ISSN:0975-9646

Securing Information: Cryptography and Steganography

[1] Ramakrishna Mathe [2] Veera RaghavaRao Atukuri [3] Dr. Srinivasa Kumar Devireddy

[1 2] *Department of Computer Science and Engineering*
Malineni Lakshmaiah Women's Engineering College, Guntur (A.P), INDIA.
[3] *Prinicpal, Nalanda Institute of Engineering and Technology,*
Sattenapally, GUNTUR

Abstract:**Today's dynamic and information rich environment, information systems have become vital for any organization to survive. With the increase in the dependence of the organization on the information system, there exists an opportunity for the competitive organizations and disruptive forces to gain access to other organizations information system. This hostile environment makes information systems security issues critical to an organization. Current information security literature either focuses on anecdotal information by describing the information security attacks taking place in the world or it comprises of the technical literature describing the types of security threats and the possible security systems. Two of the best ways to provide security is Cryptography and Steganography. Cryptography and Steganography are cousins in the spy craft family. Cryptography scrambles a message so it cannot be understood and generates *cipher text*. Steganography word is derived from Greek, literally means "Covered Writing". Steganography is the art of hiding the existence of data in another transmission medium to achieve secret communication. It does not replace cryptography but rather boosts the security using its obscurity features. It includes vast ways of secret communications methods that conceal the message's existence. These methods are including invisible inks, microdots, character arrangement and covert channels & spread spectrum communications.**

In Cryptography, the meaning of data has been changed. So, it makes intention to the hacker to hack or destroy the data. In our proposed paper, we implement a method by mixing both Cryptography and Steganography for Information security. It not only changes the meaning of data but also hides the presence of data from the hackers. In order to secure the transmission of data, Steganography has to be implemented that allow information to be sent in a secure form in such a way that the only person able to retrieve this information is intended recipient.

In this paper we proposed a method which describes two stages for sending the information securely by using the Public-key Cryptography and Steganography based on matching method. This is done in following steps:

1. **Encrypt the message using any one of the popular Public-Key Encryption Algorithms, so that only authorized parties can only be able to read the message.**
2. **Find and share stego-key between the two communication parties over insecure networks by applying Diffie Hellman Key exchange protocol.**
3. **Sender uses the secret stego-key to select pixels that it will be used to hide the message obtained in first step. Each selected pixel used to hide 8 bits of information.**

This steganographic protocol is more efficient than LSBs. It produces matching between the data bit parts and selected or least significant bits of each pixel.

Keywords: Public-Key Cryptography, Steganography, Stego-key, Diffie-Hellman, LSBs.

1. INTRODUCTION

Cryptography and Steganography are often interrelated and share the common goals and services of protecting the confidentiality, integrity and availability of information; which are some of the most important fields in computer security. Cryptography and steganography are methods of transferring private information and data through open network communication, so only the receiver who has the secret key can read the secret messages which might be documents, images or other forms of data.

Cryptography and steganography also contribute to Computer Science, particularly, in the techniques used in computer and network security for access control and information confidentiality. They are also used in many applications encountered in everyday life. Despite the differences between Cryptography and Steganography systems the requests for them have increased recently for the fast development of the Internet publicly.

Cryptography is the study of mathematical techniques related to aspects of information security such as confidentiality, data integrity, entity authentication, and data origin authentication. In addition, Cryptography is also known as the science of secret writing. The goal of cryptography is to make data unreadable by a third party. Cryptography algorithms are divided into symmetric (secret-key) and asymmetric (public-key) network security protocols. *Symmetric algorithms* are used to cipher and decipher original messages (plaintext) by using the same key. While *Asymmetric algorithms* uses public-key cryptosystem to exchange key and then use faster secret-key algorithms to ensure confidentiality of stream data. In Public-key encryption algorithms, there is a pair of keys, one key is known to the public, and is used to encrypt information to be sent to a receiver who owns the corresponding private key. The private and public keys are both different and need for key exchange.

Steganography is the science of hiding selected information from a third party. Therefore, steganography in contrast with cryptography, where the existence of the message is clear, but the meaning is obscured.

Steganography applications conceal information in other, seemingly innocent media. Steganographic results may masquerade as other file for data types, be concealed within various media, or even hidden in network traffic or disk space. There are many ways in which information and data can be exploited to conceal additional information.

For many years Information Hiding has captured the imagination of researchers. Digital watermarking and steganography techniques are used to address digital rights management, protect information, and conceal secrets. Information hiding techniques provide an interesting challenge for digital forensic investigations. Information can easily traverse through firewalls undetected.

This paper proposes a new approach to *public-key steganography* based on matching method to hide the secret information inside 24-bit image file. In the proposed method, the stego-key is generated by applying a public key exchange protocol which is based on discrete logarithm hard problem such as Diffie Hellman.

2. Related Work: Public-key Cryptosystem

During the early history of cryptography, two parties would rely upon a key using a secure, but non-cryptographic, method; for example, a face-to-face meeting or an exchange via a trusted courier. This key, which both parties kept absolutely secret, could then be used to exchange encrypted messages. A number of significant practical difficulties arise in this approach to distributing keys. Public-key cryptography addresses these drawbacks so that users can communicate securely over a public channel without having to agree upon a shared key beforehand.

An asymmetric-key cryptosystem was published in 1976 by *Whitfield Diffie* and *Martin Hellman*, who, influenced by *Ralph Merkle's* work on public-key distribution, disclosed a method of public-key agreement. This method of key exchange, which uses exponentiation in a finite field, came to be known as Diffie–Hellman key exchange. The Diffie-Hellman key exchange protocol was the first system to utilize public-key or two-key cryptography. For this reason, it is sometime called as Asymmetric *encryption*. This was the first published practical method for establishing a shared secret-key over an authenticated (but not private) communications channel without using a prior shared secret.

2.1 Public-key Cryptography:

Public-key cryptography refers to a cryptographic system requiring two separate keys, one to lock or encrypt the *plaintext*, and one to unlock or decrypt the *cipher-text*. Neither key will do both functions.

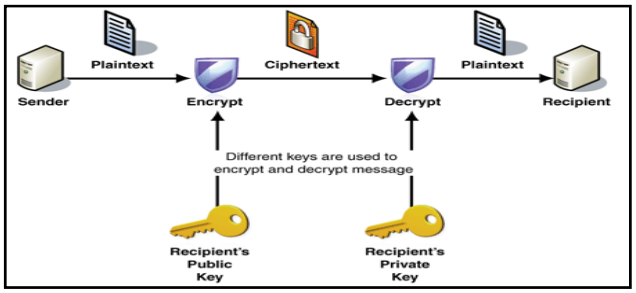

FIGURE 1: PUBLIC-KEY CRYPTOGRAPHY PROTOCOL

One of these keys is published or public and the other is kept private. If the lock/encryption key is the one published then the system enables private communication from the public to the unlocking key's owner. If the unlock/decryption key is the one published then the system serves as a signature verifier of documents locked by the owner of the private key.

In the Diffie–Hellman key exchange scheme, each party generates a public/private key pair and distributes the public key... After obtaining an authentic copy of each other's public keys, *SENDER* and *RECIPIENT* can compute a shared secret offline.

An analogy that can be used to understand the advantages of an asymmetric system is to imagine two people, *SENDER and RECIPIENT*, sending a secret message through the public mail. In this example, *SENDER* wants to send a secret message to *RECIPIENT*, and expects a secret reply from *RECIPIENT*.

With a *Symmetric-key* system, *SENDER* first puts the secret message in a box, and locks the box using a *padlock* to which he has a key. He then sends the box to *RECIPIENT* through regular mail. When *RECIPIENT* receives the box, he uses an identical copy of *SENDER*'s key (which he has somehow obtained previously, maybe by a face-to-face meeting) to open the box, and reads the message. *RECIPIENT* can then use the same padlock to send his secret reply.

In an *Asymmetric-key* system, *RECIPIENT* and *SENDER* have separate padlocks. First, *SENDER* asks *RECIPIENT* to send his open padlock to him through regular mail, keeping his key to himself. When *SENDER* receives it he uses it to lock a box containing his message, and sends the locked box to *RECIPIENT*. *RECIPIENT* can then unlock the box with his key and reads the message from *SENDER*. To reply, *RECIPIENT* must similarly get *SENDER*'s open padlock to lock the box before sending it back to her.

The critical advantage in an asymmetric key system is that *RECIPIENT* and *SENDER* never need to send a copy of their keys to each other. This prevents a third party (perhaps, in the example, a corrupt postal worker) from copying a key while it is in transit, allowing said third party to spy on all future messages sent between *SENDER* and *RECIPIENT*. So in the public key scenario, *SENDER* and *RECIPIENT* need not trust the postal service as much. In addition, if *RECIPIENT* was careless and allowed someone else to copy *his* key, *SENDER*'s messages to *RECIPIENT* would be compromised, but *SENDER*'s messages to other people would remain secret, since the other people would be providing different padlocks for *SENDER* to use.

Public key exchange cryptosystem eliminates the key distribution problem by using two keys, a private and a public key. By exchanging the public keys, both parties can calculate a unique shared key, known only to both of them.

The Diffie-Hellman Algorithm for Key Exchange

SENDER must do the following:

1. Choose a prime numbers p randomly, and choose two integer numbers a and g.
2. Compute the A (*SENDER*'s public key), as follows:

$$A = g^a \bmod p.$$

3. Send the public value A to *RECIPIENT*.
4. Compute the secret value K, as follows: $K = B^a \bmod p$.

RECIPIENT must do the following:

1. Choose an integer numbers b randomly.
2. Compute the B (*RECIPIENT*'s public-key), as follows:

$$B = g^b \bmod p.$$

3. Send the public value B to *SENDER*.
4. Compute the secret value K, as follows: $K = A^b \bmod p$.

SENDER				RECIPIENT		
Secret	Public	Calculates	Sends	Calculates	Public	Secret
a	p, g		p,g →			b
a	p, g, A	$g^a \bmod p = A$	A →		p, g	b
a	p, g, A		← B	$g^b \bmod p = B$	p, g, A, B	b
a, **s**	p, g, A, B	$B^a \bmod p = s$		$A^b \bmod p = s$	p, g, A, B	b, **s**

TABLE 1: DIFFIE-HELLMAN KEY EXCHANGE PROTOCOL

The security of the Diffie-Hellman key exchange protocol is based on the strength of the discrete algorithm and the size of the key used. However, the Diffie-Hellman protocol is considered secure against brute force attack if *p* and *g* are chosen properly (Diffie, et al., 1976). Currently, the solving of the Diffie-Hellman discrete logarithm problem will make many other public-key cryptosystem insecure. The Diffie-Hellman key exchange has a high level of the security because Diffie-Hellman protocol depends on large prime numbers which exceeds 1024 bit (Diffie, et al., 1976, Elaine et al., 2006).

2.2 Steganography:

Steganography is the science of writing hidden messages to guarantee information which is accessible only by authorized parties, and to the one has the secret key and is inaccessible to others. It is the practice of hiding information usually text messages, inside other file (host file). This practice of hiding information (steganography) is normally called stego. Steganographic systems can be grouped by the type of covers used (graphics, sound, text, executables) or by the techniques used to modify the covers such as insertion, substitution, transform domain techniques, spread spectrum techniques, statistical method, distortion techniques, and cover generation methods. Information can be hidden (or embedded) inside any type of multimedia file; image files are the most widely used today. The host files can then be exchanged over an insecure medium without anyone knowing what really lies inside of them.

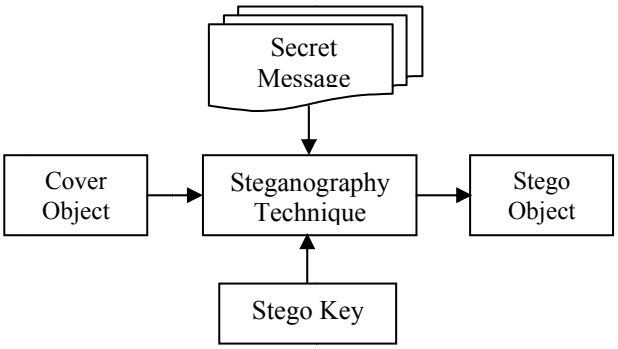

FIGURE 2: STEGANOGRAPHY PROTOCOL

Computer Steganography is basically categorized into two methods; *lossless method* and *loss method*.

Lossless method is a computerized image or sound files can be replaced without losing their functionality. Since the lossless method depends on the inability of human optic to differentiate any changing in image color or sound quality. The 16-bit sound file and 24-bit map (BMP) image file are typically used for steganographic purposes. Image and sound files are considered a well known carrier media candidates for hiding information that ease the operations on steganography. Therefore, we are focusing on 24-bit image file in our proposed method for this study. However there are many proposed steganography methods among them one of the popular methods is *Least Significant Bit.*

Least significant Bit (LSB) insertion is a common, simple approach to embedding information in a cover image[1]. The least significant bit of some or all of the bytes inside an image is changed to a bit of the secret message. When using a 24-bit image, a bit of each of the red, green and blue colour components can be used, since they are each represented by a byte. In other words, one can store 3 bits in each pixel. An 512 × 512 pixel image, can thus store a total amount of 7,86,432 (512x512x3) bits or 98304 bytes of embedded data [2]. For example a grid for 3 pixels of a 24-bit image can be as follows:

(00101101 00011100 11011100)

(10100110 11000100 00001100)

(11010010 10101101 01100011)

When the number 200, which binary representation is 11001000, is embedded into the least significant bits of this part of the image, the resulting grid is as follows:

(00101101 00011101 11011100)

(10100110 11000101 00001100)

(11010010 10101100 01100011)

Although the number was embedded into the first 8 bytes of the grid, only the 3 underlined bits needed to be changed according to the embedded message. On average, only half of the bits in an image will need to be modified to hide a secret message using the maximum cover size [2]. Since there are 256 possible intensities of each primary colour, changing the LSB of a pixel results in small changes in the intensity of the colours. These changes cannot be perceived by the human eye - thus the message is successfully hidden. With a well-chosen image, one can even hide the message in the least as well as second to least significant bit and still not see the difference [1].

In its simplest form, LSB makes use of BMP images, since they use lossless compression. LSB steganography has also been developed for use with other image file formats. The 24-bit image is typically used in LSBs method to hide the secret information.

In the above example, consecutive bytes of the image data – from the first byte to the end of the message – are used to embed the information. This approach is very easy to detect [3]. A slightly more secure system is for the sender and receiver to share a secret key that specifies only certain pixels to be changed. Should an adversary suspect that LSB steganography has been used, he has no way of knowing which pixels to target without the secret key [4]. This process actually needs a secret key that is called *stego-key*. The *stego-key* is used to control the stego process such as the selection of pixels. The selected pixel is then will be used to embed the secret binary information.

3. PROPOSED METHOD

3.1 Public Steganography in various selected regions of an image:

The proposed method describes two steps for hiding the secret information by using the public steganography based on matching method in different regions of an image.

The First step is converting the *Plain text message* into *cipher text* using Public-key Encryption algorithm.

The next step is to find the shared stego-key between the two communication parties (*SENDER & RECIPIENT*) over insecure networks by applying Diffie-Hellman Key exchange protocol (as explained above). At the end the protocol, each side recovers his/her received public key to reach the shared values between them, that's mean *SENDER & RECIPIENT* have arrived same sego-key value.

The next step in the proposed method is that the sender uses the secret stego-key to select pixels that it will be used to hide. Each selected pixel is then used to hide 8 bits binary information depending on the matching method which is summarized in four cases as shown by Table 2. Since the 8 bits data will be compared with the selected pixel's bytes, red, green & blue values respectively to produce an array of binary values as 00, 01, 10, and 11.

SENDER's side, starts comparing to search the equality, where, he takes data value and compare it with the value of the red color (± 7 – decimal value). As shown by Table 2, case no. 1, if they are equal, then the value zero (00 – binary value) is set to the array. Table 2, case no. 2, if the data value and the red value are not equivalent then the value will be compared with the green color, if they are equals (± 7 – decimal value) then the array is set to be one (01- binary value). Table 2, case no. 3, if the data value and the green value are not equivalent then the value will be compared with the blue color, if they are equals (± 7 – decimal value) then the value two (10 – binary value) is set to the array. Finally (refer to Table 2, case no. 4), If in case the secret data didn't equal any of the previous three conditions then the LSBs method is used to embed the data inside the selected pixel, and the value three (11 – binary value) is set to the array. In this case, the data value will be distributed as follows:

1. The first three bits of the data are replaced by the three least significant bits of the red byte.
2. The second three data bits are replaced by the three least significant bits of the green byte.
3. The last two data bits are replaced by the two least significant bits of the blue byte.

Case 1	If 8 bit data ≈ Red (8 bit)	Then Red value = 8 bit data	00
Case 2	If 8 bit data ≈ Green (8 bit)	Then Green value = 8 bit data	01
Case 3	If 8 bit data ≈ Blue (8 bit)	Then Blue value = 8 bit data	10
Case 4	Otherwise	Use LSBs Method	11

TABLE 2: THE FOUR MAIN CASES IN THE PROPOSED PUBLIC-KEY STEGO

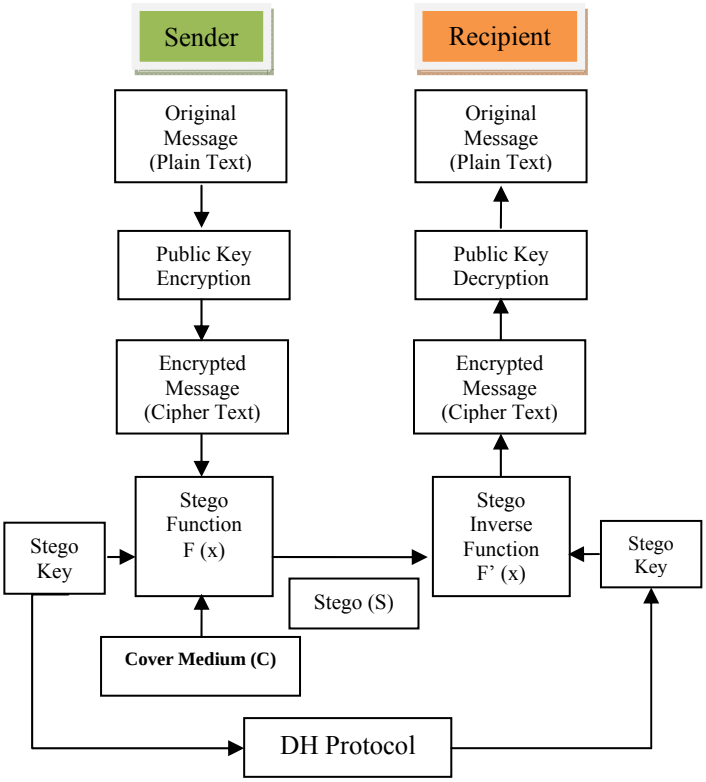

FIGURE 3: PROPOSED METHOD WORKING PROCESS

3.2 Problems and Possible solutions

Having stated that LSB insertion is good for steganography, we can try to improve one of its major drawbacks: the ease of extraction. We don't want that a malicious attacker be able to read everything we are sending.

This is usually accomplished with two complementary techniques:

- Encryption of the message, so that who extracts it must also decrypt it before it makes sense
- Randomizing the placement of the bits using a cryptographical random function (scattering), so that it's almost impossible to rebuild the message without knowing the seed for the random function.

In this way, the message is protected by two different keys, acquiring much more confidentiality than before. This approach protects also the integrity of the message, being much more difficult (we could say at least computationally infeasible) to counterfeit the message.

The two most important issues in these problems are:

- the choice of images
- the choice of the format (24-bit or 8-bit, compressed or not)

The cover image first of all must seem casual, so it must be chosen between a set of subjects that can have a reason to be exchanged between the source and the receiver. Then it must have quite varying colors, it must be "noisy", so that the added noise is going to be covered by the already present one. Wide solid-color areas magnify very much any little amount of noise added to them.

Second, there is a problem with the file size that involves the choice of the format. Unusually big files exchanged between two peers, in fact, are likely to arise suspicion. Since we need to have small image file sizes, we should resort in using 24-bit images, because their size is more likely to be considered as normal.

FIGURE 4: THE RESULT OF EMBEDDING THE TEXT WITH S-TOOLS

4. RESULTS AND DISCUSSION

We implemented the public-key steganography based on matching method in different selected regions of an image to show the performance of the proposed method.

In our implementation, we used 600×400 bitmap image file to hide 5 KB text data. As discussed earlier, both of the two communication parties should find the secret key (stegokey) first by applying Diffie-Hellman public-key exchange protocol to perform high level of security.

As in Table 2, the 8 bits data will be hidden inside 1 pixel, hence the 600x400, 24 bit image file can accept approximately 240000 bytes of data. This is compared with well known stego method such as LSBs (Johnson et al., 1998) which needs 3 pixels to hide 1 byte of data. We can also adjust the bit-rate at which we can hide the data in the selected region. Nevertheless, the proposed steganographic protocol is more efficient than LSBs, since the algorithm used the matching method to get identical pixel's bytes. However, the proposed method resorts to the LSBs method to distribute the secret data in case if the 8 bit of data is not matched with any of the previous three bytes (red, green, and blue).

CONCLUSION

Steganography is a really interesting subject and outside of the mainstream cryptography and system administration that most of us deal with day after day. As steganography becomes more widely used in computing there are issues that need to be resolved. There are a wide variety of different techniques with their own advantages and disadvantages.

This paper has shown the possibility of using matching method in different regions of an image in public steganographic protocol. The security of the proposed steganography depends on Encryption algorithm and Diffie-Hellman public key exchange protocol. We can add more security to the information which needs to be transmitted through public networks using this method.

ACKNOWLEDGMENTS

I thank Dr. D. Srinivasa Kumar, for his helpful feedback on this work. I would also like to thank my family and friends who encouraged me in doing this work.

REFERENCES

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

[2] Krenn, R., "Steganography and Steganalysis", http://www.krenn.nl/univ/cry/steg/article.pdf (19)

[3] Wang, H & Wang, S, "Cyber warfare: Steganography vs. Steganalysis", *Communications of the ACM*, 47:10, October 2004

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

[5] Cryptography and Network Security Principles and Practices, 4th edition by William Stallings.