International Journal of Scientific Research & Engineering Trends
Volume 9, Issue 3, May-Jun-2023, ISSN (Online): 2395-566X

Steganography with Maximum Standard Deviation Embedding Technique

Cyril J
Department Of Computer Science
Birla Institute of Technology and Science, Pilani
Bangalore

Abstract- Security is of paramount importance in this current age, where technology has become an integral part of our lives. Steganography is a less explored topic in the field of cyber security which can be used to transmit data securely. Steganography is a practice of hiding secret information within a cover medium. The medium could be of any format. In ancient Greece messages were hidden within the was coating of tables. Today it is being used in digital communication to hide sensitive information. However, steganography can be used to hide malware of illegal information and as a result it is important for organizations to be aware of potential risks and to take appropriate measures against steganographic attacks. The Maximum Standard Deviation Embedding Algorithm is designed based on the spatial domain of an image for simplicity and makes use of mean and standard deviation of the image pixels to embed data. It is a performance friendly, robust algorithm that hides data efficiently.

Keywords- Steganography, Standard Deviation, Image Processing Data Hiding, Encryption, Pixel Shuffling

I. INTRODUCTION

Steganography is the practice of hiding information inside a cover medium without visibly altering the original medium. The medium in this case could be anything including but not limited to audio files, image files, videos files etc., The term originates from the Greek term Steganos meaning ‘covered’ and Graphein meaning ‘writing’.

Steganography techniques can be broadly classified based on the domains as Spatial Domain Steganography where data is embedded into the pixels of the cover image directly and Frequency Domain Steganography where the cover image is converted from spatial to frequency domain using methods like Fourier Transform or Discrete Cosine Transform and then data is embedded.

Maximum Standard Deviation Embedding Technique comes under Spatial Domain Steganography techniques and is one of its kind where the cover medium is divided into equal non overlapping m x n blocks and the mean and standard deviation of each block is calculated. This provides insight on which block has the most deviation or distortion in an image and the secret data is embedded in those blocks to avoid detection.

1. Steganography Domains:
There are multiple domains based on which steganography algorithms are designed.

- **Spatial Domain:** Spatial domain is the image space that is divided into uniform pixels according to spatial coordinates for the given resolution. The pixels contained in the spatial domain consist of components RGB or RGBA depending on the image type where RGB corresponds to Red, Green, and Blue and RGBA is an extension of RGB with an additional Alpha channel that specified the opacity of the color. Steganography Algorithms based on spatial domain directly modify the values of RGB to embed data. Some techniques that use special domain for Steganography are Least Significant Bit (LSB) technique, pixel value differencing (PVD), and Spread spectrum technique.
- **Frequency Domain:** In frequency domain the image is first converted to its frequency distribution that is each image value at a position P represents the amount that the intensity values in the image I vary over a specific distance related to position F. In frequency domain changes in the image position corresponds to the changes in the spatial domain for the image. Algorithms that embed data using the frequency domain include Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), etc.,
- **Distortion Domain:** Distortion techniques store information by signal distortion and measure deviation from the original cover in the decoding step. This technique is not used widely but some algorithm based on distortion techniques include Use of error

correction codes in steganography, Image blurring with sequential LSB embedding, etc.,

2. Masking and Filtering:
Masking and filtering techniques make use of the image properties to embed data. They are suitable for 24-bit image pixels. It is like placing watermarks in an image but in this case, it is invisible. Here the embedding data can be read by modifying the opacity of the carrier image. This technique is mainly used to add copyright information to images to avoid illegal usage.

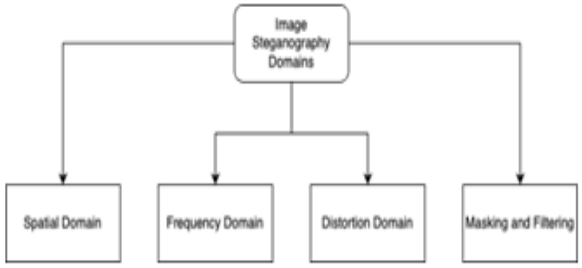

Fig 1. Different domains in image steganography

II. PROPOSED TECHNIQUE

The proposed technique is termed as Maximum Standard Deviation Embedding Technique (MSD). The method involves the usage of mathematical functions to calculate the standard deviation of image blocks and sort the blocks in descending order. In this manner the embedding starts from the block having the highest standard deviation. The algorithm also involved pixel shuffling which adds to the complexity of the algorithm.

Maximum Standard Deviation (MSD) Technique makes use of the spatial domain of images to embed data. This technique embeds data using the Least Significant Bit (LSB) embedding.

1. Algorithm:
Embedding message into the carrier image
The given image is divided into mutually exclusive blocks of order p x q. Each block has a fixed size n.

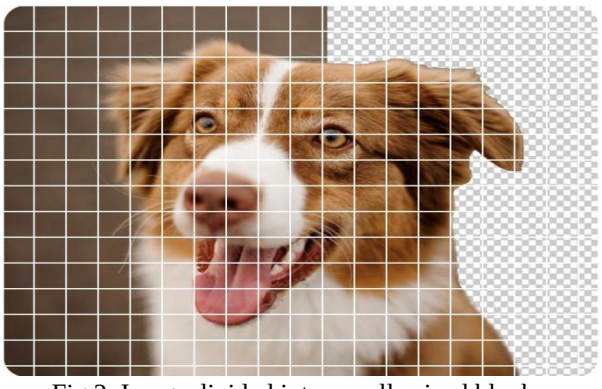

Fig 2. Image divided into equally sized blocks.

The mean and standard deviation of each block is calculated using the following formula:

Mean of Block i:

$$Mean_i = \Sigma\ (pixel_{ii} + pixel_{ij} + \ldots\ldots + pixel_{in}) / n$$

Where n is the number of pixels.

Standard Deviation of Block i:

$$SD_i = \sqrt{\ (\Sigma\ ((pixel_{ii} - mean_i)^2 + (pixel_{ij} - mean_i)^2 + \ldots (pixel_{in} - mean_i)^2) / (n - 1)}$$

Where n is the number of pixels and meani is the mean calculated for each block.

The resulting standard deviation is pushed into a map with the block index ij as the kay and the standard deviation as the value.

The map is then sorted in descending order of the standard deviation.

```
{
  "11": 233.4,
  "24": 212.67,
  .....
  .....
  .....
  "88": 99.1
}
```

Fig 3. Block Standard Deviation Map.

With the help of array rotation perform a 90º clockwise rotation of the image pixels. This can be achieved by performing a transpose of the given image block and then performing a reverse on each row.

The input message that is to be hidden is now encrypted using an autogenerated encryption key. This encryption key is embedded in the encrypted text with a unique delimiter before and after the key.

Fig 4. Resultant Image after performing pixel shuffling.

They encrypted message is now embedded in the image pixels starting from the pixel having the highest standard

deviation by flipping the LSB of the RGB components of the pixel.

The pixels shuffled using array rotation is revered and the steganographic image is returned that looks identical to the original image.

2. Extracting the hidden message from the carrier:

The steganographic image is divided into equal blocks of order p x q where the size of each block must remain the same as the size used during embedding (n).

Standard deviation of each block is calculated, and a map is generated, sorted in descending order of the standard deviation.

A 90º clockwise rotation of the image pixels is performed to achieve the shuffled image.

Now the image blocks are traversed in the order of the standard deviation map.

The LSB of the RGB values are all extracted which returns the encrypted text that was hidden in the carrier image.

The encryption key is extracted from the text with the help of the unique delimiters.

This key is then used to decrypt the encrypted text.

3. Shuffling algorithm:

The pixel shuffling algorithm used performs a 90º clockwise rotation of all the pixels in the block. This is done using the following technique:

The selected block is evaluation to get the height and width of the block. Each pixel in the block is indexed in a 2-D array A.

$$\begin{bmatrix} \mathbf{pixel_{11}} & \mathbf{pixel_{12}} \\ \mathbf{pixel_{21}} & \mathbf{pixel_{22}} \end{bmatrix}$$

Now perform a transpose of the array to attain A^T

$$\begin{bmatrix} \mathbf{pixel_{11}} & \mathbf{pixel_{21}} \\ \mathbf{pixel_{12}} & \mathbf{pixel_{22}} \end{bmatrix}$$

Now each row is reversed to get the resultant array Reversed A^T.

$$\begin{bmatrix} \mathbf{pixel_{21}} & \mathbf{pixel_{11}} \\ \mathbf{pixel_{22}} & \mathbf{pixel_{12}} \end{bmatrix}$$

The resultant array is a 90º clockwise rotation of the array A. Areverse shuffling can be done by following the steps in reverse.

Reverse each row of the array Reversed A^T to get A^T.

$$\begin{bmatrix} \mathbf{pixel_{11}} & \mathbf{pixel_{21}} \\ \mathbf{pixel_{12}} & \mathbf{pixel_{22}} \end{bmatrix}$$

Perform a transpose on the A^T to get the original array$(A^T)^T$ = A

$$\begin{bmatrix} \mathbf{pixel_{11}} & \mathbf{pixel_{12}} \\ \mathbf{pixel_{21}} & \mathbf{pixel_{22}} \end{bmatrix}$$

4. Embedding Algorithm (LSB):

The input text is initially converted into binary form. For example, the string ‘hello’ is converted to:

Table 1. Binary equivalent of string 'hello'

h	e	l	l	o
01101000	01100101	01101100	01101100	01101111

The binary equivalent of ‘h’ requires at least 3 pixels in an RGB image and 2 pixels in an RGBA image.

The selected pixel is evaluation and the binary values of RGB are extracted.

E.g., The RGB values of the following pixel is

Fig 5. Selected Pixel.

Table 2. RGB values of the selected pixel

R	G	B
8	6	141
00001000	00000110	10001101

The data bits are now embedded into the RGB components of the pixel

E.g., Embedding the bits 011 into the pixel

The first 3 bits of the letter h (01101000) is picked and embedded in the LSB of the RGB values.

Embedding bit 0
R (00001000) is converted to 00001000 LSB (0) = 0 hence it is retained

Embedding but 1
G (00000110) is converted to 00000111 LSB (0) ≠ 1 the LSB is flipped.

Embedding but 1
B (10001101) is converted to 10001101 LSB (1) = 1 hence it is retained

The pixel value is converted to RGB (8, 7, 141)

Fig 6. Resultant Pixel.

III. STEGANOGRAPHY STANDARD

Loading Standards:
Loading capacity depends on various factors like the number of pixels in the image, type of the image (RGB or RGBA). To generalize the storage capacity of a carrier image using the MSD technique is given by:

Number of Characters = (C x P) / 8

Where C is the number of channels 3 for RGB and 4 for RGBA
P is the number of pixels in the image.

2. Security:
The algorithm returns an image identical to the original image thereby impossible to detect by a human eye. It does provide considerable immunity against steganalysis techniques. Even if the image has been classified as a stegged image it will be impossible to extract the hidden data with 3 layers of security.

3. Cost Effectiveness:
All the libraries used to build this tool are open source and does not require any paid resources. Thus, the algorithm is very cost effective.

4. Quality:
The quality of the data stored is preserved when they are extracted. The embedding algorithm ensures that the image also does not have any visible impact thereby maintaining the quality of the image as well as the hidden data.

5. Indistinct:
The cover image and the steganographic image created by the MSD technique are not distinguishable by human eye thereby making the image indistinct and perfect.

IV. TOOL SPECIFICATION AND INSTRUCTIONS

The Maximum Standard Deviation Embedding Tool is build using python. In order to run the tool it is necessary to have the latest version of python installed in your system. The tool mainly consists of the following components.

- Secure fernet key generator
- Text Encryption Decryption Module
- Pixel shuffle module
- MSD module.

The user interface mainly contains two tabs Encode and Decode.

1. Encode:
The encode tab is used to encrypt the text and embed the text into the selected image.

- **Select Image:** Used to select the carrier image.
- **Encryption Key:** The Fernet Key module generates a secure key automatically to encrypt the input text.
- **Text Area:** Contains the text that needs to be embedded in the image.

Fig 7. MSD tool encoding tab.

2. Decode:
The decode tab is used to extract the hidden text from the image and decrypt the extracted text using the encryption key.

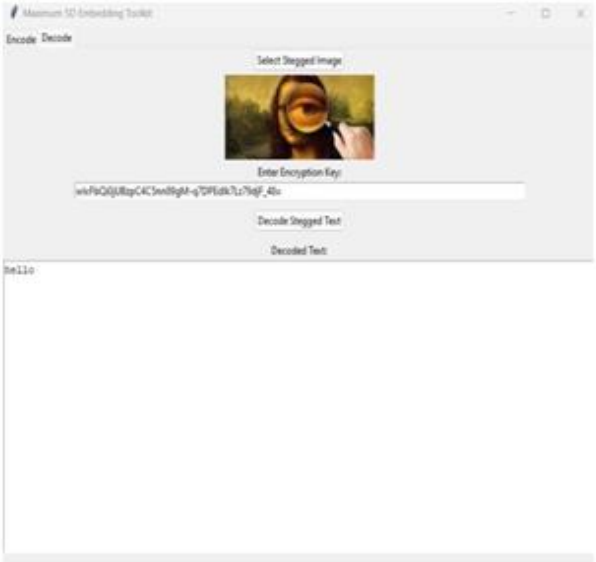

Fig 8. MSD tool decoding tab.

- **Select Stegged Image:** As the name suggests this button is used to select the stegged image.

- **Encryption Key:** This key is automatically extracted from the picture and is populated.
- **Decoded Text Area:** The decoded text is then decrypted using the encryption key and is displayed in this area.

V. CONCLUSION

The Maximum Standard Deviation (MSD) Embedding Toolkit is a comprehensive solution for effective steganography and lossless image manipulation. It incorporates well-established embedding techniques, such as Least Significant Bit (LSB) embedding in the spatial domain, to ensure optimal performance. Additionally, the toolkit includes encryption capabilities, enhancing data security and making the embedded text virtually untraceable.

While steganography techniques have legitimate applications, it's crucial to acknowledge that they can also be misused for malicious purposes. Embedding malwares or other harmful content in files using steganography techniques poses a potential risk, enabling bad actors to carry out malicious activities on the unsuspecting user's device. It is important to promote ethical usage of such tools and ensure that they are employed for lawful and responsible purposes, while also maintaining awareness of the potential risks associated with the misuse of steganography techniques.

REFERENCES

[1] Fateh, M., Rezvani, M., & Irani, Y. (2021, February 5). A New Method of Coding for Steganography Based on LSB Matching Revisited. Security and Communication Networks, 2021, 1–15. https://doi.org/10.1155/2021/6610678
[2] Abdulfetah, A., Sun, X., & Yang, H. (2010, September 15). Robust Adaptive Video Watermarking Scheme using Visual models in DWT domain. Information Technology Journal, 9(7), 1409–1414. https://doi.org/10.3923/itj.20 10.14 09.1414
[3] HIDING TEXT DATA IN AUDIO SIGNALS USING 3BIT LSB TECHNIQUE AND DCT APPROACH. (2020). International Journal of Engineering Sciences & Research Technology, 9(8), 87–92. https://doi.org/10.29121/ijesrt.v9.i8.2020.11
[4] R. Wang, "Discrete-time Fourier transform," Introd. To Orthogonal Transform. no. 1, pp. 146–219, 2013.
[5] Sencar, H. T. (2006, October 1). Performance study of common image steganography and steganalysis techniques. Journal of Electronic Imaging, 15(4), 041104. https://doi.org/10.1117/1.2400672
[6] Omoomi, M., Samavi, S., & Dumitrescu, S. (2010, April 13). An efficient high payload ±1 data embedding scheme. Multimedia Tools and Applications, 54(2), 201–218. https://doi. org/1 0.1007/s11042-010-0517-z
[7] Albahar, M. A., Olawumi, O., Haataja, K., & Toivanen, P. (2018). Novel Hybrid Encryption Algorithm Based on Aes, RSA, and Twofish for Bluetooth Encryption. Journal of Information Security, 09(02), 168–176. https://doi.or g/10.4 236/jis.2018.92012
[8] X. Zhang and D. Zhang, "Research on Encryption Algorithm Based on Python," 2017 International Conference on Computer Technology, Electronics and Communication (ICCTEC), Dalian, China, 2017, pp. 586-588, doi: 10.1109/ICCTEC.2017.00132.
[9] Reversible steganography in image : an overview and review. (2016). International Journal of Latest Trends in Engineering and Technology, 8(1).
[10] StegExpose – Steganalysis Tool For Detecting Steganography In Images – Darknet – Hacking Tools, Hacker News & Cyber Security. (2014, September 15). Darknet – Hacking Tools, Hacker News & Cyber Security. https://www.darknet.org .uk/2014/09/stegexpose- steganalysis-tool-detecting-steganography-images/
[11] The Dark Side of Steganography. (2015, August 13). IEEE Spectrum. https://spectrum.ieee.org/the-dark-side-of-steganography
[12] Research Shows Image-Based Threat on the Rise. (2007, October 18). Dark Reading.
[13] https://www.darkreading.com/risk/research-shows-image-based-threat-on-the-rise