Maya Hussein and Nada Badawi

CSCE 2022-2

December 2021

# Steganography

*"The government doesn't want any system of transmitting information to remain unbroken, unless it's under its own control."*

Isaac Asimov

**Abstract.**

*Steganography* is the art of concealing data within other files. This paper discusses the technology that steganography presents with its five major types. Its focus is shifted on image steganography and presents its implementation using the Least Significant Bit Algorithm and the Fibonacci Sequence. The paper also provides an overview of steganographies widespread applications and an analysis of the algorithm presented.

# 1 Introduction

STEGANOGRAPHY is the process of manipulating files and data in which a hidden message is inserted into it. It is originally derived from the term *steganographia* which means "concealed writing" in Greek [2]. Essentially it hides the message to appear to be as part of the data sent, i.e. to be in the form of thin lines of an image or visible dots of a letter. Steganographic messages aren't necessarily secretive; they can be stored as invisible inks, character arrangements, microdots, covert channels, and digital signatures [3]. Unlike cryptography, which is the practice of encoding messages using a secret key, steganography focuses on repulsing the attention as a methodology for disrupting the user. It also rejects the notion of "security through obscurity." Steganography simply applies Kerchoff's Principle. It was originally rooted to 440 BC in Greece when Herodotus, an ancient Greek writer and philosopher uses this technique in sending his message during the Ionian Revolt [2]. Likewise, as technologies advanced and secrecy required a complex level of transitions, Steganography was welcomed into the digital world.

Nowadays, sending a hidden message inside a file or an image is quite easy because in electronic communications, the data is usually implemented inside of a "transport layer" namely media files, programs, or protocols. There are five major types of Steganography: (1) Text Steganography, (2) Image Steganography, (3) Video Steganography, (4) Audio Steganography, and (5) Network Steganography. Typically, the larger the file size, the more subtle the message is concealed [3]. For instance, the recipient might receive an overwhelming large image file with colors and pixels slightly adjusted corresponding to a letter in the alphabet. The change in tones and gradients goes unnoticeable for the naked eyes immediately recognize. Hence, Steganography, although a threatening tool, has aided in several technological facilities, and advancements in general.

## 2 Motivation

Nowadays, with billions of messages being sent to and from end-to-end encrypted sources. Is it possible that Steganography can be easily implemented; if so, how can we recognize it?

## 3 Topic Definition

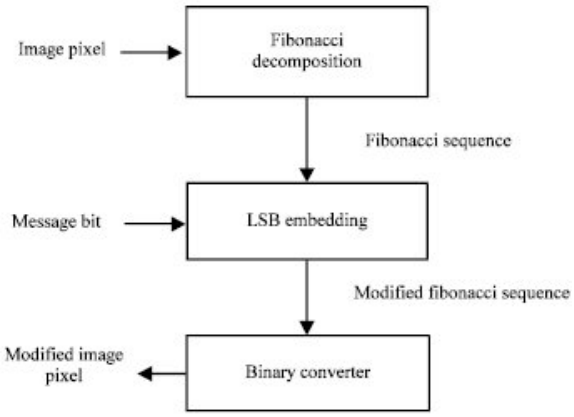
"*Steganography* is a method of hiding secret data, by embedding it into an audio, video, image, or text file. It is one of the methods employed to protect secret or sensitive data from malicious attacks" [1].

## 4 Literature Survey of Methodologies and Algorithms

Steganographs can be implemented using various techniques and algorithms; on simpler terms, it can be implemented using the Fibonacci Algorithm. However, it must be represented using pixel's value, a binary domain representation of 8-bit planes. The Fibonacci Algorithm allows for steganographs applications in low resolution images.

In Fibonacci domain, the pixel's value allows the availability of more bit-planes to embed secret bits into the media file used with significant degradation [1].

The figure below illustrates the sequence of Fibonacci in steganographizing content. During the first step the media file chosen must be iterated over from its current pixel value to its Fibonacci Representation, where 1 secret bit is embedded into the sequence by the Least Bit Algorithm (LSB) [4]. The LSB Algorithm changes the 8th bit of all bytes inside an image into a the bit of the




secret message. The Fibonacci sequence then converts the bits into binary digits before writing into the new image.

The algorithm that hides secret bits into higher LSB layers of image pixel color component in a Fibonacci domain in order to improve the robustness and security of hidden data embedded into the stego-image [4]. Below is a summary of the steps required to steganographize a media file:

1. User inserts password used in key stream
2. LSB algorithm takes last bit and generates key stream in the first step using represented into Fibonacci representation. The secret bits are embedded into a last LSB layer of decomposed image pixel color component.
3. The new image is then serviced and channeled for download with concealed image
4. Once downloaded, a private key is sent to the receiver to produce data for the private key for creating the key stream. The key stream is then generated to decipher the color detections.

Below is a sample of the Steganography Algorithm using LSB and Fibonacci Sequence:

**Algorithm 1** Steganography using LSB and Fibonnaci Sequence

```
#Checks whether the bit is set or not at a particular position.
isBitSet(char ch, int pos):
    ch = ch >> pos
    if ch & 1   return true
    return false

main(int argc, char** argv):
    if argc != 4
        print "Arguments Error"
        exit program

    image = imread(argv[1])

   #stores original image
    if image is empty
        print "image error!"

    opens file for text information
    and reads the first character

    file.get(ch)
    bit_count = 0

    #Checks if message is encoded or not
    encoded = false

    # To hide text into images. each of the 8 bits are
    stored in the LSB of the pixel values (Red,Green,Blue).

    for(row = 0 -> image.rows)
        for(col = 0 -> image.cols)
            for(color = 0 -> 2)

                #stores the pixel details
                Vec3b pixel = image.at<Vec3b>(Point(row,col))

                if bit is 1 :
                    change LSB of present color value to 1.
                if bit is 0 :
                    change LSB of present color value to 0.
```

```
                #update the image with the changed pixel values
                image.at<Vec3b>(Point(row,col)) = pixel

                #increment bit_count to work on next bit
                bit_count++;

                if (last_null_char is true && bit_count is 8)
                    encoded = true

                if(bit_count == 8)
                    bit_count = 0
                    get next character from file

                if(end of file)
                    last_null_char = true;
                    ch = '\0'
    #whole message was not encoded
    if(!encoded)
        print "Message too big. Try with larger image.\n"
        exit(-1);

    #Write the steganographic image
    imwrite(argv[3],image);
```

# 5 Applications

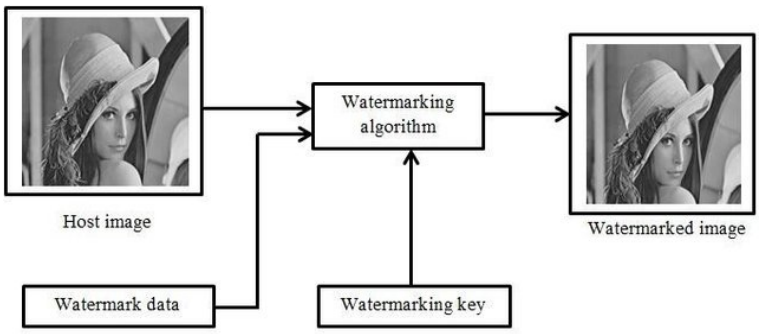
Steganography is used in various ways and has many applications. One of those uses is data protection and secrecy. Messages and information can be transferred from the sender to the receiver without a third party recognizing anything odd about the message, which is why it may be preferred over cryptography. This is extremely helpful when it comes to data security systems and passing military data [7]. Therefore, steganography is widely used in military communication.

Another important use of steganography is the embedding of data in images in media database systems [5]. To illustrate, images have the date in which the photo was taken or the image was saved embedded. Such information helps media database systems to sort the images and give any related information a user may need.

Steganography is also used in watermarking data, such as videos and images. Watermarking is commonly used in preserve the copyrights of the owner if the content. This can be achieved by fingerprinting [6]. Fingerprinting is identifying and tracking data across the network to know who accessed what exactly [8]. The information collected by data fingerprinting can be collected and embedded in the original data so the owner would be able to track and learn who accessed or shared their content illegally.

# 6 Algorithm Analysis

The algorithm mentioned earlier does not require high space complexity since it utilizes a one-dimensional array to store the information needed for the embedding of data in images. For the time complexity, the algorithm has several sequential selective statements, which makes the worst-case complexity




be O(1). The algorithm also uses three nested for loops to store and change the pixel details using more selective statements. This makes the worst-case complexity to rise to O($n^3$). The algorithm trades off the time complexity for the space complexity.

# 7 Conclusion

Overall, the paper discusses the importance and feasibility of the steganography technology. It has five different kinds: Text, Image, Video, Audio, and Network. Steganography can be used for a wide range of reasons, one of which is fingerprinting and preserving copyrights, military communication, and media database systems. So, it can be used in data security systems or in everyday life for easiness and accessibility.

# References


[1] AnuIyer.*Image Steganography in Cryptography.* Geeks for Geeks, 2021.

[2] David Kahn *History of Stegenography.* Great Neck. 2017.

[3] Rosziati Ibrahim & Teoh Suk Kuan *Steganography Algorithm to Hide Secret Message inside an Image.* University Tun Hussein Onn Malaysia, 2011

[4] Tuan Duc Nguyen, Somjit Arch-int & Ngamnij Arch-int *A Secure Steganographic Algorithm Based on Fibonacci Representation Using Cellular Automata.* Research Journal of Information Technology, 2014

[5] Eiji Kawaguchi *Applications of Steganography.* KIT-STEGROUP, 2018.

[6] Frank Y. Shih *Digital Watermarking and Steganography: Fundamentals and Techniques.* Taylor & Francis, 2017.

[7] Aditya Kumar Sahu *Digital image steganography and steganalysis: A journey of the past three decades.* Research Gate, 2020.

[8] Nate Lord *What is File Fingerprinting?.* Digital Guardian, 2018.