and JPEG of 1024 pixels wide by 768 pixels high would have a size in excess of 2 megabytes and would contain around 295 KB of characters.

3.4 Pseudorandom Number Generator(PRNG)

A **pseudorandom number generator** (**PRNG**) is an algorithm for generating a sequence of numbers that approximates the properties of random numbers. The sequence is not truly random in that it is completely determined by a relatively small set of initial values, called the PRNG's **state.** Although sequences that are closer to truly random can be generated using, hardware random number generator *pseudorandom* numbers are important in practice for simulations and are central in the practice of cryptography and procedural generation. Common classes of these algorithms are linear congruential generators, Lagged Fibonacci generators, linear feedback shift registers, feedback with carry shift registers, and generalised feedback shift registers[7].

Linear Congruential Generator(LCG)

In this system LCG is used for generating random numbers for a stego key. Linear congruential generator represents one of the oldest and best-known pseudorandom numbers generator(PRNG).
LCGs are defined by the recurrence relation :

$X_{i+1}=(aX_i+b) \bmod M$,

where ,

X_i – the sequence of random values,
M – the modulus M>0,
a – the multiplier 0<a<M,
b – the increment 0<b<M,
X_o – the seed $0<X_0<M$,

are integer constants that specify the generator.

The period of a general LCG is at most *M*, and for some choices of *a* much less than that. The LCG will have a full period if and only if:

1. b and M must be relatively prime
2. For each prime p such that p divide M, a-1 is a multiple of 4.
3. a-1 is a multiple of 4.

In mathematic, the integers a and b are said to be coprime or relatively prime if they have no common factor other than 1 and –1 ,or equivalently if their greatest common divisor is 1.

4. Implementation of Steganographic Information hiding system

In this section, the main processes for information hiding system are described. In this system, it consists of two processes : the embedding and the extraction.

4.1 The Embedding and Extracting Data

For data embedding process the stego key is firstly selected generated from LCG pseudo random number generator. And then the typing messages are encrypted with stego-key using XOR function. In this paper, if we want to embed the encrypted message character 'A' (01000001) , we need 8 bytes of color pixels . According to Least Significant Bit (LSB), the binary representation of the hidden data is taken and overwrite the LSB of each byte within the cover image. If 24-bit color is used , the amount of change will be minimal and indiscernible to the human eye. Three adjacent pixels (nine bytes) with the following RGB encoding are as follows:

R	G	B
10010101	00001101	11001001
10010110	00001111	11001010
10011111	00010000	11001011

To "hide" the following 9 bits of data (the hidden data is encrypted prior to being hidden) 101101101, overlaying these 9 bits over the LSB of the 9 bytes above, and will get the following (where bits in bold have been changed).

R	G	B
10010101	0000110**0**	11001001
1001011**1**	0000111**0**	1100101**1**
10011111	00010000	11001011

9 bits are successfully hidden but at a cost of only changing 4, or roughly 50% of the LSBs.

Embedding the Stego Key

In order to embed secret key messages, it must first be encrypted with a key generated using LCG. The encrypted messages, key length, key itself and length of message can then embed under the selected image. In this system these value are embedded sequentially as shown in Figure 1. Key does not need to be transferred in the system. It is one of the advantages of this system. After that the result (Stego-image) is sent to the desired destination.

Size of messages	First 4 bytes (int) of available bytes in the cover image.
Size of key	Next 4 bytes.
Key	According to the second 4 bytes value.
Encrypted Messages	Encrypted messages

Figure 1: Stego image Format

For data extracting process can be done in the backward way of embedding process. First, size of messages and size of key must be extracted to know where extracting process must be stopped and by using LSB 1Bit technique to recover the original data from stego image.

4.2 System Design Overview

From the sender side, hidden messages and cover image are embedded with Stego key which using Linear Congruential Generator (PRNG). After this process, Stego image is as the output. And then, the sender sends Stego image to the receiver.

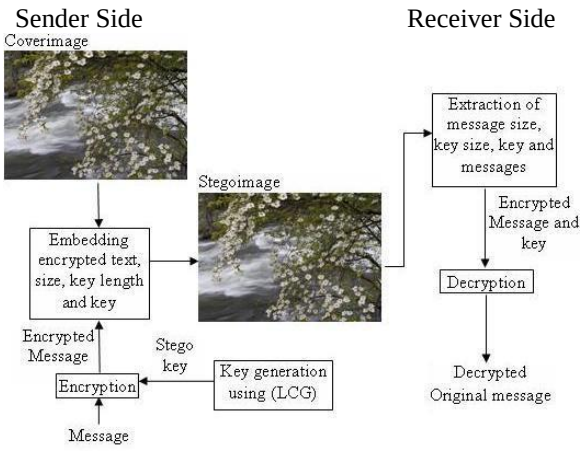

Figure 2: Overview of System Design

Data embedding and extracting steps are as follows:

Data Embedding

1. Get a cover image.
2. Take the information to be hidden (messages).
3. Generate LCG key to encrypt messages.
4. Encrypt message with generated key.
5. Combine cover image with the information to be hidden. (follow LSB algorithm)
6. Send defined receiver.

Data Extracting

During extraction, size of messages is extracted.

1. Extract key size.
2. Extract key according to the key size.
3. Extract messages according to the messages size.
4. Decrypt it using extracted key.

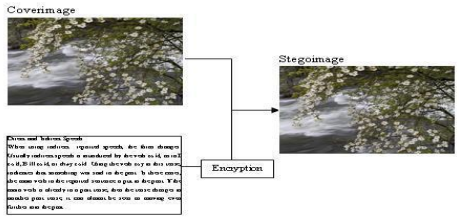

Figure 3: Producing A Stego-Image

5. Conclusions

In this paper, message is embedded and extracted to and from the image by using LSB 1 Bit insertion technique. Image steganography can be used for hidden information in secure communication. A stego-key has been applied to the system during embedment of the message into the cover image. In this paper the messages are embedded sequentially into the cover-image pixels instead of randomly. The best known Pseudo Random Number Generator, LCG applies for the generation of the stego key is one of the facts for the system to have robustness. The robustness of the system become better and it does not need to distribute the secret stego key between the end users. The system is implemented with Secret Key Steganography. So, it is more secure than Pure Steganography and hard for malicious attackers to recover the embedded message due to robustness of the system. This is so because without the knowledge of the valid key, it is difficult for a third party to get the original message.

References

[1] B. Dunbar, "A detailex look at Steganographic Technique and their use in an Open Systems Environment ", SANS Institute Reading Room site

[2] F. Khan, "PROPOSED IDEA FOR STEGANOGRAPHY"

[3] K. Curran, "An Evaluation of Image Based Steganography Methods", Internet Technologies Research Group, Institute of Technology, Ireland

[4] L.Y. Por, W.K. Lai, Z. Alireza, B. Delina, "StegCure: An Amalgamation of Different Steganographic Methods in GIF ".

[5] M. Mohamed Amin, S. Ibrahim, "INFORMATION HIDING USING STEGANOGRAPHY", Department of Computer System & Communication , UNIVERSITI TEKNOLOGI MALAYSIA 2003

[6] T. Morkel , M.S. Olivier ,"AN OVERVIEW OF IMAGE SEGANOGRAPHY", Department of Computer Science, University of Pretoria, South Africa

[7] http://www.Wikipedia.com, "Pseudorandom number generator" , the free encyclopedia