Vol 6 • Number 2 April - Sep 2015 pp. 253-257 **Impact Factor: 2.5** **Available at** www.csjournals.com

RAKS. M. PK 1.0, an Efficient Methodology to Determine The Steganography Signature Of Steganography Tools

Er. Rakesh Kumar[1], Pratik Kumar[2]
[1]Computer Science Researcher, Department of Computer Science & Engineering, Ranchi University, India
[2]B. Tech. Final Year, Department of Computer Science & Engineering, CIT, Ranchi University, India
rsharma.ranchi2009@gmail.com,, pratikkumar2808@gmail.com

Abstract: As we all know, Steganography is being used in terrorist activities these days. Terrorist used to hide information behind carrier image in form of text or image which may be transferred sent or received through Social Networking Sites such as Facebook, Tweeter, Linkedin, etc. but unfortunately, no social networking sites have control over such Stego Image transfer. In this paper, we have discussed why RAKS. M. PK 1.0 is better and efficient over other techniques of to detect whether an image is stego or normal by explaining the working mechanism of RAKS. M. PK 1.0. With the help of this paper we put your effort to focus on those issues that make it difficult to find the Steganographic Signature of the different Steganography tools. Also, we have tried to find out the Steganographic Signature of "Securengine Pro" by the analysis of different graphs and patterns.

Keywords: Steganography, RAKS. M. PK 1.0, Steganography Signature, Known Cover Attack, Security Threats.

1. **Introduction:** Nowadays, a number of Steganography tools are available and use of Steganography is increasing day by day where Cryptography is not so very efficient. People tries to communicate with one another secretly. Law enforcement authorities have concerns in the trafficking of illicit materials through web page images, audio, and other files. It is very important to have the knowledge of methods of detecting hidden information and understanding the overall structure of this technology as it is crucial in uncovering these activities. People are using Steganography to avoid these policies and to send the message secretly.

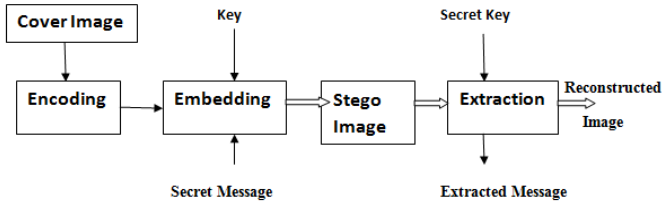


Fig. 1.1

In this paper, we have given all give a brief definition of Steganography and Steganalysis in general to provide a good understanding of these two terms but more importantly, we will talk about how to detect the existence of hidden information in case of Known Cover Attack and try to find out the Steganography Signature of "Securengine Pro" so that it will lead to uncover the Steganography Signature of various Steganography tools.

2. Terminologies Used in this original Research Paper:

Steganography[1]:the word Steganography comes from the Greek name "steganos" (hidden or secret) and "graphy" (writing or drawing) and literally means hidden writing. Steganography uses techniques to communicate information in a way that is hidden. Steganography hides the existence of a message by transmitting information through various carriers. Its goal is to prevent the detection of a secret message. In other words, Steganography is the art and science of writing hidden message in such a way that no one, apart from the sender and the intended recipient, suspects the existence of the message, a form of security through obscurity. The advantage of Steganography, over the Cryptography alone, is that messages do not attract attention to themselves, as they look like any normal bitmap or jpeg image.

Steganalysis[1]: Steganalysis is the discovery of the existence of hidden message; therefore, like Cryptography and Cryptanalysis, the goal of Steganalysis is to discover hidden information and to break the security of its carriers.

Steganography Signature[1]: It is nothing but the repetitive pattern which is generally found in Stego images created with the help of different Steganography tools available in the present days. The repetitive pattern reveals the identification or signature of a tool can reveal the existence of hidden information.

We can analyze these patterns by comparing the original cover images with the Stego images and try to see the differences. This is called a Known Cover Attack. Finally after finding the Steganography signature of a particular Steganography tool, we will arrive to a conclusion that whether an image is a stego-image or not. If the pattern found in the graph of an image is matched with the Steganography Signature then we can say it's a stego image.

Zero matrix[2]: It is the special type of matrix in which all the elements are zero and is denoted by '0' e.g:

$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$ is a null matrix.

Row matrix[2]: A special type of matrix which has a single row is called a row matrix e.g:

$$[1\ 2\ 3\ 4\ 5]$$

Column matrix[2]: A special type of matrix which has a single column e.g.

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

3. Techniques that are the part of Steganography[1]:

Watermarking:

a. Protects copyright owners of digital documents by hiding a signature in the information in a way that even a modified part of the document conserves the signature.

b. Prevents discovery by marking in a hidden and unique way every copy of a confidential document.

Cover Channel:

a. Allows people to communicate secretly by establishing a secret communication protocol.

b. Allows non-authorized communication through authorized communication of a firewall.

4. Types of attacks used by the Steganalyst[1]:

Stego-only Attack: Only the Stego-object is available for analysis. For example, only the Stego-Carrier and the hidden information are available.

Known Cover Attack: The original cover object is compared with the Stego-object and pattern differences are detected. For example, the original image and the image containing the hidden information are available and can be compared.

Known message Attack: A known message attack is the analysis of known patterns that correspond to hidden information which may help against attacks in the future. Even with the message, this may be very difficult and may be considered the same as a stego-only attack.
Chosen Stego Attack: The Steganography tool (algorithm) and Stego-objects are known. For example, the software and the stego-carrier and hidden information are known.

Chosen Message Attack: the Steganalyst generates a Stego-object from some Steganography tool or algorithm of a chosen message. The goal in this attack is to determine the corresponding patterns in the Stego-object that may point to the use of specific Steganography tools or Algorithms.

Known Stego Attack: The Steganography tool (algorithm) is known and both the original and Stego-object are available.

5. Tools that we have used in our Research Work.

a. Turbo C++ IDE
b. Securengine Pro v1.0
c. MATLAB R2014a

Theories used in our research works:

a. Concept of System Time Freezing[3].
b. Concept of Steganography Signature[1].

6. Initial Research Work: We have used bitmap image of size 20x20 in our research. It has been observed that when Steganography is done on a gray scale image, there occurs some changes in its pixel values which leads to small distortions in the image. Using Securengine Pro v1.0, we performed Steganography on a gray scale 20x20 bitmap image, the text file we used contained only letter ‘a’, and the password used was ‘aaaa’.

We have found after multiple research that each time a Steganography is performed on a bitmap image using same text file and the password, even then every time changes occurs in different pixel values without any fixed pattern detected in the change of pixel positions used for hiding the bit values. We arrived at a conclusion that these changes in the pixel positions actually depends on the System time. In our next research we first froze the system time and performed the above research again. Disappointingly we noticed that still the pixel positions used during Steganography are not constant. Satisfied though, we found that the changes occurring now are quite less as compared when the system time was not frozen.

7. Proposed Work:

a. First of all we froze the system time by using the following code[4]:

(Please Refer Text Box 1)

b. We used Securengine Pro v1.0 to make a no. of Stego images. For this purpose we used a text file that contains “a” as text, “aaaa” used as password. AES algorithm is used for encrypting the text to create Stego images using the Securengine Pro v1.0.

c. We followed three methodologies for the analysis to find some common properties. For this purpose we developed three algorithms of our own i.e. RM 1.0, RM 2.0, RMP 3.0

C++ code to freeze the system time

```
#include<dos.h>
#include<stdio.h>
void main()
{
        int i, p_hr, p_min, p_sec,
        p_humd;
        struct time t;
        gettime(&t);
        printf("system time freezed
        ... !!!");
        for(i=0; ; i++)
        settime(&t);
}
```

Text Box 1

Algorithm RMP 1.0:

Step I: Read a Normal image as well as its corresponding Stego image in MATLAB.

Step II: Convert Normal image and its corresponding Stego image both from rgb matrices to gray scale matrix using the MATLAB command "rgb2gray".

Step III: Convert both the gray scale matrices into column matrix.

Step IV: Convert both the vertical matrices into row matrix.

Step V: Lastly plot the graphs.

Note: Before applying this Algorithm we found that even changing the rgb image into gray scale, the hidden text message remains unchanged and it may be retrieved easily by the authorized user.

Flowchart of Algorithm RMP 1.0:

MATLAB Code:

```
a = imread(' D:\sti\N_Image1.bmp '); b = imread(' D:\sti\Stego_img1.bmp');
a1 = rgb2gray(a); b1 = rgb2gray(b);
a2 = a1(:); b2 = b1(:);
a2_ = a2'; b2_ = b2';
plot(a2_, 'color', 'red'); hold on;
plot(b2_, 'color', 'green');
```

Algorithm RMP 2.0:

Step I: Read a Normal image as well as its corresponding Stego images in MATLAB.

Step II: Convert Normal image and its corresponding Stego image both from rgb matrices to gray scale matrix using the MATLAB command "rgb2gray".

Step III: Now find the difference by subtracting gray scale matrix of Stego image from that of normal image and store the result in another matrix of same dimension.

Step IV: If the resultant matrix is a zero matrix (i.e. all the elements of that very matrix is zero) then it will be a normal image otherwise it will be a Stego image.

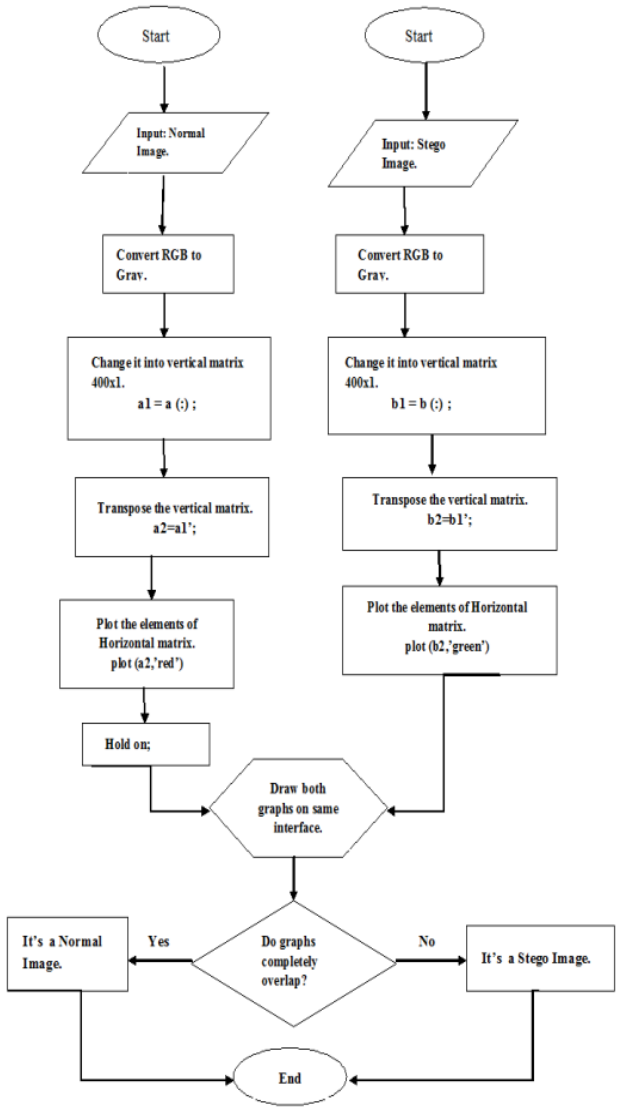


Fig. 7.1

MATLAB Code:

```
a = imread(' D:\sti\N_Image1.bmp '); b = imread(' D:\sti\Stego_img1.bmp');
a1 = rgb2gray(a); b1 = rgb2gray(b);
c = a1-b1;
```

If c is a zero matrix, it's a normal image otherwise it's a Stego image.

Algorithm RMP 3.0:

Step I: Prepare a chart / table in MS Excel sheet with the following columns:

a. BIT NO. IN WHICH CHANGE IS OCCURRED
b. NO. OF BITS CHANGED
c. NO. OF 1'S IN 1ST
d. NO. OF 0'S IN 1ST
e. NO. OF 1'S IN 2ND
f. NO. OF 0'S IN 2ND
g. DIFFERENCE

Where NO. OF 1'S IN 1ST and NO. OF 0'S IN 1ST denotes the no. of 1's and 0's in the binary form of pixel values of normal image and Stego image respectively.

Where NO. OF 1'S IN 2nd and NO. OF 0'S IN 2nd denotes the no. of 1's and 0's in the binary form of pixel values of normal image and Stego image respectively.

DIFFERENCE means the change in the pixel values of the normal image as well as the Stego image.
Step II: Feed the data as of required.
Step III: Analyze the chart / table.

8. Result and Analysis:

By using Algorithm RMP 1.0, we got graphs like this:

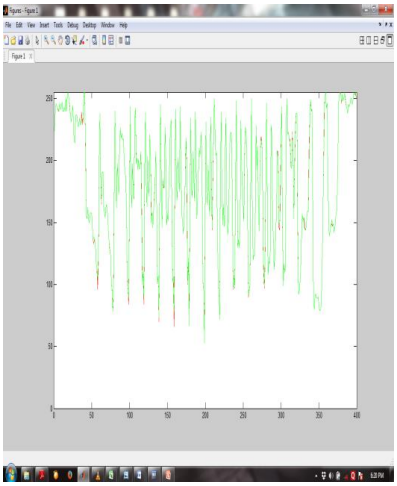


Fig. 8.1.

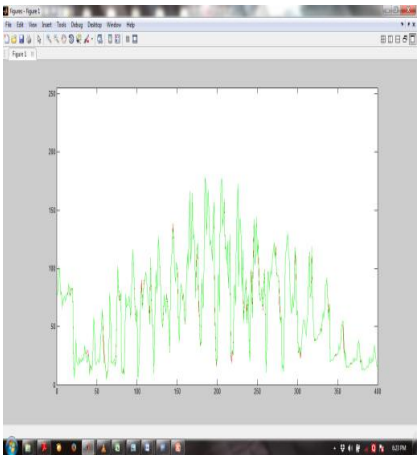


Fig. 8.2

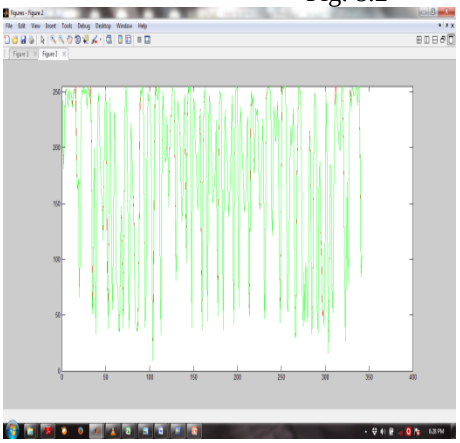


Fig. 8.3

By using Algorithm RMP 2.0, we got matrices like this:

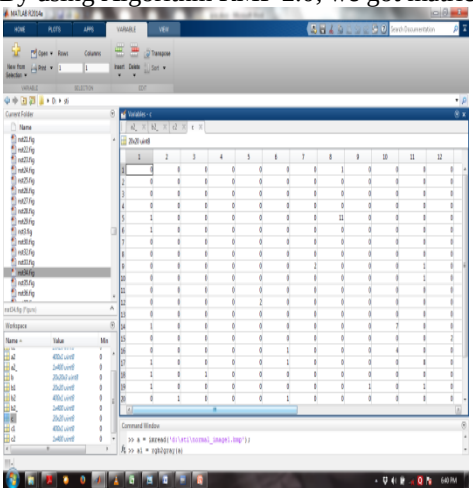


Fig.8.4

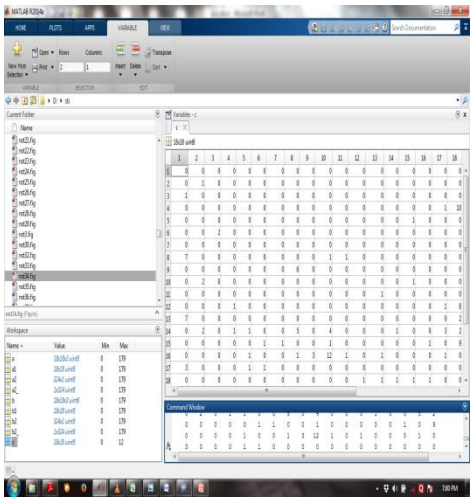
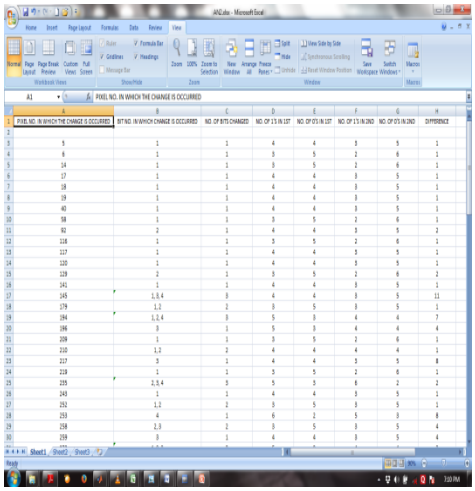


Fig. 8.5

By using Algorithm RMP 3.0, we got chart / table like this:



9. Conclusions:

After the analysis of the above methodologies we came into the following conclusions:

a. Conclusion 1:

Our Algorithm RAKS. M. PK 1.0 is found to be very efficient in case of Known Cover Attack.

b. Conclusion 2:

If $A_1 = \begin{pmatrix} A_{11}, A_{12}, \ldots\ldots, A_{1n}; \\ A_{21}, A_{22}, \ldots\ldots, A_{2n}; \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ A_{m1}, A_{m2}, \ldots\ldots, A_{mn} \end{pmatrix}$

and

$B_1 = \begin{pmatrix} B_{11}, B_{12}, \ldots\ldots\ldots, B_{1n}; \\ B_{21}, B_{22}, \ldots\ldots\ldots, B_{2n}; \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ B_{m1}, B_{m2}, \ldots\ldots, B_{mn} \end{pmatrix}$

Then the elements of 'B' will always be less than or equal to the elements of 'A' whereas 'A' and 'B' are the gray scale matrix of the Normal as well as Stego image.

c. Conclusion 3:

If a1, b1 be the number of ones (1's) and zeros (0's) of normal image and a_2, b_2 are the number of ones (1's) and zeros (0's) of stego image then $a_2 < a_1$ and $b_2 > b_1$ always. [From Algorithm RMP 3.0].

d. Conclusion 4:

From Algorithm RMP 3.0, we conclude that the change occurring in the pixel value is only upto a maximum of four bits (i.e. 1st bit, 2nd bit, 3rd bit and the fourth bit). That means maximum change that may occur in the pixel value is 8+4+2+1 i.e. 15. If the change will occur in between 5th to 8th bit the quality of the image will not be same as of normal image.

References:

[1] SANS Institute InfoSec Reading Room: http://www.sans.org/reading-room/whitepapers/stenganography/steganalysis-detecting-hidden-information-computer-forensic-analysis-1014

[2] B. S. Grewal / Higher Engineering Mathematics / Linear Algebra : Determinants and Matrices / Page 30-31/ Thirtyninenth Edition.

[3] P. Munda, S. Gola, B. Kumari, "Security Threats Related To Stegano Images", IJCSET, ISSN No. 2229-3345, Vol. 3, no. 11, pp. 530-536, November 2012.

[4] FREAK SENSE, "How to freeze computer time using simple computer code": http://freaksense.com/how-to-freeze-computer-time-using-simple-computer-code/