Image Steganography with LSB Algorithm

Arsalan Jabbari, B.Sc student in Shiraz University, Iran.
AmirHossein Roodaki, B.Sc student in Shiraz University, Iran.

Abstract

One of the essential kinds of encryption and data hiding is using steganography techniques; Steganography is the study and practice of masking information within objects in such a way that it deceives the spectator as if there is no data hidden within the object. Simply put, it hides information in plain sight so only the intended recipient can see it. In order to satisfy this need, we should prepare a secret key so that only the people with a key can decrypt the code!
There are several types and formations of Steganography. After searching in digital kinds of Steganography, we decide to encrypt a secret key which is a text in an image, using LSB Algorithm.

Keywords

Data Encryption, Digital Steganography, Image Steganography, Least Significant Bit, LSB Algorithm.

Introduction

Before implementing, let us review where this technique and algorithm came from and how they can help us reach our goal of encrypting data in an image. The word Steganography came from the Greek words stegos, meaning cover, and grafia, meaning writing. In image steganography, the information is hidden exclusively in images. The most common and popular algorithm to solve modern-day Steganography is to use LSB (least significant bit) of the picture's pixel information. This technique works best when the file is longer than the message file.
The difference between Steganography and Cryptography is that cryptography concentrates on keeping the contents of a message secret, whereas Steganography concentrates on maintaining the existence of a message secret. Steganography and cryptography are both ways to protect data from undesirable parties. This article describes the LSB algorithm used for the exact purpose of image steganography. We will discuss how it works, this algorithm's advantages, and its implementation in upcoming sections.

Implementation

To implement this algorithm, we have to do some steps to prepare what our algorithm needs. The first step is reading an image from a file, which we implement in the getPicture function in two dimension list to avoid time and memory complexity in the following steps.
To implement the LSB algorithm, we have to have an encoded image with a pixels format that indicates five elements mention in figure 1 for the exact purpose of each pixel; its row, column, Red color, Green color, and Blue color, which know as RGB.

Figure 1

Let us discover the implementation of the text that we will encrypt in an image. We have to use ASCII code to represent every character of our text. Every character may have eight-bit binary code, and every pixel has six expecting bits for the exact purpose of encrypting data(two for every color of every pixel).
So we have to use the least common multiple of 6 and 8, which is 24 for addressing. So the length of our message, which has to encrypt in 24 bits, calculate as metadata.
Know that we increment these 24 with step 6 to inform the implementation number of reserved bits in every pixel for the exact purpose of encrypting.
Furthermore, we must implement the same for every character conversion to ASCII code with 8 mentioned in Figure 2 for the exact purpose of optimality.

LETTER	ASCII VALUES	BINARY VALUES
A	065	01000001
I	105	01100100
D	100	01101001

Figure 2

Each character of the secret message and each pixel of the base image must convert into binary values. The user has to input the secret message as the password. After inserting a secret message into the cover image file, the resulting encrypted image will send to the receiver through the desired communication channel. While defining the starting point of entrenching LSB, the secret text has first gathered from the user. Our task is to summate secret messages for each character's ASCII value calculation.

Pixels: (00100111 11101001 11001000)
(00100111 11001000 11101001)
(11001000 00100111 11101001)
A: 01000001
Result: (00100110 11101001 11001000)
(00100110 11001000 11101000)
(11001000 00100111 11101001)

Figure 3

Hence the average of the characters' values computation is our task. While substituting the secret message into the LSB of the cover image, the first LSB position has chosen according to the calculated average value of the input of the secret message's characters. Then the substitution processing will continue until the end of the secret message.(overall in Figure 3)

Encryption Steps:

1. Read the Image from files.
2. Encode Image by pixels in 5-element format.
3. Read the text that we are going to encrypt, from files.
4. Encode text in ASCII to start encrypting.
5. Get a password from the user.
6. Encrypt the text using the password.
7. Put data in pixels using the LSB algorithm (Two least significant bits).
8. Create a new Image from the modified pixels.

Decryption Steps:

1. Read the encrypted Image from the files.
2. Verify the receiver user by asking for the password.
3. After passing the verification step, Increment in the least significant bits of RGB indexes of pixels till reaching the end of selected bits for encrypting.
4. Concatenate the bits two by two to extract the secret message.

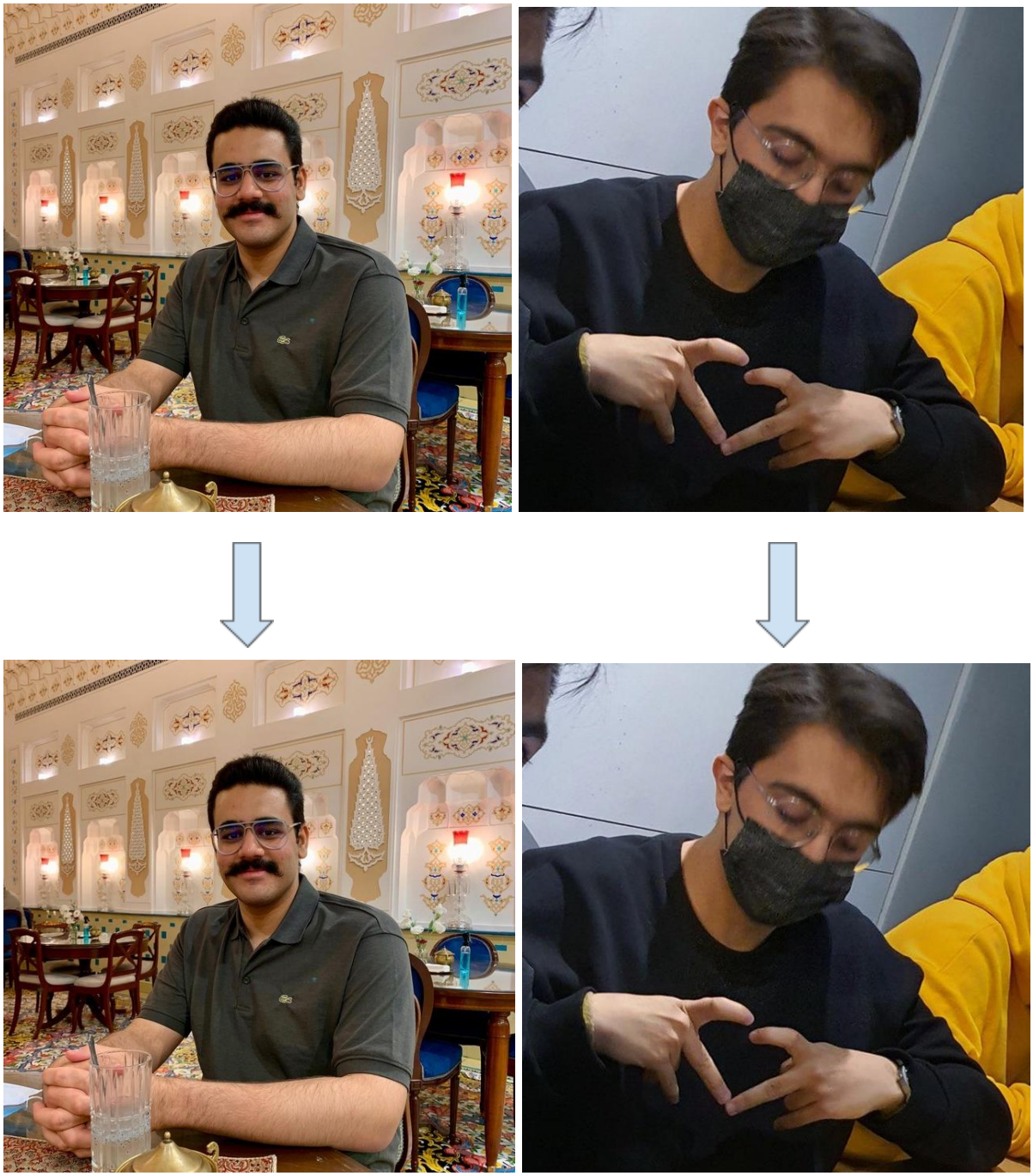

Figure 4

Application

The offered approach in this project uses a steganographic method called image Steganography. The application creates a stego image of personal data embedded and protected with a highly secured password. The project intends to generate a steganographic application that provides good security. The proposed approach provides increased security and can protect the message from stego attacks. The image resolution does not change much and is negligible when we embed the message into the image, and the image has protected with a personal password. So, it is not possible to damage the data of unauthorized personnel. We will use the Least Significant Bit algorithm in this project to develop a faster and more reliable application with a moderate compression ratio compared to other algorithms. The application's primary limitation is its design for the exact purpose of bit map images (.bmp). It accepts only bit map images as a carrier file, and the reduction depends on the document and carrier image size. The security using the Least Significant Bit Algorithm is satisfactory. However, we can improve the level to a certain extent by varying the carriers and using different keys for the exact purpose of encryption and decryption.

Two other technologies that are closely related to Steganography are watermarking and fingerprinting. These technologies are mainly concerned with the protection of intellectual property. In watermarking, all instances of an object are "marked" similarly. The kind of information hidden in objects when watermarking is usually a signature to signify origin or ownership for the exact purpose of copyright protection. With fingerprinting, on the other hand, different, unique marks have embedded in distinct copies of the carrier object supplied to different customers. Enables the owner's intellectual property to identify customers who break their licensing agreement by supplying the property to third parties. This paper describes the LSB algorithm used for image steganography to illustrate the security potential of Steganography for business and personal use.

For the exact purpose of some other applications, we can mention enabling secret communication, complementing regular encryption, which is harder to break and need first to find the encrypted secret text. It needs to be decrypted and has tremendous use in Military Applications.

Resources

[1]

Mano, M. M. (1991). *Digital design*. Prentice-Hall. Retrieved January 21, 2023, from https://books.google.com/books/about/Digital_Design.html?id=bm9smAEACAAJ

[2]

Aggarwal, A., Sangal, A., & Varshney, A. (2019). IMAGE STEGANOGRAPHY USING LSB ALGORITHM. *International Journal of Information Sciences and Application*, *11*(Special), 5. https://www.ripublication.com/irph/ijisaspl2019/ijisav11n1spl_19.pdf. 0974-2255

[3]

Kadam, K., Koshti, A., & Dunghav, P. (2012). Steganography Using Least Significant Bit Algorithm. *International Journal of Engineering Research and Applications (IJERA)*, *2*(3), 4. https://d1wqtxts1xzle7.cloudfront.net/28319421/BF23338341-libre.pdf?1390873939=&response-content-disposition=inline%3B+filename%3DIJERA_www_ijera_com.pdf&Expires=1674299141&Signature=Qf80jTyl3CZwIGUYOBfsefCcKjPMUfoFaO3vY3UKYvQQEQYDaNZkAjoSpBZ3XxeimHzsLF~Mb. 2248-9622