Image Steganography

Saketh Katari
Computer Science Department
IIIT-Delhi
Delhi, India
katari15045@iiitd.ac.in

***Abstract*—This document explains a new strategy to transfer confidential data by combining steganography with cryptography. LSB steganography is further improved by randomly storing confidential data in an image which makes it difficult to retrieve. Considering human visual system, it is difficult for the user to detect the changes in stego image.**

***Index Terms*—steganography, LSB, image, cryptography, random**

I. Introduction

Confidential data is everywhere and it is necessary to transfer it securely over internet. Cryptography comes into picture when the data is with you but nothing can be made out of it, as it is a cipher, making no sense. Steganography hides confidential data within some other data (say image) and makes it difficult to even detect that there is a secret in it. Cryptography combined with steganography provides a 2 layer protection - detecting the confidential data (steganography) and making sense out of it (Cryptography).

Images are used (among audio, video, text etc) as a medium to store the confidential data. Any data such as text, image, audio etc can be embedded in an image and is tranferred to the destination. If an attacker gets this image, it would be difficult to detect that there is some hidden data in it, as it looks lika a normal image.

II. Related work

Most of the traditional LSB steganography techniques store the confidential data in the LSBs (Least Significant Bit) of every pixel without any randomization, as shown below [1].

Confidential data
100110011

Pixels before embedding confidential data

R	G	B
10010110	01101111	01101011
01011100	11000110	01000011
10111111	11100110	00101110

Pixels after embedding confidential data

R	G	B
1001011**1**	0110111**0**	0110101**0**
0101110**1**	1100011**1**	0100001**0**
1011111**0**	1110011**1**	0010111**1**

During this process, the LSBs of some pixels are replaced, which modifies the image. LSB has less effect (changes by 1) on the overall value; hence it is modified instead of MSB.

However, this is not secure because, an attacker might just access the LSBs of every pixel and retrieve the confidential data. Instead, some randomization is necessary while embedding data so that, when an attacker tries to retrieve data, it is difficult to produce the same pattern that was randomly generated.

It is necessary to transfer the random pattern to the destination to retrieve data that is embedded.

The performance of a method or an algorithm is evaluated on the basis of the change in the image before and after embedding data in it. An algorithm is considered better if it produces minimal change in the image after embedding data in it.

III. Proposed model

This model adds cryptography and randomization to the traditional LSB steganographic techniques.

A. Notation

Cover_image - The image (before embedding) that is supposed to carry the confidential data in it.

Stego_image - The image (after embedding) that contains confidential data in it.

location_data - Meta data (Data about confidential_data) that identifies the location of confidential_data within a stego_image.

Component_of_a_pixel - The Red, Green and the Blue components (RGB).

LSB - Least Significant Bit of a binary number.

pr_key_src - Sender's private key

pr_key_dest - Receiver's private key

pub_key_src - Sender's public key

pub_key_dest - Receiver's public key

B. Overview

Cover Image + Confidential data + Location data

↓

Stego image

↓

Encrypted Stego image

Location Data → Encrypted location data

Encrypted stego image + Encrypted location data

↓

Internet

↓

Destination

Encrypted stego image + Encrypted location data

↓

Decrypted stego image + Decrypted location data

↓

Confidential data

C. Pre-processing

A random location_data is generated with necessary length (typically, equal to the size of confidential_data). Let it be 'abc'. This is converted to 8-bit ASCII values '97, 98, 99'. Finally, it is converted to a binary string '01100001, 01100010, 01100011'.

Similarly, another binary string is produced for the confidential_data.

D. Embedding

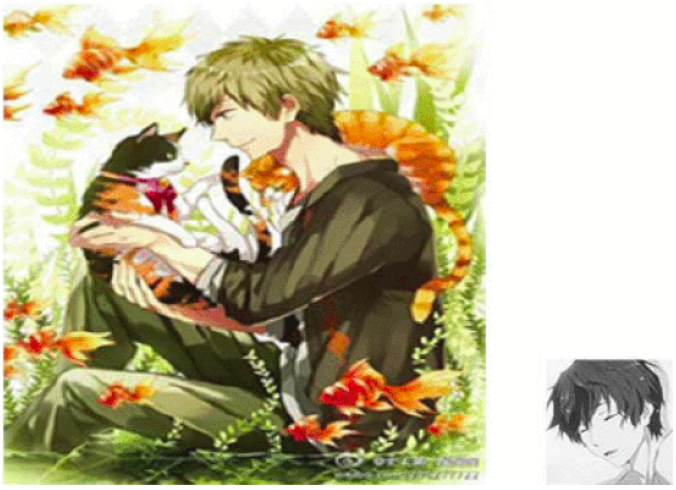

Fig. 1. Left - Cover_image, right - confidential_data_image [2].

Grayscale image (Black and white) in Fig.1 is to be embedded in the color image to it's left.

Each bit is taken from the confidential_data and from the location_data; let them be c_bit and l_bit respectively. c_bit is embedded in one of the components of a pixel which is determined by l_bit.

It is observerd that the human eye is most sensitive to the green component of the light [2]. So, LSB of the green component is not replaced.

A pixel is chosen in lexicographical order (row_number, column_number in lexicographical order). For a given l_bit and c_bit, l_bit is XORed with LSB of green component. If it gives 0, LSB of blue is replaced with c_bit, else, LSB of red is replaced with c_bit [2].

Fig. 2. Left - Cover_image, right - stego_image [2].

As you can see in Fig.2, it is difficult to distinguish between the left (before embedding confidential_data) and right images (after embedding confidential_data)

E. Encryption

Stego_image is first encrypted with the pr_key_src and then with the pub_key_dest. Encrypting with pr_key_src is referred as a digital signature. Encrypting with pub_key_dest makes sure that only the intended user can decrypt it with his/her private key [2].

pub_key_dest (pr_key_src (data))

Similarly, location_data is encrypted.

Now, encrypted_stego_image along with encrypted_location_data is transferred to destination.

F. Decrpytion

Stego_image is first decrypted using the pr_key_dest and then with the pub_key_src. Now, the receiver has the decrypted stego_image .

pub_key_src (pr_key_dest (data))

Similarly, decrypt the location_data.

G. Extracting embedded data

A pixel is chosen in lexicographical order (row_number, column_number in lexicographical order). For each bit l_bit in location_data, XOR l_bit with the LSB of the green component. If it's 0, append the LSB of blue to the confidential_data, else, append the LSB of red to the confidential_data.

l_bit - 1

Pixel -
R(10010111)
G(01101111)
B(01101010)

c_bit = LSB_of_Green **XOR** l_bit
= 0 **XOR** 1 = 1 i.e LSB of red component = 1

Similarly, construct the whole confidential_data which results in a binary string. Convert this binary string to 8-bit ASCII values and finally to ASCII characters.

H. PSNR (Peak Signal to Noise Ratio)

As name indicates, it is the ratio of signal to noise. Higher the PSNR lesser the noise in an image.

PSNR of traditional LSB based steganography techniques is around 53 db [1]. This method gives a PSNR of around 56 db, better than traditional PSNR.

I. Security

It has two layers of security - steganography and cryptography.

It's difficult to detect confidential_data that is embedded in an image. If an attacker somehow gets encrypted data, it's difficult to decrypt it.

IV. Conclusion

Traditional LSB steganography technique has been improved in three ways -

1. Two way security with cryptography.
2. Making it difficult for an attacker to predict the random pattern that is used to embed confidential_data in a cover_image.
3. PSNR is increased from 53 db to 56 db.

Acknowledgment

I would like to thank 'Technical Communication [COM 301A]' staff and IIIT-Delhi (Indraprastha Institute of Information Technology).

References

[1] S. M. Masud Karim, Md. Saifur Rahman, Md. Ismail Hossain, "A new approach for LSB based image steganography using secret key". Computer and Information Technology (ICCIT), 2011 14th International Conference, 22-24 Dec. 2011, Dhaka, Bangladesh.

[2] Xinyi Zhou, Wei Gong, WenLong Fu, LianJing Jin, "An improved method for LSB based color image steganography combined with cryptography". Computer and Information Science (ICIS), 2016 IEEE/ACIS 15th International Conference, 26-29 June 2016, Okayama, Japan.