Security Enhancement: Combining Cryptography and Steganography

Dhawal Seth
CSE
Vellore Institute of Technology
Vellore, Tamil Nadu, India.

L. Ramanathan
Assistant Professor
School of Computing Sciences and Engineering
Vellore Institute of Technology
Vellore, Tamil Nadu, India.

Abhishek Pandey
CSE
Vellore Institute of Technology
Vellore, Tamil Nadu, India.

ABSTRACT

In today's world of high technology, it is not safe to share confidential and important data on any network. Intruders are always in wake of it. They hack the data and use it for their benefit. These malicious people try to gain benefit, get attention, or to harm someone. In either case, message sender or receiver has to pay the price. To avoid these undesirable acts, Steganography and cryptography are used together to ensure security of the covert and secure message. One of the most efficient and secure algorithms is Data Encryption Standard (DES). Steganography is the art and science of writing hidden messages in such a way that no-one apart from the sender and intended recipient even realizes there is a hidden message.

Keywords

Cryptography, Encryption, Decryption, Steganography, Cipher text.

1. INTRODUCTION

Cryptography is a science and act of manipulating messages to make them secure. At first, Cipher used simple encrypting technique to secure his military messages. To encrypt messages, one simply used succeeding third alphabet in place of actual alphabet .For example, if one wanted to send "attack"; one sent "dwwdfn". The receivers knew the key, so they decrypted the message and in case any unauthorized person got the message he could not understand it. DES (Data Encryption Standard) is one of the cryptographic algorithms.

An original message to be transformed is called plain text and resulting message after transformation is called cipher text. The process of converting plain text to cipher text is called encryption. The reverse process is called decryption. Encryption and decryption requires the use of secret key. The art of secret writing is called Steganography. It is the art and science of invisible communication. This is accomplished through hiding information in other information, thus hiding the existence of the communicated information. The word Steganography is derived from the Greek words "stegos" meaning "cover" and "grafia" meaning "writing" defining it as "covered writing". In image Steganography the information is hidden exclusively in images. It includes hiding messages in other messages like text or picture and which can be decoded by person knowing the key.

2. THE PROPOSED METHOD

In this paper, we propose the combination of encryption and Steganography to enhance the security of the data to be sent. The whole process is to be carried in three steps which are as follows:

2.1. The Encryption

DES algorithm works by encrypting groups of 64 message bits, which is the same as 16 hexadecimal numbers. To do the encryption, DES uses "keys" where are also apparently 16 hexadecimal numbers long or apparently 64 bits long. However, every 8th key bit is ignored in the DES algorithm, so that the effective key size is 56 bits. But, in any case, 64 bits (16 hexadecimal digits) is the round number upon which DES is organized. DES has 19 steps for encrypting the plaintext. They are as follows:-

1: This step performs initial permutation on 64 bit plaintext. This step is independent of key. There is not much effect of permutation on security.

2: Now iterations are performed. 16 different iterations performed in 16 steps and each iteration uses different key. The key initially has 64 bits but out of it the least significant digits are taken out. They are used for parity checking. This makes the key of 56 bits. Permutation is performed to the 56 bits and 48 bits are taken out to be used as key.

3: As shown in figure 1 below, to perform iteration, 64 bit input is divided into 2 equal portions denoted by L (i-1) and R (i-1). For 1<=**i**<=16, using a function **f** which operates on two blocks--a data block of 32 bits and a key K_i of 48 bits--to produce a block of 32 bits. Let + denote XOR addition, (bit-by-bit addition modulo 2). Then for i going from 1 to 16 we calculate

$\mathbf{L_i = R_{i-1}}$
$\mathbf{R_i = L_{i-1} + f(R_{i-1},K_i)}$

The output generates two 32 bit blocks L(i) and R(i). The left part of output is simply the right part of input. The right part of output is bitwise XOR of left part and function of right part of input and key at given iteration. The key at each step is derived from initial 56 bit.

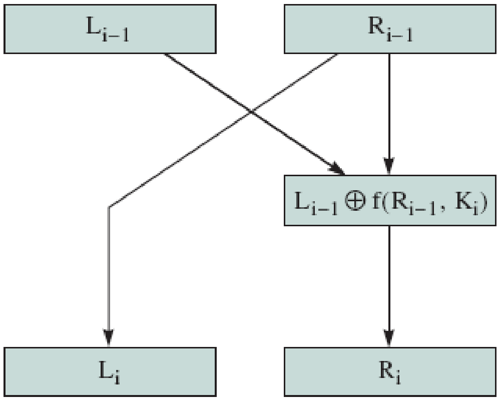

Figure 1

We have not yet finished calculating the function f. To this point we have expanded R_{i-1} from 32 bits to 48 bits and XORed the result with the key K_i. We now have 48 bits, or eight groups of six bits. We now do something strange with each group of six bits: we use them as addresses in tables called "S boxes". Each group of six bits will give us an address in a different S box. Located at that address will be a 4 bit number. This 4 bit number will replace the original 6 bits. The net result is that the eight groups of 6 bits are transformed into eight groups of 4 bits (the 4-bit outputs from the S boxes) for 32 bits total.

Write the previous result, which is 48 bits, in the form:

$K_i + E(R_{i-1}) = B_1B_2B_3B_4B_5B_6B_7B_8$,

where each B_i is a group of six bits. We now calculate

$S_1(B_1)S_2(B_2)S_3(B_3)S_4(B_4)S_5(B_5)S_6(B_6)S_7(B_7)S_8(B_8)$

where $S_i(B_i)$ refers to the output of the i-th S box.

Example: For the first round, we obtain as the output of the eight S boxes:

$K_1 + E(R_0)$ = 011000 010001 011110 111010 100001 100110 010100 100111.

$S_1(B_1)S_2(B_2)S_3(B_3)S_4(B_4)S_5(B_5)S_6(B_6)S_7(B_7)S_8(B_8)$ = 0101 1100 1000 0010 1011 0101 1001 0111

The final stage in the calculation of **f** is to do a permutation **P** of the S-box output to obtain the final value of **f**:

$f = P(S_1(B_1)S_2(B_2)...S_8(B_8))$

The permutation **P** is defined in the following table. **P** yields a 32-bit output from a 32-bit input by permuting the bits of the input block. This whole procedure is repeated 16 times.

4: After 16 iterations, 32 bits of left and 32 bits right are swapped.

5: Then the reverse permutation is applied. This is just opposite to what applied in the initial step.

After these steps, cipher text is obtained. The following figure 2 shows all steps of encryption:

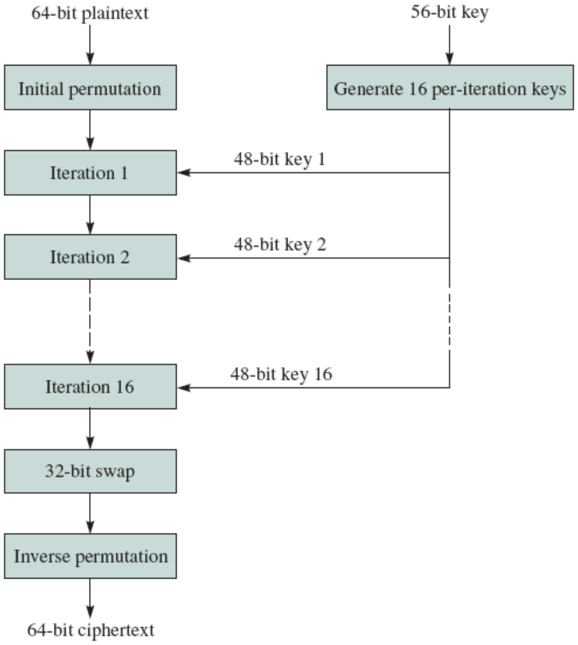

Figure 2

2.2 The Steganography

Now we have the encrypted data that is to be sent over the channel to the receiver such that it is not hampered. So the cipher text obtained in the above step is taken and hidden into an image using the process of Steganography. Steganography serves two main purposes:

1. Encode message of plain info.
2. Decode Message from steg Data.

Plain info is an input this is normally derived from encrypted data. Cover data is another input. In our case, it is a picture message. The minimum amount of Cover data is related to the amount of plain info. The process one is of encoding the plain info with the cover data.

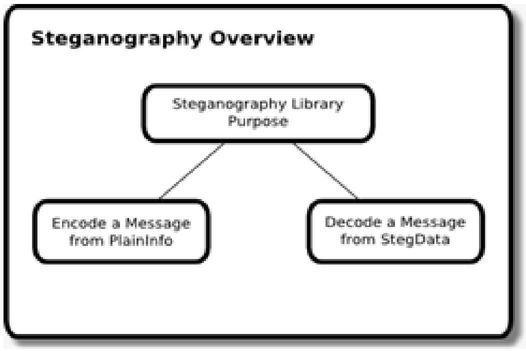
Steganography Overview
Steganography Library Purpose
Encode a Message from PlainInfo
Decode a Message from StegData

Figure 3

The cover data will normally have to be prepared in some way to encode the data. The output is the resulting steg data. The Input is:

- Plain Info (encrypted data from DES)
- Image (cover data)

An image file is merely a binary file containing a binary representation of the color or light intensity of each picture element (pixel) comprising the image.

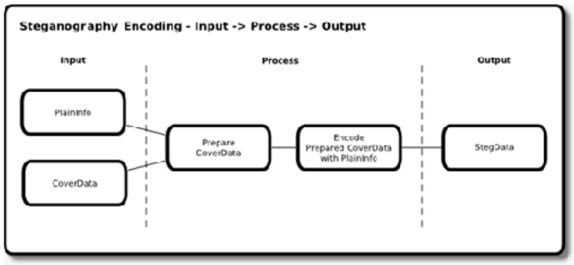
Steganography Encoding - Input -> Process -> Output
Input
Process
Output
PlainInfo
CoverData
Prepare CoverData
Encode Prepared CoverData with PlainInfo
StegData

Figure 4

Images typically use either 8-bit or 24-bit color. When using 8-bit color, there is a definition of up to 256 colors forming a palette for this image, each color denoted by an 8-bit value. A 24-bit color scheme, as the term suggests, uses 24 bits per pixel and provides a much better set of colors.

In this case, each pixel is represented by three bytes, each byte representing the intensity of the three primary colors red, green, and blue (RGB), respectively. The size of an image file, then, is directly related to the number of pixels and the granularity of the color definition. A typical 640x480 pix image using a palette of 256 colors would require a file about 307 KB in size (640 • 480 bytes), whereas a 1024x768 pix high-resolution 24-bit color image would result in a 2.36 MB file (1024 • 768 • 3 bytes).

GIF and 8-bit BMP files employ what is known as *lossless* compression, a scheme that allows the software to exactly reconstruct the original image. JPEG, on the other hand, uses *lossy* compression, which means that the expanded image is very nearly the same as the original but not an exact duplicate. While both methods allow computers to save storage space, lossless compression is much better suited to applications where the integrity of the original information must be maintained, such as Steganography. While JPEG can be used for stego applications, it is more common to embed data in GIF or BMP files. The Process is:

- We have to check cover Data amount against plain Info.
- Encode plain Info with cover Data (picture).

The approach to hiding data within an image file used here is called *least significant bit (LSB) insertion*. In this method, we can take the binary representation of the hidden_data and overwrite the LSB of each byte within the cover_image. If we are using 24-bit color, the amount of change will be minimal and indiscernible to the human eye. As an example, suppose that we have three adjacent pixels (nine bytes) with the following RGB encoding:

10010101 00001101 11001001
10010110 00001111 11001010
10011111 00010000 11001011

(Before)

(After)

Now suppose we want to "hide" the following 9 bits of data (the hidden data is usually compressed prior to being hidden): 101101101. If we overlay these 9 bits over the LSB of the 9 bytes above, we get the following (where bits in **bold** have been changed):

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

Output:

- Steg Data

Hereby the Steganography process is consummated with the hiding of the data in the image. Now this image can be sent to the receiver over an unsecure channel.

2.3 The Decryption

To decrypt the message on the receiver's end, first the cipher text has to be decoded out of the image in which it is hidden. This module is just opposite to encryption module. All the steps followed for encrypting the plain text are applied in reverse order to cipher text to get the original message.

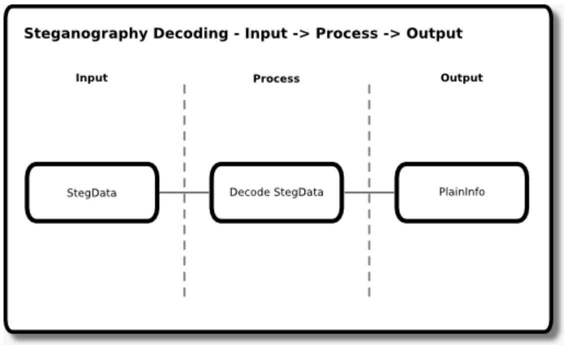

Figure 5

Steg data is the input, the decode method has to relate to the encode method of the Steg data. The process is to decode the Steg data. Plain info should then result as the output.

Input:
- Steg data

Process:
- Decode Steg data

Output:
- Cipher Text

The plain info obtained is the encrypted data of DES algorithm. This cipher text is decrypted by following the DES algorithm steps explained above using the symmetric key the receiver is having. All the same steps are repeated again in the same sequence to get the deciphered text i.e. the plain information.

3. CONCLUSION

Cryptography can protect your data from thieves and impostors. You can encrypt the files on your hard disk so that even if your enemies gain physical access to your computer, they won't be able to access its data. Cryptography can make it hard to forge email and hard to read other people's messages. Steganography is a really interesting subject and outside of the mainstream cryptography and system administration that most of us deal with day after day. But it is also quite real; this is not just something that's used in the lab or an arcane subject of study in academia.

Although encrypted data is difficult to decipher, it is relatively easy to detect. Encryption only obscures a message's meaning, not its existence. Therefore, Steganography, a technique that hides the existence of a message, can be used to supplement encryption. The technique can be used everywhere which requires transfer of data through network. It can be used to transfer the military messages. It can be used in banks to secure important information from intruders. It can be used by companies to secure their confidential data. It is beneficial for securely storing sensitive data, such as hiding system passwords or keys within other files.

5. REFERENCES

[1] Anderson, R.J. & Petitcolas, F.A.P., "On the limits of Steganography", IEEE Journal of selected Areas in Communications, May 1998.

[2] Applied Cryptography, Second Edition - John Wiley & Sons, Bruce Schneier.

[3] Communication networks: fundamental concepts and key architectures - Alberto León-García, Indra Widjaja.

[4] Davies, D.W.; W.L. Price (1989). Security for computer networks, 2nd edition John Wiley & Sons.

[5] Digital Steganography: Hiding Data within Data- Artz, D.; Los Alamos Nat. Lab., NM.

[6] Introduction to Cryptography – Ranjan Bose – Tata Mc-Grew– hill Publisher ltd, 2001.

[7] Jamil, T., "Steganography: The art of hiding information is plain sight", IEEE Potentials, 18:01, 1999.

[8] Johnson, N.F. & Jajodia, S., "Exploring Steganography: Seeing the Unseen", Computer Journal, February 1998.

[9] Moerland, T., "Steganography and Steganalysis", Leiden Institute of Advanced Computing Science.

[10] R. L. Rivest, A. Shamir, L. Adleman. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. Communications of the ACM 21, 2 (Feb, 1978), 120-126.

[11] Silman, J., "Steganography and Steganalysis: An Overview", SANS Institute, 2001.