International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 12, Number 12 (2019), pp. 2317-2320

Destroying Embedded Malicious code in Image (DEMI): A Proposed Method

Meshaiel M. Alsheail[1] and Dina M. Ibrahim[1,2]

[1]Information Technology Dept., College of Computer, Qassim University, Buraidah, KSA.
[2]Computers and Control Engineering Dept., Faculty of Engineering, Tanta University, Egypt.
(ORCID: 0000-0002-2991-7299*[1]*, 0000-0002-7775-0577*[2]*)

Abstract

Image plays a significant role in the modern communication information exchange which enabled cybercriminals to use images to play a big part in spreading malicious code hidden in an image. In this paper, we propose a new approach to destroy the hidden code embedded in the image without affecting the content of it, we called our proposed method Destroying Embedded Malicious code in Image (DEMI). In our proposed method, we assumed that the attacker uses the Least-Significant Bit (LSB) technique to embed the code data in the cover image. The proposed algorithm employs the Sample Pair Analysis technique (SPA) to estimate the number of flipped LSBs in the image then flip the first bit (least) which changes the embedded code. Our experimental results demonstrate the effectiveness of the proposed technique and its ability to remove the embedded malicious code while preserving image contents.

Keywords: Steganography, LSB, Steganalysis, SPA.

I. INTRODUCTION

Users prefer to use images to share and view information through the Internet. Cyber criminals take advantages of the trust user thought in images. However, the image is innocent until the user double click on it which activate the executable malicious code that hidden on it. Then, the image will no longer innocent as before. Hiding messages into a medium so there is no sign of the existence of the hidden data is an old way of communication known as steganography, using a digital image as a cover to hide data "Stego message" in a way that is nondeductible easily is popular [1]. Steganography techniques help cybercriminals to embed malicious or harmful code into images that look normal or innocent [2].

This paper is structured as follows; Section II presents a brief background on image steganography. In Section III, the proposed DEMI method is illustrated and discussed. The experiment analysis and results are shown in Section IV. Finally, we draw the conclusion in Section V.

II. BACKGROUND

Image steganography known for hidden messages "data" into an image with taking into account that the change in the appearance of the image is not easily observed with the naked eye. This technique is a popular technique that can spread secret messages into images quickly in the Internet [3].

One of the popular methods of image steganography is using the least-significant bit (LSB). LSB is a simple approach of steganography by using least significant bits of an image that used as a cover to embed the bits of the message (the secret data) into it [4]. Embedded into the last bit changes the image in a very simple way that cannot be detected or even observed with user bare eye. In a gray image uses bits of each pixel in the image. However, this method uses 3 pixels of a 24-bit color image a bit of each of the red, green and blue color [5]. LSB done by replacing the last bit in each pixel of the original image into the bit of the hidden message. This replacement does not significantly affect image quality. It is an advantage that a person cannot notice any change in the picture. LSB considered one of the most famous way of image steganography because of it is ease and effectiveness [5]. Also, LSB has an advantage in steganographic image quality [6]. Because of the simplicity of LSB steganography, plenty of popular steganographic tools built on it such as S-Tools V. 4.00 [7].

Since the steganography used to hide data in digital media, the steganalysis do the opposite by use it to extract hidden information. Steganalysis is the art of extracting the secret message by determine the length of the hidden information or the modification ratio. However, at this moment there are different steganography techniques encourage the existence of different quantitative steganalysis [8]. One of the quantitative steganalysis methods that corresponding to LSB steganography method is sample pair analysis (SPA). Authors in [9] introduce a new simple and fast approach (SPA) sample pair analysis to detecting steganography in least significant bit (LSB) in digital media like images by estimating with high accuracy the embedded message length in the least significant bits of signal samples. SPA is a statistical approach used with LSB embedding operations with high sensitivity that measuring sample pairs [9]. Although the LSB is popular, this technique is less reliable to have complete recovery for the hidden data when Stego images exposed to image manipulation. Thus, any modification to the data in the Stego image will destroy the embedded data [10]. Thus, in this paper we proposed our approach of detecting and destroy embedded code in images without affecting the content of the image.

III. DESTROYING EMBEDDED MALICIOUS CODE IN IMAGE (DEMI): A PROPOSED METHOD

We have proposed an approach, which we called DEMI, to remove the risk of open image injected by executable malicious code that will harm your computer or used to steal personal

information. Figure 1 shows the diagram for the proposed method.

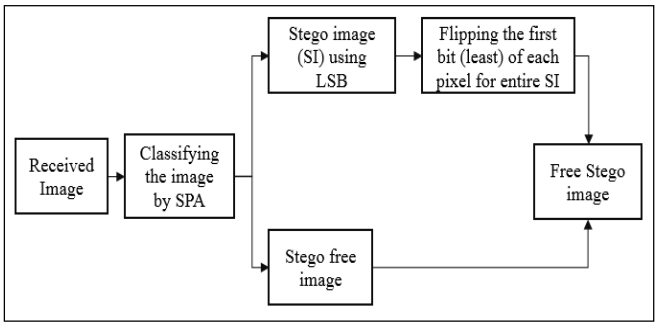

Fig. 1. DEMI proposed method diagram

In our experiment, assuming that the attacker uses LSB technique to embed the harmful code in the cover image before send it to the victim as an innocent image. When the user receives an image, at first, we assume it is an injected image SI (Stego-image) that means the image used as a cover image to hide malicious executable code. The attacker waits for the receiver to click on the image to look at it. Directly when the victim clicks on the image it opens an instant download page hiding behind the image and immediately starts downloading the malicious file. To avoid such a problem Destroying Embedded Malicious code in Image (DEMI) algorithm will work in two phases: first, analysis the image to check if there are any embedded codes. Second change the LSB to destroy any hidden contents.

III.I First Phase (Analysing the Image)

In the first phase, we assumed that the used image is a Stego-image (SI) which is the known name for images after steganography. We analyze SI by applying the Sample Pair Analysis technique (SPA) to estimate the number of flipped LSBs in the image. As done in research [9], in their study showed that the decision threshold can be estimated as 0.023. If the threshold greater than or equal to 0.023 that means the image contain a hidden information (hidden data) classify as Stego image. If the result less than 0.023, the image is Stego free [9]. After classifying the image into Stego or not by analyzing it using SPA, if the image considers as a Stego image move to the second phase.

III.II Second Phase (Destroying the Embedded Code)

In the second phase, we perform XOR operations on SI by flipping the first bit (least) of each pixel and that can be applied to the entire SI. This process will result in a new image and all hidden data will be destroyed [10].

Furthermore, we maintain a fair quality with less color degradation of the images. We can summarize our proposed method, Destroying Embedded Malicious code in Image, (DEMI) Method into three steps:

- Step 1: We assume that the attacker has been used LSB technique to embed the spam data in the cover image. Hence, we utilize sample pair analysis technique (SPA) to estimate the number of flipped LSBs in the SI, it refers to the average error and could be estimated as 0.023 according to [9].
- Step 2: Since we discovered that the SI has been detected to contain hidden data, we scramble and flip the first bit (least) of each pixel for entire SI.
- Step 3: return Stego free image while preserving the contents of the image.

Figure 2 illustrates the Pseudocode of the steps of our proposed DEMI method.

Pseudocode of (DEMI) Proposed Method

- Input: Image (SI)as Stego Image where we assuming it has hidden data
- Output: Stego free image SF,

1. Let SF ← SI
   if SPA (SI) > = Decision threshold t then
   go to Step 2
   else No Stego detected, go to Step 3
2. **for** each pixel u in SI rows do,
   for each pixel v in SI columns do,
   SF (u, v) ← SI (u, v) XOR 1
3. return SF

Fig. 2. Pseudocode of the proposed DEMI method

Figure 3 demonstrates that it is very difficult to human eye to notice or recognize any changes in the appearance of the image. The figure presents three appearance of the same image: original colored image, Stego colored image, and destroyed colored image before and after applying DEMI method.

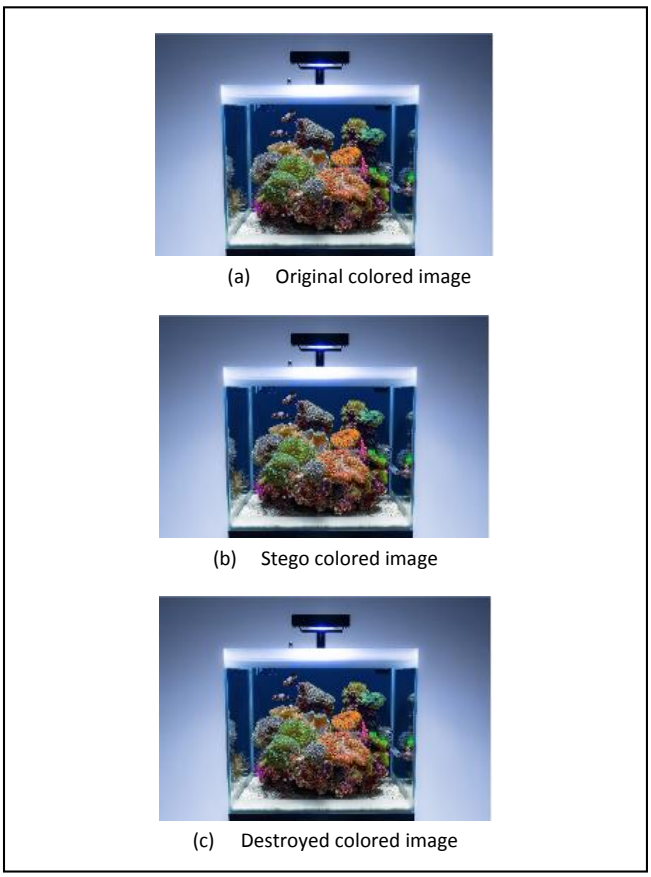

Fig. 3. Before and after applying DEMI proposed method to color

The flowchart of the proposed Destroying Embedded Malicious code in Image (DEMI) Method is presented in Fig. 4

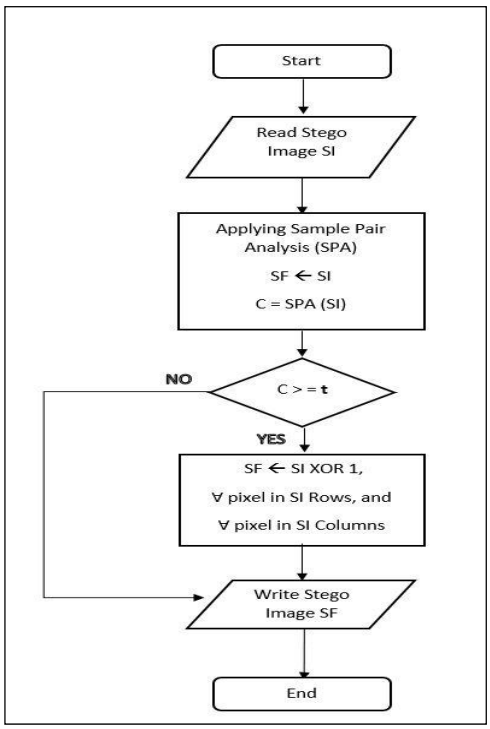

Fig. 4. Flowchart for the proposed DEMI method

Authors in [13] proposed a product that provides the same feature but with some differences. The notifications in this product are sent to the user through email and they display the baby stream using the web browser while in this paper, sending notifications and displaying the baby stream through the Android Application.

IV. EXPERIMENT ANALYSIS AND RESULTS

More than 20 grayscale and color Stego images with embedded contents were used in this experiment to proof the effectiveness of our proposed DEMI method. Figure 5(a) represents an example of an original grayscale image before embed the harmful content on it.

Shape and appearance of image remains the same after harmful content has been injected in the image (Stego image) as per shown in Fig. 5(b), and Fig. 5(c) represents the image after destroying and treating it and remove the hidden data to be Stego free image. The three images look the same for human eye. However, Fig. 6 represent the different in bit after performing XOR operations to flip the first bit (least) of each pixel for entire image.

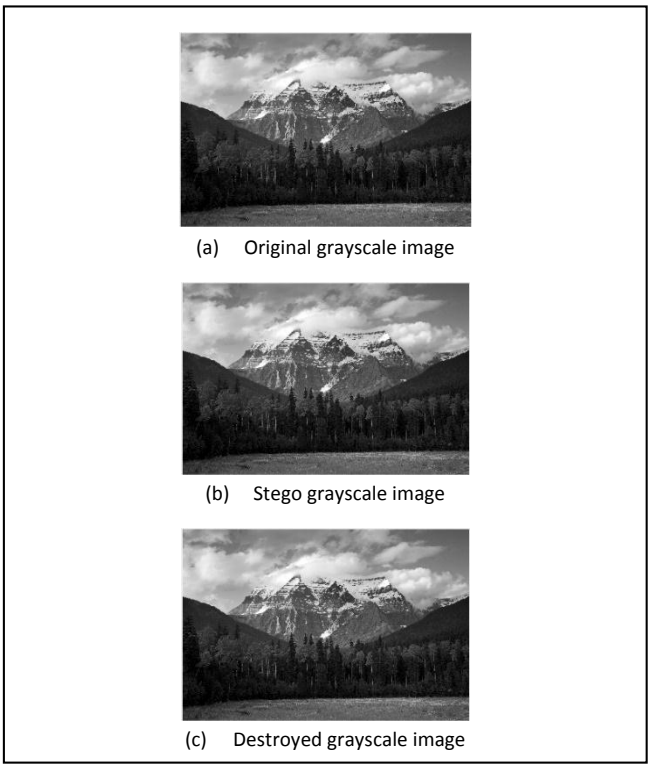

Fig. 5. Before and after applying DEMI proposed method to grayscale image

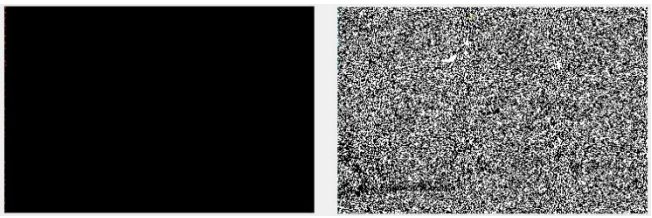

Fig. 6. The different in bit after performing XOR operations

V. CONCLUSION

Since steganography is used to hide information into digital media, the steganalysis do the opposite by use it to extract hidden information. For this purpose, we proposed a method for Destroying Embedded Malicious code in Image, as we called DEMI Method. In DEMI method, we assumed that attackers used LSB technique on the Stego images. We tested the method on colored and grayscale images. Our experimental results demonstrate the efficiency of the proposed technique and its ability to remove the embedded malicious code while preserving image contents.

REFERENCES

[1] Sharifzadeh M., Aloraini M., and Schonfeld D., "Adaptive Batch Size Image Merging Steganography and Quantized Gaussian Image Steganography," in IEEE Transactions on Information Forensics and Security, vol. 15, pp. 867-879, 2020. doi: 10.1109/TIFS.2019.2929441

[2] Darbani A., AlyanNezhadi M. M., and Forghani M., "A New Steganography Method for Embedding Message in JPEG Images," 2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI), Tehran, Iran, 2019, pp. 617-621. doi: 10.1109/KBEI.2019.8735054

[3] Kanojia P. and Choudhary V., "LSB Based Image Steganography with The Aid of Secret Key and Enhance its Capacity via Reducing Bit String Length," 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2019, pp. 257-262. doi: 10.1109/ICECA.2019.8821917

[4] Chitradevi B., Thinaharan N., and Vasanthi M., Data Hiding Using Least Significant Bit Steganography in Digital Images. 2017. doi: http://doi.org/10.5281/zenodo.262996

[5] Li B., He J., Huang J., and Shi Y. Q., ‘‘A survey on image steganography and steganalysis,’’ International Journal of Information Hiding Multimedia Signal Process., vol. 2, no. 2, pp. 142–172, 2011.

[6] Kaur P., Singh H., Gupta A., and Girdhar A., "An improved steganographic approach to diminish data modification for enhancing image quality," 2014 International Conference on Medical Imaging, m-Health and Emerging Communication Systems (MedCom), Greater Noida, 2014, pp. 329-333. doi: 10.1109/MedCom.2014.7006027.

[7] Malekmohamadi H. and Ghaemmaghami S., "Steganalysis of LSB based image steganography using spatial and frequency domain features," 2009 IEEE International Conference on Multimedia and Expo, New York, 2009, pp. 1744-1747. doi: 10.1109/ICME.2009.5202858

[8] Joseph P. and Vishnukumar S., "A study on steganographic techniques," 2015 Global Conference on Communication Technologies (GCCT), Thuckalay, 2015, pp. 206-210. doi: 10.1109/GCCT.2015.7342653

[9] Dumitrescu S., Wu X., and Wang Z., “Detection of lsb steganography via sample pair analysis,” IEEE Transactions on Signal Processing, vol. 51, no. 7, pp. 1995–2007, July 2003.

[10] Simalaotao P., “The prototype development of traceability system of LSB steganography in image files with Base64 and MD5 encoding,” Journal of Thai Interdisciplinary Research, vol. 14, no. 1, pp. 29-34, 2019.