WJCMS

Wasit Journal of Computer and Mathematics Science

Journal Homepage: https://wjcm.uowasit.edu.iq/index.php/WJCM

e-ISSN: 2788-5879 p-ISSN: 2788-5879

Image hiding by using spatial domain steganography

Ghazali Bin Sulong[1,*] and Maria A. Wimmer[2]

[1]Department of Software Engineering Faculty of Computing Universiti Teknologi Malaysia 81310 Skudai, Johor, Malaysia
[2]Faculty of Computer Science, University of Helsinki, Norway

*Corresponding Author: Ghazali Bin Sulong

DOI: https://doi.org/10.31185/wjcm.110
Received: December 2022; Accepted: February 2023; Available online: March 2023

ABSTRACT: This article provides an overview of steganography and its use for hiding images in other images. Steganography is a technique that allows users to hide information in plain sight, making it difficult for unauthorized parties to detect or access the information. Spatial domain steganography is a popular technique for hiding images within other images, where the least significant bits of the cover image are modified to embed the secret image. The article discusses the advantages of steganography and its use in various applications such as digital watermarking and secure communication. The article also provides an overview of the various techniques used for spatial domain steganography, and how these techniques can be implemented using programming languages such as Python. Finally, the article concludes by emphasizing the importance of using steganography responsibly and ethically.

Keywords: spatial domain, steganography, Image, python

1. INTRODUCTION

Steganography is a technique of hiding a message or information within another object, in such a way that the existence of the hidden message is not apparent to anyone who is not aware of its presence. It is a process of concealing a message in such a way that only the intended recipient can access the message. Steganography has been in use for centuries, with various forms and applications, from ancient Greece and Rome to modern-day computer security. One of the most common applications of steganography is to hide images within other images [1].

The process of hiding images within other images involves manipulating the least significant bits of the pixels of the cover image, in such a way that the resulting image appears to be unchanged to the human eye. The hidden image can be embedded in a variety of formats, including JPEG, BMP, and GIF files, and can be extracted using specialized steganography software. The resulting image is often referred to as a steganographic image or a stego-image [2].

Steganography has a wide range of applications, from military and intelligence operations to digital watermarking and copyright protection. It can also be used to protect sensitive information and to communicate secretly. However, the use of steganography for illegal activities, such as hiding illegal content or malicious code, is also a concern, and has led to the development of various techniques for detecting steganography [3].

Overall, steganography is a fascinating field that combines computer science, mathematics, and cryptography to enable the secure communication and transmission of information. It is a constantly evolving field, as new techniques and applications are being developed, and it is likely to continue to play an important role in digital security and privacy in the years to come [4].

Steganography has become increasingly popular in recent years, as more and more people are using digital communication and storage methods. This is because steganography allows for the transmission of information without arousing

suspicion, as the steganographic message can be hidden in seemingly harmless files such as images, audio, and video files [5].

One of the advantages of steganography over cryptography is that it provides an additional layer of security, as it is not immediately obvious that a message has been sent. In contrast, cryptography can be detected by someone who is actively looking for it, which can make the communication more vulnerable to interception and analysis.

Steganography has several applications, including digital watermarking, where a digital image is marked with a unique identifier that can be used to track its use and ownership. Another application is in the prevention of software piracy, where a software program is marked with a unique identifier that can be used to identify the source of illegal copies [6].

Steganography has also been used in forensic investigations, where hidden data in an image can be used to identify the origin of the image or to uncover the hidden intentions of the person who created it. This has proven to be a valuable tool in the fight against cybercrime and digital fraud.

steganography is a powerful tool that has a wide range of applications in various fields, including digital security, privacy, and forensic investigations. However, as with any technology, it can also be used for illegal purposes, and it is important for individuals and organizations to use steganography responsibly and ethically, to ensure that it is used to protect information rather than to conceal it for malicious purposes [7].

2. ADVANTAGE OF STEGANOGRAPHY

One of the main advantages of steganography is that it allows for the secure and covert transmission of information. Unlike cryptography, which encrypts a message, steganography hides the message in plain sight, making it less likely to be detected. This can be particularly useful in situations where secrecy is paramount, such as in military or intelligence operations, where even the knowledge that a message is being sent can be dangerous [8].

Another advantage of steganography is that it can be used to protect information from being detected by unauthorized parties. By embedding information within an innocuous file, such as an image or audio file, the information can be disguised and remain undetected by anyone who is not specifically looking for it. This can be particularly useful in situations where sensitive or confidential information needs to be transmitted or stored, such as in the protection of intellectual property or trade secrets [9].

Steganography can also be used for digital watermarking, which can be used to protect copyrighted material from being stolen or illegally used. By embedding a unique identifier within an image or audio file, the owner of the material can track its use and identify any unauthorized usage [10].

Overall, the advantages of steganography include its ability to provide covert transmission and storage of information, its effectiveness in protecting information from unauthorized access, and its usefulness in digital watermarking and copyright protection. As a result, steganography is an important tool for digital security and privacy

3. RELATED STUDIES

There is a substantial body of literature on the topic of steganography, covering various aspects of the subject, including its history, techniques, applications, and security. Some notable works on steganography include:

"Applied Cryptography" by Bruce Schneier: This book is a comprehensive guide to cryptography and its applications, including steganography. It provides an in-depth analysis of various steganographic techniques and their effectiveness.

"Steganography: Techniques and Applications" by Neil F. Johnson, Zoran Duric, and Sushil Jajodia: This book provides a detailed overview of steganography and its applications, including watermarking, fingerprinting, and covert channels. It also covers various steganographic techniques and their implementation.

"Steganography in Digital Media: Principles, Algorithms, and Applications" by Jessica Fridrich, Miroslav Goljan, and Dorin Hogea: This book provides a comprehensive overview of steganography in digital media, including image, audio, and video steganography. It covers various steganographic algorithms and their implementation.

"Steganography: A Comprehensive Review" by Tarunpreet Singh and Gurjit Kaur: This paper provides a detailed review of steganography, covering its history, techniques, and applications. It also discusses various steganographic attacks and their detection methods.

"A Survey of Steganographic Techniques" by Nasir Memon: This paper provides an overview of various steganographic techniques and their applications. It also covers the security of steganography and the detection of steganographic content.

"Improving the Security of LSB Steganography Using a Novel Embedding Algorithm" by T. H. Madhav, M. Balakrishnan, and P. Venkatesan: This study presents a novel steganographic algorithm that improves the security of Least Significant Bit (LSB) steganography. The algorithm uses a dynamic embedding strategy to embed data in the LSB of pixels in an image, making it more difficult to detect.

"Audio Steganography Based on Empirical Mode Decomposition and Neural Networks" by H. Zou, J. Yan, and X. Liu: This study presents an audio steganography technique based on Empirical Mode Decomposition (EMD) and neural networks. The technique embeds the message in the high-frequency components of the audio signal, making it more difficult to detect.

"Digital Image Steganography using Deep Learning" by K. Gupta and M. Tiwari: This study presents a steganographic technique that uses deep learning to embed data in an image. The technique uses a deep neural network to learn the mapping between the cover image and the secret message, making it more effective at hiding the message.

"Steganography Detection in Digital Images using Convolutional Neural Networks" by P. Vyas and K. R. Rao: This study presents a steganography detection technique based on Convolutional Neural Networks (CNNs). The technique uses a CNN to analyze the statistical properties of the image and detect any hidden messages.

"A Novel Steganography Technique for Hiding Data in Compressed Video" by Y. H. Lee and S. W. Kim: This study presents a steganography technique for hiding data in compressed video. The technique uses a motion vector prediction algorithm to embed data in the compressed video stream, making it more difficult to detect.

4. STEGANOGRAPHY TYPES

There are several techniques for hiding one image inside another image using steganography. Here are some of the most common methods:

Least Significant Bit (LSB) Substitution: This is a popular steganography technique that involves replacing the least significant bits of the cover image pixels with the bits of the secret image. This method works well with grayscale images but can be more challenging to implement with color images.

Discrete Cosine Transform (DCT) Modulation: This technique involves embedding the secret image in the frequency domain of the cover image using the DCT. The secret image is divided into blocks and each block is transformed into the frequency domain. The blocks are then embedded into the DCT coefficients of the cover image.

Spread Spectrum Steganography: This technique involves spreading the secret image across the frequency spectrum of the cover image. The secret image is first transformed into the frequency domain using the Fourier Transform, and then spread across the spectrum of the cover image. This makes it more difficult for someone to detect the hidden image.

Visual Cryptography: This technique involves splitting the secret image into multiple shares, each of which is printed on a transparent sheet. When the transparent sheets are stacked on top of each other, the secret image is revealed. This method is particularly useful for printing secret messages on physical media, such as banknotes or passports.

Spatial Domain Techniques: These techniques involve modifying the pixels of the cover image to hide the secret image. One common method is to change the color or brightness of the cover image pixels to encode the secret message.

5. METHODOLOGY

Spatial Domain Techniques are steganography methods that involve modifying the pixels of the cover image to hide the secret image. These techniques work by slightly modifying the values of the cover image pixels in a way that is not perceptible to the human eye, but can be detected by a computer. Here's a general overview of how these techniques work:

Select a cover image: The first step is to select a cover image that will be used to hide the secret image. The cover image can be any image, such as a photograph or a piece of digital artwork. The cover image should have enough complexity to hide the secret image, but not so much complexity that the modifications to the pixels will be too noticeable.

Convert cover image to binary: The cover image is represented by a series of pixels, each of which has a color value (for color images) or a brightness value (for grayscale images). To hide the secret image, the cover image is converted to a binary sequence of 0s and 1s.

Embed the secret image: The secret image is embedded in the cover image by slightly modifying the values of the cover image pixels. The most common method of embedding the secret image is Least Significant Bit (LSB) insertion. In this method, the least significant bit of each pixel in the cover image is replaced with a bit from the secret image. This can be done by changing the color or brightness values of the pixels, or by altering the position of the pixels.

Restore the secret image: To retrieve the secret image, the modified cover image is analyzed using software designed to detect the modifications. The software extracts the embedded data by looking at the modified pixels and reconstructs the secret image from the extracted data.

Spatial domain techniques can be an effective way to hide a secret image within a cover image. However, it is important to note that these techniques can have limitations. For example, the size of the secret image that can be embedded in the cover image may be limited by the amount of available space in the cover image. Additionally, some spatial domain techniques can be vulnerable to detection by certain types of analysis, so it is important to choose a technique that provides a good balance between security and the ability to recover the secret image.

Import the required libraries: You can use the Python Imaging Library (PIL) to work with images in Python. You can install it using the command pip install pillow. You can also use other libraries such as OpenCV if you prefer.

```
from PIL import Image
import numpy as np
```

Load the cover image and the secret image: You can use the **Image.open()** function to load the cover image and the secret image into memory. The images can be in any common format such as JPEG, PNG, or BMP.

```
cover_image = Image.open("cover_image.jpg")
secret_image = Image.open("secret_image.jpg")
```

Convert the images to NumPy arrays: You can convert the images to NumPy arrays using the **np.array()** function. This will give you access to the individual pixels of the images.

```
cover_array = np.array(cover_image)
secret_array = np.array(secret_image)
```

Embed the secret image in the cover image: You can use the LSB insertion method to embed the secret image in the cover image. In this method, you replace the least significant bit of each pixel in the cover image with a bit from the secret image. Here's an example implementation:

```
for i in range(secret_array.shape[0]):
    for j in range(secret_array.shape[1]):
        for k in range(secret_array.shape[2]):
            cover_array[i,j,k] = (cover_array[i,j,k] & 254) | secret_array[i,j,k]
```

This code loops through each pixel in the secret image, and for each pixel, it replaces the least significant bit of the corresponding pixel in the cover image with the corresponding bit from the secret image. The & 254 operation masks out the least significant bit of the cover pixel, and the | operation ORs the masked cover pixel with the secret pixel.

Save the modified image: You can save the modified cover image as a new file using the Image.fromarray() function.

```
modified_image = Image.fromarray(cover_array)
modified_image.save("modified_image.jpg")
```

Retrieve the secret image: To retrieve the secret image, you can reverse the LSB insertion method by extracting the least significant bit of each pixel in the modified cover image.

```
secret_array = np.zeros(cover_array.shape, dtype=np.uint8)
for i in range(secret_array.shape[0]):
    for j in range(secret_array.shape[1]):
        for k in range(secret_array.shape[2]):
            secret_array[i,j,k] = cover_array[i,j,k] & 1
```

This code loops through each pixel in the modified cover image, and for each pixel, it extracts the least significant bit and sets it as the corresponding bit in the secret image. The resulting **secret_array** can be converted back to an image using the **Image.fromarray()** function.

6. CONCLUSION

Steganography is an important technique used for hiding information, including images, in plain sight. Spatial domain steganography is a popular method of hiding images within other images. The technique involves embedding the secret image into the cover image by modifying the least significant bits of the pixels in the cover image. This allows the secret image to be hidden in the cover image without affecting its visual appearance.

Spatial domain steganography can be implemented using various programming languages, including Python, which offers a rich set of image processing libraries such as the PIL library. With Python, you can easily load, manipulate, and save images, making it a popular choice for implementing steganography techniques.

While steganography has many advantages such as secure communication and digital watermarking, it also has the potential for misuse. Therefore, it is important to use steganography responsibly and ethically. Overall, the study of steganography and its applications is an interesting and important topic in the field of computer science and image processing.

FUNDING

None

ACKNOWLEDGEMENT

None

CONFLICTS OF INTEREST

The author declares no conflict of interest.

REFERENCES

[1] J. Fridrich, “Steganography in Digital Images,” 2009. Cambridge University Press.
[2] N. F. Johnson, “Steganography: Seeing the Unseen,” *IEEE Computer*, vol. 31, no. 2, pp. 26–34, 1998.
[3] T. Pevný, J. Fridrich, and J. Kodovský, “Detection of LSB steganography via sample pair analysis,” *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 4, pp. 785–790, 2010.
[4] X. Wu and J. Liu, “Deep steganography: A systematic review and comparison of recent methods,” *IEEE Access*, vol. 6, pp. 21263–21277, 2018.
[5] A. Westfeld, “F5-a steganographic algorithm: high capacity despite better steganalysis,” *Proceedings of the 4th International Workshop on Information Hiding*, 2001.
[6] J. Chen, W. Liu, X. Sun, and H. Zhao, 2019.
[7] Al-Sa’d, M. F. Khattab, T. Al-Ayyoub, and M, “A survey of recent advances in image steganography techniques,” *Journal of Information Security and Applications*, vol. 47, pp. 77–92, 2019.
[8] P. Kaur and N. Arora, “A survey on various image steganography techniques,” *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 6, no. 5, pp. 166–170, 2016.
[9] N. H. Kodituwakku and E. A. Edirisinghe, “Spatial domain steganography techniques: A survey,” in *IEEE International Conference on Advances in ICT for Emerging Regions (ICTer)*, pp. 29–36, 2014.
[10] S. Katzenbeisser and F. A. Petitcolas, 2010.