RSA Encrypted Image Steganography

Hiding Data Within An Image File

Ali Alwan
Department of Electrical and Computer Engineering
Villanova University
Villanova, PA

***Abstract*—Steganography is a method of hiding secret messages in a cover object to communicate between a sender and a receiver. The security of confidential information has always been a major issue to prevent unauthorized access. Steganography is a technique used to fulfill a secure transfer of data. In this paper, we proposed using LSB technique with RSA algorithm for image steganography to provide more security for the transfer of data. This technique involves hiding the data bits into the LSB of RGB pixel values of a cover image. This technique encrypts the data before hiding it into the cover image. In the case that there is a malicious user that accesses the cover image during its transmission, only the ciphertext can be extracted from the image, allowing only the intended user to have access to the data being transferred.**

***Index Terms*—Steganography, Cryptography, LSB technique, RSA Encryption - Decryption**

I. Introduction

The term steganography is derived from the Greek word *steganos*, meaning hidden or covered. Steganography is the practice of concealing a message to hide its existence from unwanted eyes. Steganography helps obscure the fact that there is sensitive data hidden in the file or other content carrying the hidden message. The purpose of steganography is to conceal and deceive. It is a form of covert communication that can use any medium to hide messages.

Today, steganography is significantly more sophisticated allowing a user to disguise massive amounts of data within various computer files. These types of steganography are often used in conjunction with cryptography so that the information is doubly protected; first the data is encrypted and then hidden so that an adversary has to first discover the data and then decrypt the extracted ciphertext.

II. History of Steganography

Forms of steganography has been widely used for centuries. The first recorded use of steganography can be traced back to 440 BC in Greece, in which Histiaeus, the ruler of Milteus, tattooed a message on the shaved head of one of his slaves and had the hair grow back. He then sent the slave to his son-in-law, who shaved the slave’s head revealing the message.

In ancient times, steganography was mostly done physically, such as using invisible ink or hiding documents using microdot. In the centuries that followed, more modern forms of steganography were created. Through the massive advancement in technology, steganography has moved to the digital world.

III. How Steganography Is Used Today

Steganography primary usage is to create private and secure communication between a sender and receiver. However, there are a number of uses for steganography besides just establishing a covert communication channel. One of the most widely used applications of steganography is digital watermarking. A watermark is an identifying image or computer file to provide authenticity. A digital watermark is the replication of an image, text, or any other document with some hidden trademark so that the source of the document can be properly authenticated.

However, steganography can also be used maliciously. Hackers use steganography to hide malware or viruses within a computer file to attack a victim’s computer. The victim is not aware of the malicious software as it is hidden in an ordinary file but gets executed once it enters the victim’s system.

IV. Types of Steganography Techniques

In today’s digital world, steganography can use various different techniques to hide data within a digital file. For each different digital media source, a different technique can be used to embed the information into the file. There are five different types that steganography can use as a medium to transport information. The five cover objects that we’ll discuss are image files, text files, audio files, video files, and within a network or protocol.

A. Image Steganography

Hiding the data by taking the cover object as an image is known as image steganography. Embedding information within an image file is an excellent type of steganography as it has a high level of redundancy. There are a huge number of bits present in the digital representation of an image making it possible to change few of the bits without being noticeable. The secret message is embedded in the image by changing the RGB pixel values of an image. The most common technique is to use LSB-technique as this only changes the least significant bit of a pixel, making it undetectable to the human eye, which can be seen in Figure 1 below. However, this only work for lossless image file formats, such as PNG, RAW, BMP, etc. For these image file formats, no bits of information are lost during compression. In lossy compression file formats, such as JPEG, redundant bits are lost during compression making it possible for the embedded information to get altered. A way to avoid this issue is by implementing frequency domain technique, so during compression the bits of the message are not lost. This is done by using cosine transformation as a technique for lossy image file formats.

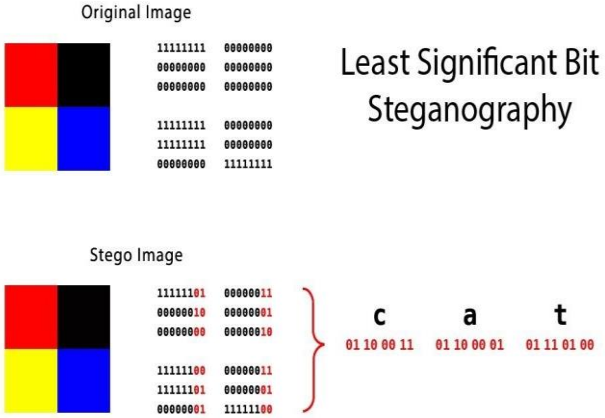


Figure 1: LSB-technique within an image file

B. Text Steganography

The process of hiding information within text files is known as text steganography. This technique involves a variety of different methods such as changing the format of existing text, changing words within a text, generating random number sequences, or using context-free grammars to generate readable texts. Text steganography can be separated into three different methods. Format based methods involves altering physically the format of text to conceal the information, but this method is not that secure as it is possible for these changes to become detected. Random and statistical generation methods involve the concealing of information in random looking sequences of characters, or the statistical properties of word length and letter frequencies are used to create the embedded message. Linguistic methods consider the linguistic properties of generated and modified text and uses linguistic structure as the space in which messages are embedded.

C. Audio Steganography

In audio steganography, the private message is embedded into an audio signal which alters the binary sequence of the corresponding audio file. The process of hiding information in digital signals is a much more complicated process when compared to the other different steganography techniques. One common method to implementing audio steganography is by using least significant bit encoding. This is similar to image steganography as the bits of information is encoded into the bits of the audio signals, which can be seen in Figure 2 below. Other methods to audio steganography include parity encoding, phase encoding, and spread spectrum. As audio steganography slightly alters the original audio file, the human ear is more sensitive to these slight changes, increasing the chance of the added noise being noticed.

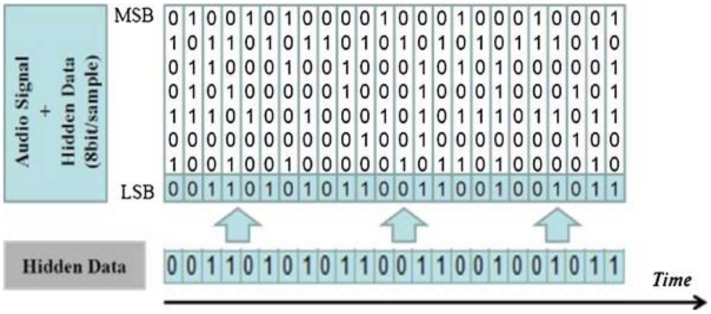


Figure 2: LSB-technique within an audio file

D. Video Steganography

The process of hiding data within a digital video format is known as video steganography. A big advantage of this technique is that it is able to embed a large amount of data when compared to the other techniques, and this is due to the file size of a video being significantly bigger than other digital files. Video steganography can be seen as a combination of image and audio steganography since a video is a series of images along with an audio signals. There are two different methods of embedding data within a video file. The first method is to embed the information into the uncompressed raw video and then compressing it later. Whereas the second method is to embed the information directly into the compressed data stream of the video.

E. Network Steganography

Lastly, network steganography is the technique of embedding information within network control protocols used in data transmission. Data can be secretly hidden in protocols like TCP, UDP, ICMP, and other protocols. It is possible to use steganography in some covert channels within the OSI model. In Figure 3 below, it shows different examples of how data can be secretly embedded in the different protocol layers throughout the OSI model. One example of network steganography is by hiding information into the header of a TCP/IP packet is some fields that are not essential.

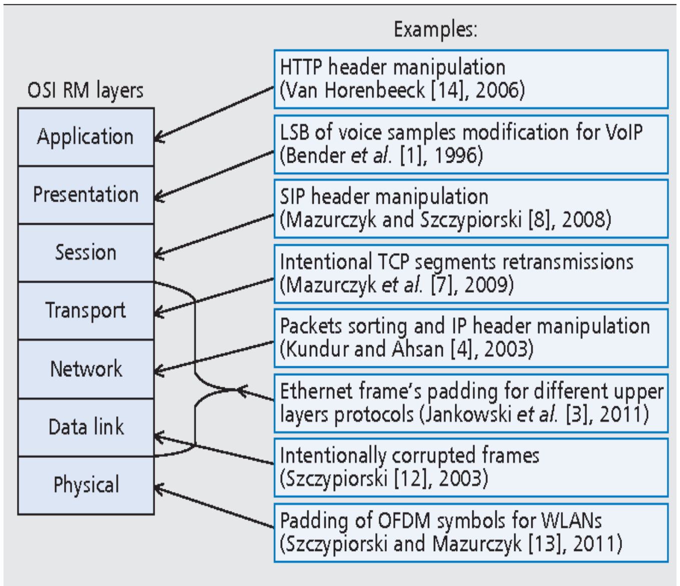

Examples:
OSI RM layers
Application
Presentation
Session
Transport
Network
Data link
Physical
HTTP header manipulation (Van Horenbeeck [14], 2006)
LSB of voice samples modification for VoIP (Bender *et al.* [1], 1996)
SIP header manipulation (Mazurczyk and Szczypiorski [8], 2008)
Intentional TCP segments retransmissions (Mazurczyk *et al.* [7], 2009)
Packets sorting and IP header manipulation (Kundur and Ahsan [4], 2003)
Ethernet frame's padding for different upper layers protocols (Jankowski *et al.* [3], 2011)
Intentionally corrupted frames (Szczypiorski [12], 2003)
Padding of OFDM symbols for WLANs (Szczypiorski and Mazurczyk [13], 2011)

Figure 3: Network steganography within the OSI model

V. Cryptography vs. Steganography

Cryptography and steganography are both used to protect information from unauthorized access. However, they go about this in different ways. Cryptography changes the information into ciphertext, which is unreadable data without a decryption key. So if someone is able to intercept the data in transmission, they be able to see that the message has been encrypted to prevent intruders from reading the contents. On the other hand, steganography does not change the format of the information being transmitted but conceals the existence of the information. The data is always visible when using cryptography but never visible using steganography. A key is essential for cryptography whereas in steganography, it is not required but offers extra security if used. As steganography can use any sort of digital media as a carrier as previously discussed, cryptography is usually text based. There are key differences between cryptography and steganography, however, if used together it can provide a great amount of security to protect information. They can be implemented together by first applying the cryptography algorithm and then embedding the ciphertext into a cover medium, it can then be extracted and decrypted by only the intended receiver, which can be seen in Figure 4 below.

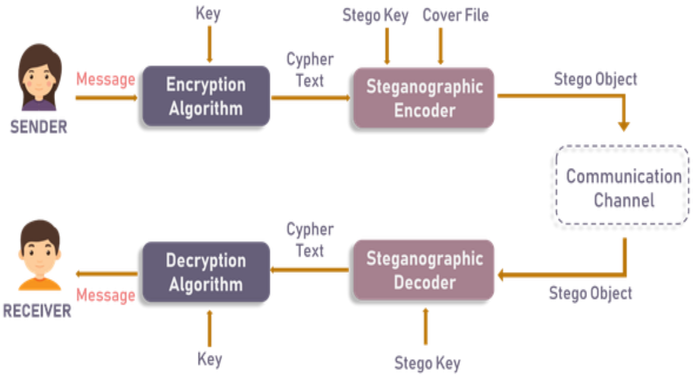


Figure 4: Framework on cryptography and steganography

VI. Steganalysis

Steganalysis is the study of detecting hidden messages using steganography. The goal of steganalysis is to identify if a digital media file is a steganographic file or just an ordinary file. In Figure 5 below, it illustrates how steganalysis can be used to determine if a suspected stego object has hidden information within it or it is just a cover object with not data embedded in it, and there are two different methods to steganalysis: specific and universal steganalysis.

The most common technique is to perform statistical steganalysis as this analyzes the underlying statistics of a digital media file. The statistics on a file undergoes changes when embedded with the secret information, making it possible to analyze these changes to determine Specific steganalysis is established by analyzing the embedding technique used and determining certain statistics. This method of steganalysis requires a detailed knowledge of the embedding process of the information, allowing this method to provide very accurate results. Universal steganalysis is not tailored towards a specific embedding process, so it requires no prior knowledge on the steganographic technique used to be able to detect a secret message hidden within a file. Therefore, steganalysis is the technology that attempts to defeat steganography by detecting the secret information embedded within a digital media file.

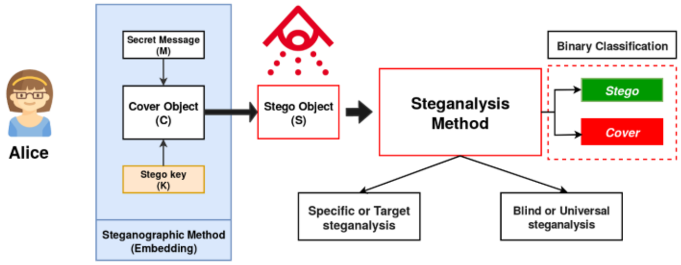


Figure 5: Steganalysis method

VII. Technical Part

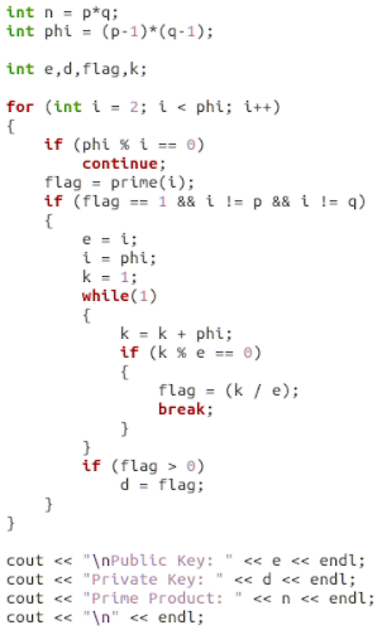
The software program demonstrated below creates steganographic images through LSB encoding and implements RSA encryption to increase security. There are three main phases to this program that will further be discussed: key generation, encryption and LSB encoding, and LSB decoding and decryption.

A. Key Generation

The intended receiver of the secret information uses the key generation phase to create a public key that can be used by others to send a message. In this example demonstrated through Figure 6 below, the RSA algorithm key generation procedure can be explained as follows:

(i) Select two large strong prime numbers, p and q. Let n = p q.
(ii) Compute Euler's totient value for n: phi (n) = (p - 1) (q - 1).
(iii) Find a random number e satisfying 1 < e < phi (n) and relatively prime to phi (n) i.e., gcd (e, phi (n)) = 1.
(iv) Calculate a number d such that de = 1 mod phi (n).

This phase creates the public and private key that can be used by the sender and receiver to transmit secure, encrypted information.

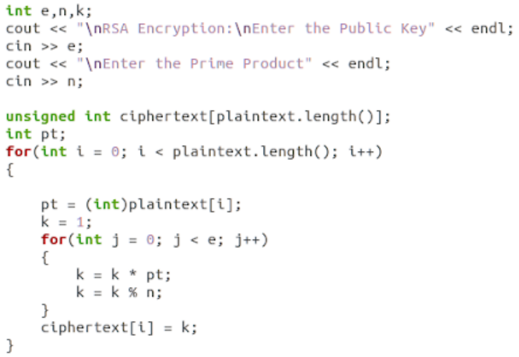


```
int n = p*q;
int phi = (p-1)*(q-1);

int e,d,flag,k;

for (int i = 2; i < phi; i++)
{
    if (phi % i == 0)
        continue;
    flag = prime(i);
    if (flag == 1 && i != p && i != q)
    {
        e = i;
        i = phi;
        k = 1;
        while(1)
        {
            k = k + phi;
            if (k % e == 0)
            {
                flag = (k / e);
                break;
            }
        }
        if (flag > 0)
            d = flag;
    }
}

cout << "\nPublic Key: " << e << endl;
cout << "Private Key: " << d << endl;
cout << "Prime Product: " << n << endl;
cout << "\n" << endl;
```

Figure 6: Key Generation Phase

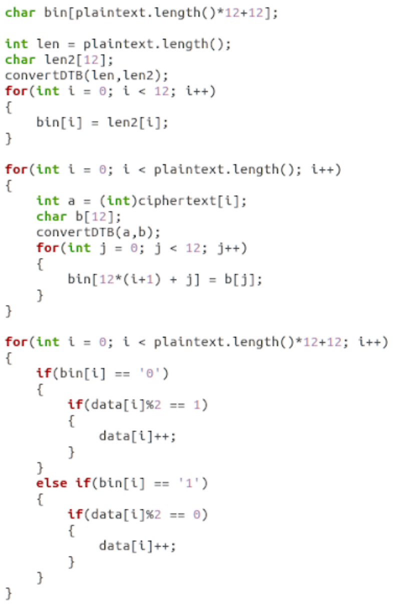
B. RSA Encryption and LSB Encoding

A steganographic image is created by encoding the hidden information into a cover image file. The secret message gets embedded in the image file as ciphertext from RSA encryption using the generated public key. Demonstrated in Figure 7 below, the plaintext of the message converts into ciphertext by using the encryption algorithm: c = m^e mod n, where c is the ciphertext and m is the plaintext. The ciphertext gets encoded by adjusting the least-significant bit of the image pixels. The least significant bits of each RGB pixel value are altered to represent the binary equivalent of the ciphertext, which can be seen in Figure 8 below. Additionally, the first byte of the embedded information is the length of the secret message to determine the number of bits needed to be read during the decoding phase. So this phase of the program is able to successfully save steganographic images to memory.



```
int e,n,k;
cout << "\nRSA Encryption:\nEnter the Public Key" << endl;
cin >> e;
cout << "\nEnter the Prime Product" << endl;
cin >> n;

unsigned int ciphertext[plaintext.length()];
int pt;
for(int i = 0; i < plaintext.length(); i++)
{

    pt = (int)plaintext[i];
    k = 1;
    for(int j = 0; j < e; j++)
    {
        k = k * pt;
        k = k % n;
    }
    ciphertext[i] = k;
}
```

Figure 7: RSA encryption



```
char bin[plaintext.length()*12+12];

int len = plaintext.length();
char len2[12];
convertDTB(len,len2);
for(int i = 0; i < 12; i++)
{
    bin[i] = len2[i];
}

for(int i = 0; i < plaintext.length(); i++)
{
    int a = (int)ciphertext[i];
    char b[12];
    convertDTB(a,b);
    for(int j = 0; j < 12; j++)
    {
        bin[12*(i+1) + j] = b[j];
    }
}

for(int i = 0; i < plaintext.length()*12+12; i++)
{
    if(bin[i] == '0')
    {
        if(data[i]%2 == 1)
        {
            data[i]++;
        }
    }
    else if(bin[i] == '1')
    {
        if(data[i]%2 == 0)
        {
            data[i]++;
        }
    }
}
```

Figure 8: LSB encoding

C. LSB Decoding and RSA Decryption

During the decoding process, the least-significant bits of the steganographic image are read to extract the embedded ciphertext in binary form, which is then converted into decimal form. The ciphertext is decrypted using RSA algorithm if the appropriate private key is entered into the program, allowing the secret message to be displayed to the authorized receiver. This demonstrated software program uses a combination of steganography and cryptography to ensure security of a message within a covert communication channel. Therefore, this technical work is able to implement RSA cryptography in the creation of steganographic images.

VIII. Conclusion

Steganography is the practice of concealing a message to hide its existence from outside users. Cryptography is the practice of modifying a message to become unreadable to outside users without a secret key. However, the usage of steganography and cryptography together is a robust technique to ensure security of private information. In this work, the private information gets embedded in an image file as ciphertext using RSA algorithm. The information is encoded into the RGB pixel values of an image using the LSB technique. Therefore, implementing cryptography within steganographic images ensures maximum security of a message by hiding the meaning and existence of it.

References

[1] O. Elharrouss, N. Almaadeed and S. Al-Maadeed, "An image steganography approach based on k-least significant bits (k-LSB)," *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*, 2020, pp. 131-135.

[2] K. C. Nunna and R. Marapareddy, "Secure Data Transfer Through Internet Using Cryptography and Image Steganography," *2020 SoutheastCon*, 2020, pp. 1-5.

[3] K. Thangadurai and G. Sudha Devi, "An analysis of LSB based image steganography techniques," *2014 International Conference on Computer Communication and Informatics*, 2014, pp. 1-4.

[4] N. Subramanian, O. Elharrouss, S. Al-Maadeed and A. Bouridane, "Image Steganography: A Review of the Recent Advances," in IEEE Access, vol. 9, pp. 23409-23423, 2021, doi: 10.1109/ACCESS.2021.3053998.

[5] A. Gutub, A. Al-Qahtani and A. Tabakh, "Triple-A: Secure RGB image steganography based on randomization," 2009 *IEEE/ACS International Conference on Computer Systems and Applications*, 2009, pp. 400-403.

[6] Stanger, J. (2021, November 2). *The ancient practice of steganography: What is it, how is it used and why do cybersecurity pros need to understand it*. Default. Retrieved December 9, 2021, from https://www.comptia.org/blog/what-is-steganography#:~:text=Steganography%20is%20the%20practice%20of,to%20friends%20using%20invisible%20ink.&text=This%20revealed%20the%20secret%20message%20I%20wanted%20to%20share.

[7] Dickson, B. (2020, February 6). *What is steganography? A complete guide to the ancient art of concealing messages*. The Daily Swig | Cybersecurity news and views. Retrieved December 11, 2021, from https://portswigger.net/daily-swig/what-is-steganography-a-complete-guide-to-the-ancient-art-of-concealing-messages.

[8] Says, M., & Mahor. (2018, August 30). *Difference between steganography and cryptography (with comparison chart)*. Tech Differences. Retrieved December 11, 2021, from https://techdifferences.com/difference-between-steganography-and-cryptography.html.

[9] *Image steganography in cryptography*. GeeksforGeeks. (2021, October 26). Retrieved December 11, 2021, from https://www.geeksforgeeks.org/image-steganography-in-cryptography/.

[10] *RSA algorithm in cryptography*. GeeksforGeeks. (2021, January 5). Retrieved December 12, 2021, from https://www.geeksforgeeks.org/rsa-algorithm-cryptography/.

[11] Paladion. (n.d.). *Steganalysis*. Protecting your business to face cybersecurity challenges. Retrieved December 11, 2021, from https://www.paladion.net/blogs/steganalysis.