IOSR Journal Of Computer Engineering (IOSRJCE)
ISSN: 2278-0661 Volume 1, Issue 2 (May-June 2012), Pp 34-48
www.iosrjournals.org

Image Steganography and Global Terrorism

Kaustubh Choudhary
Scientist, Defence Research and Development Organisation,
Naval College of Engineering, Indian Naval Ship Shivaji,
Lonavla, Maharashtra, India

Abstract- *This paper informs the reader how an innocent looking digital image hides a deadly terrorist plan. It analyses the strengths of image steganography and the reasons why terrorists are relying on it. It also aims to generate interest among the scientific community about Image steganography so that experts from multiple disciplines may join hands for devising better steganalysis techniques against the global terrorism. In this paper a basic digital image is analyzed and based on this analysis an LSB based steganographic algorithm is designed. Using this algorithm a software is developed and its performance is compared with various other steganographic tools available on the internet. The result of this comparison explains us why steganography is so much preferred by terrorists. Various images and image related tables used in this paper are generated using Matlab Simulation Environment.*

Keywords: *CIA Compliance, Cyber Crime, Image Steganography, Information Security, LSB Insertion, Terrorism.*

I. INTRODUCTION

Information Security rests on the three pillars of Confidentiality, Integrity and Availability of the communication channel. The trio is jointly called CIA compliance of the information network. The confidentiality means concealment of the sensitive information or software from the person who is not supposed to know the same. Integrity means trustworthiness of the data. It can be insured only if the data is obtained from the dependable sources and is frequently updated time to time. Availability means ability to use the information or resource as and when desired by the concerned user. To achieve this, most command and base level military communication use hack proof dedicated Wide Area Networks based on special Mil-Grade software protocol stack and a separate hardware at the physical layer itself. But even terrorist organizations need CIA Compliant communication channel. But most of their agents operate deep inside the civilian population and cannot afford dedicated networks used by their government counterparts. In such a scenario only other communication channel available to them is the World Wide Web. In order to use Internet as a CIA Compliant network, these terror organizations are developing and using special kind of softwares which allows them to clandestinely communicate with full secrecy [1]. Fig.1 explains the general working of one such most frequently used software.

It uses combination of steganography (*Explained in next para*) and cryptography. Both the communicating parties have the same software represented in the Fig.1 by Z. This software Z (mathematically it is crypto-stego function) is fed with the information D (ASCII Characters), a Cover-image C (any image file) and a alpha-numeric secure key K. At first step it encrypts the information D using secure-key K using any particular Encryption algorithm. In next step this encrypted data is hidden in the cover image C. This modified image containing the data D is called as Stego-image and represented in Fig 1 by S. The stego-image is sent to the receiver either through E-mail or by uploading the same in any particular website. The receiver feeds this Stego-image S in to the software Z (possessed by both the communicating parties) along with the secure-key K (known to both the users) and gets the required data D. Ample documentation and research articles are available on the internet regarding various encryption standards like AES, DES, 3DES and IDEA to mention few. So in this paper we are concentrating on the steganographic component of any such software.

Steganography is the art and science of writing hidden messages in such a way that no one, apart from the sender and intended recipient suspects the existence of the message. It is based on invisible communication and this technique strives to hide the very presence of the message itself from the observer. It is two part word of Greek origin- stecho or covered and graph or writing. The earliest known stegenography technique was developed in ancient Greece around 440 B.C. Herodotus's Histories describes the earliest type of stegenography. It states that *"The slave's head was shaved and then a Tattoo was inscribed on the scalp .When the slave's hair had grown back over the hidden tattoo, the slave was sent to the receiver. The recipient once again shaved the slave's head and retrieved the message".*

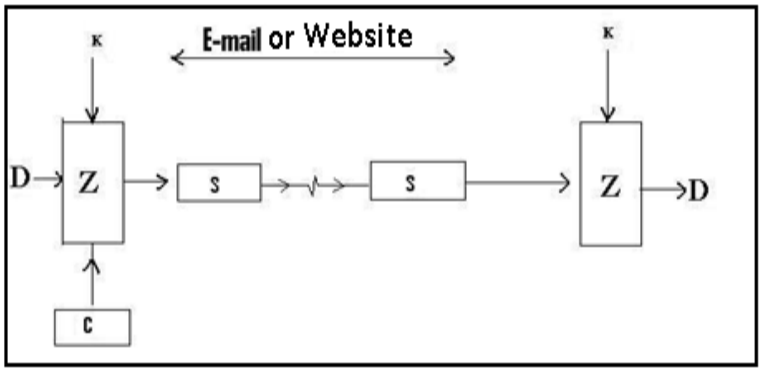

Fig 1 General Working of the Clandestinely Communicating Sofware

All steganographic techniques use Cover-Object and the Stego-Object. Cover-object refers to the object used as the carrier to embed the messages into it. In the above example the slave's head is the cover object. In modern context Images, file systems, audio, and video as well as HTML pages and even email-spams can be used as cover objects. Whereas stego-object is the one which is carrying the hidden message. I.e. in the above example the '*slaves head with fully grown hair and a hidden tattoo*' is acting as a stego-object. In the software mentioned above the Digital Image with hidden data is the stego-object.

Image steganalysis (finding out the information stored in the stego-image) is a challenging task. Analysis of the image requires complex mathematical and statistical operations to be performed on large three dimensional matrices and consumes very high computational power. Also we do not have any foolproof statistical and mathematical function capable of crisply differentiating the stego-image from bulk of innocent images flowing through the web every day. This identification of the steganographic signature in the stego-image is the most difficult part of any steganalysis algorithm. Somehow even if we identify the stego-image still the lack of knowledge of the stego-function (the algorithm which is converting cover-image into stego-image) makes the process of determining the hidden data very difficult as we do not know in what form and in which part of the stego-image the data is hidden. This makes image and video steganography the most favorite tool among the terrorist networks. It is said that Al-Qaeda has been using it long before 9/11 suicide attacks (September 11, 2001) [2] [3] [4]. In the software mentioned in Fig. 1 the secret message is hidden using Digital Image as the cover object. Thus before proceeding to Image Steganography, the analysis of the Digital Image in detail is a prerequisite.

II. Digital Images

Every Digital Image consists of several smallest possible discrete picture elements known as Pixels. For better analysis of the digital image a very small 11 x 8 pixel 24 bit BMP Image [5] (image with 11 rows and 8 columns of pixels) has been created as shown in Fig 2.a. Since the image is too small for printing purposes so the same is enlarged by pixel spreading (Fig 2.a) and pixel enlargement (Fig 2.a). This image is analyzed using Matlab to produce detailed view of each pixel (shown as rectangular blocks) along with the row and column number of each pixel as given in Fig 2.b.

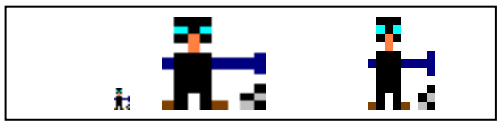

Fig 2.a 24 bit 11 x 8 Pixel BMP Image

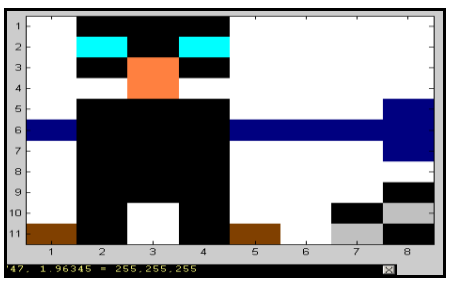

Fig 2.b Image analyzed using Matlab

Each pixel is made up of combination of three primary colors- Red, Green and Blue. It is the combination of the varying intensity levels of these three primary colors which produces all other colors in the digital image. In Fig. 3 the same 11 x 8 pixel image is shown as combination of three primary colors. If P(i,j) represents Pixel located at position (i,j) in the image. Then any image of dimensions m x n can be represented as matrix of pixels P(i,j) with i varying from 1 to m and j varying from 1 to n. Further each pixel P(i,j) is the combination of intensity levels of the Red, Green and Blue at the location (i,j) in the image. Also memory required for storing every pixel P(i,j) is the sum of memory required to store the intensity levels of Red, Green and Blue at the location (i,j). Mathematically every pixel P(i,j) = {R(i,j) ,G(i,j) ,B(i,j)} where R(i,j), G(i,j), B(i,j) represents the intensity levels of Red, Green and Blue colors at the location (i,j). In a 24 bit BMP image every pixel consumes 24 bits for storing the 8 bit color intensity levels of Red, Green and Blue. Hence, all color variations are derived from the combination of these three colors and their varying intensity levels represented by 8 bits (where $(00000000)_2$ corresponds to intensity of decimal 0 and $(11111111)_2 = 2^0+2^1+2^2+2^3+2^4+2^5+2^6+2^7$ corresponds to intensity level of decimal 255).

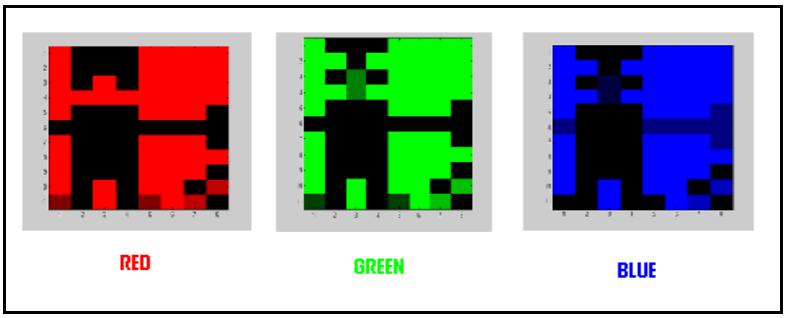

Fig 3 Image as Combination of Varying Intensity Levels of Three Primary Colors

Thus any pixel can be combination of 256 different shades (ranging from intensity level of 0 to intensity level of 255) of red, green and blue resulting in 256 x 256 x 256 or more than 16-million colors. Whole range of shades of Red, Green and Blue varying from intensity level 255 to intensity level of 4 are shown in Fig 4. Intensity level of 0 in all three colors appears Black.

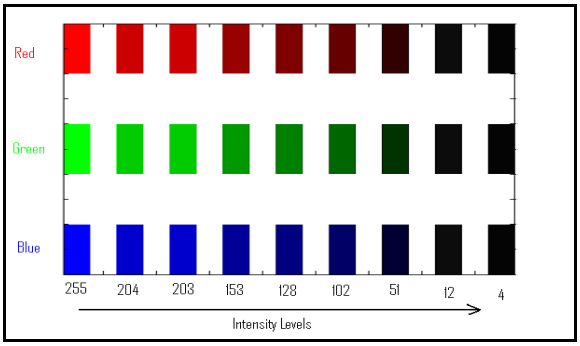

Fig 4 Range of Shades of Red, Green and Blue

Digital images are stored in a three-dimensional matrix format with row and column number corresponding to pixel position. There are 3 unsigned 8 bit values (these values are in Binary format but for ease of representation they are shown in equivalent Decimal) at each pixel position (row & column). In Table 1.a, Table 1.b and Table 1.c the same 11 x 8 pixel 24 bit BMP Image is shown as 3 two-dimensional matrices corresponding to Red, Green and Blue intensity levels respectively. For instance element present in the first row and first column of this image matrix (E(1,1)) contains 255,255,255 as its RGB values. So any element located at i^{th} row and j^{th} column of the image matrix will be now onwards represented by E(i,j) in this entire paper. Further every element E(i,j) contains 3 more elements R(i,j), G (i,j) and B(i,j) jointly known as RGB value of any arbitrary pixel (i,j).

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	255	0	0	0	255	255	255	255
2	255	0	0	0	255	255	255	255
3	255	0	255	0	255	255	255	255
4	255	255	255	255	255	255	255	255
5	255	0	0	0	255	255	255	0
6	0	0	0	0	0	0	0	0
7	255	0	0	0	255	255	255	0
8	255	0	0	0	255	255	255	255
9	255	0	0	0	255	255	255	0
10	255	0	255	0	255	255	0	192
11	128	0	255	0	128	255	192	0

Table 1.a Matrix R showing Red value of each Pixel Position
(Red Layer of the Image)

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	255	0	0	0	255	255	255	255
2	255	255	0	255	255	255	255	255
3	255	0	128	0	255	255	255	255
4	255	255	128	255	255	255	255	255
5	255	0	0	0	255	255	255	0
6	0	0	0	0	0	0	0	0
7	255	0	0	0	255	255	255	0
8	255	0	0	0	255	255	255	255
9	255	0	0	0	255	255	255	0
10	255	0	255	0	255	255	0	192
11	64	0	255	0	64	255	192	0

Table 1.b Matrix G showing Green value of each Pixel Position
(Green Layer of the Image)

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	255	0	0	0	255	255	255	255
2	255	255	0	255	255	255	255	255
3	255	0	64	0	255	255	255	255
4	255	255	64	255	255	255	255	255
5	255	0	0	0	255	255	255	128
6	128	0	0	0	128	128	128	128
7	255	0	0	0	255	255	255	128
8	255	0	0	0	255	255	255	255
9	255	0	0	0	255	255	255	0
10	255	0	255	0	255	255	0	192
11	0	0	255	0	0	255	192	0

Table 1.c Matrix B showing Blue value of each Pixel Position
(Blue Layer of the Image)

III. Image Steganography

We have just explored that in a 24 bit BMP image every pixel occupies 24 bits or 3 Bytes. Out of these 24 bits each of the three primary colors occupies 8 bits. These 8 bits carry the respective color intensity levels at any given pixel. Hence amount of color information carried by two least significant bits (LSB) is 4 color levels ((00000**00**), (00000**01**), (00000**10**) and (00000**11**) corresponding to intensity levels of 0, 1, 2 and 3 respectively) out of 256 possible color levels. Thus if the last two bits are changed the pixel will have minimal degradation of 4/256 or 1.5625%.This minor degradation is psycho-visually imperceptible. Marginal alteration of color intensity (from intensity level of 255 to 250) is shown in the Fig 5 which proves that such minor changes

are totally imperceptible to human eye due to inherent limitations in Human Visual System (HVS) [6].

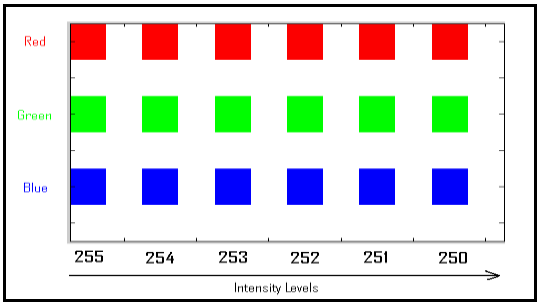

Fig 5 Marginal alteration of Intensity Levels

But at the cost of this negligible degradation we get 6 bits (2 bits each from red, green and blue) out of every pixel for transmitting our secret data. Although a general ASCII standard requires 7 bits for storing a character. But even with 6 bits we can produce (000000 = 0 to 111111=63) 64 symbols. In English language we have only 26 alphabets. So even after mapping every 6 bit code to a particular English alphabet (small letters) we are still left with 38 more 6 bit binary symbols. 36 out of them can be used for representing normal day to day common symbols like !@$%^&*(){}?,+= - " SPACE etc. While 2 of them (111110 and 111111) are reserved for two special purposes and here onwards will be referred as special symbols.

The encoding algorithm (algorithm for embedding data in cover-image) will operate over the image, pixel by pixel and embeds character in one pixel at a time. Once all the characters are embedded in the image the special symbol 111111 is inserted in the immediate next pixel (i.e. the pixel immediately next to the one containing the last input character. This helps the decoding algorithm (algorithm for extracting data from stego-image) by informing it to stop reading the image from next pixel onwards as no more data is embedded in further pixels. We call such a character *Terminator Character* `Tr' and represent it by 111111. Also due to limitations imposed by 6 bits we were unable to make provisions for capital letters. So we use a special character which when follows any normal character implies that it was in Capitals. We call it *Capitals character* 'Cl' and represent it by 111110. We are using our same old 11 x 8 BMP image for storing string *"hello world"*. The code used for representing these 9 characters (d, e, h, l,o, r, w, ", SPACE) is shown in Table 2. In Fig 6 the stego-image generated after embedding the string is shown as combination of 3 primary colors. This stego-image is shown as 3 two-dimensional matrices in Table 3.a, Table 3.b and Table 3.c correspond ing to red, green and blue layers respectively.

Character	**Binary Symbol**
d	000100
e	000101
h	001000
l	001001
o	001111
r	010010
w	010111
"	111100
SPACE	000000
`Tr' (Termination)	111111

Table 2 Characters and their Codes

Now if we compare Table 1 with Table 3 then we see that only first two rows have changed. For instance RGB values of element E(1,1) is modified from 255, 255, 255 (Table 1) into 255,255,252 (Table 3). This is because binary code of first character **"** of the *string "hello world" is* 111100 (Table 2). So values 11, 11, 00 are inserted as the last two LSB of R(1,1), G(1,1) and B(1,1) respectively. Since R(1,1) and G(1,1) already contained 255 or $(11111111)_2$ so no change has occurred in them. As G(1,1) also contained 255 but the value to be inserted in its LSB was 00 so the new value stored by G(1,1) is 252 or $(11111100)_2$. Similarly RGB value of E(1,2) is changed from 0,0,0 to 0,2,0 for embedding character `h' given by code 001000. Similarly all other characters of the string are embedded. Only first two rows have changed because only 14 characters are inserted: 13

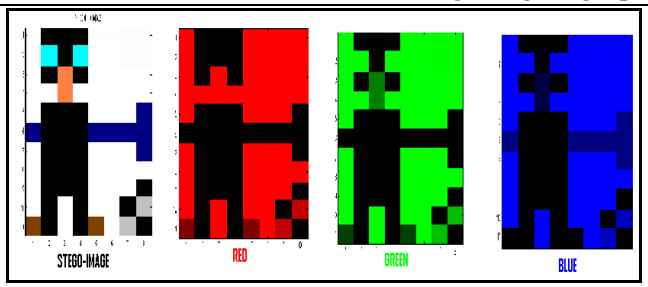

Fig 6 Stego-Image (string *"hello world"* embedded) as combination of Three Primary Colors

ROW								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	255	0	0	0	252	252	252	253
2	252	1	0	0	255	255	252	252
3	255	0	255	0	255	255	255	255
4	255	255	255	255	255	255	255	255
5	255	0	0	0	255	255	255	0
6	0	0	0	0	0	0	0	0
7	255	0	0	0	255	255	255	0
8	255	0	0	0	255	255	255	255
9	255	0	0	0	255	255	255	0
10	255	0	255	0	255	255	0	192
11	128	0	255	0	128	255	192	0

Table 3.a Matrix R' showing Red value of each Pixel Position in the Stego-Image containing "hello world"

ROW								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	255	2	1	3	255	255	252	253
2	255	252	3	253	255	255	252	252
3	255	0	128	0	255	255	255	255
4	255	255	128	255	255	255	255	255
5	255	0	0	0	255	255	255	0
6	0	0	0	0	0	0	0	0
7	255	0	0	0	255	255	255	0
8	255	0	0	0	255	255	255	255
9	255	0	0	0	255	255	255	0
10	255	0	255	0	255	255	0	192
11	64	0	255	0	64	255	192	0

Table 3.b Matrix G'showing Green value of each Pixel Position in the Stego-Image containing "hello world"

ROW								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	252	0	1	0	252	255	252	255
2	255	254	0	252	252	255	252	252
3	255	0	64	0	255	255	255	255
4	255	255	64	255	255	255	255	255
5	255	0	0	0	255	255	255	128
6	128	0	0	0	128	128	128	128
7	255	0	0	0	255	255	255	128
8	255	0	0	0	255	255	255	255
9	255	0	0	0	255	255	255	0
10	255	0	255	0	255	255	0	192
11	0	0	255	0	0	255	192	0

T**a**ble 3.c Matrix B' showing Blue value of each Pixel Position in the Stego-Image containing "hello world"

(length of "hello world") + 1 (Terminator Character). Each row contains 8 elements so only two rows are consumed for embedding data. At 14th pixel (in row major order) represented by element E(2,6) the terminator character (given by code 111111) is embedded after which character insertion stops from next row onwards .

IV. Algorithm for Steganography

Any pixel P(i,j) consists of 3 eight bit RGB values R(i,j), G(i,j) and B(i,j). For insertion of data we are modifying only the last two LSBs of this eight bit number. Let this eight bit number be represented by **K** in decimal format and $k_8k_7k_6k_5k_4k_3k_2k_1$ in binary format. Hence $\mathbf{(K)}_{DECIMAL}$ or simply $\mathbf{K} = (k_8k_7k_6k_5k_4k_3k_2k_1)_2$ is one of the RGB values for any arbitrary pixel P(i,j). Mathematically $k_1 = \mathbf{K}$ modulo 2 and $k_2 = (\mathbf{K}/2)$ modulo 2. For embedding the character the two LSBs k_2 and k_1 are replaced by 2 other bits c_2 and c_1 respectively as per the character's code (Table 2).

This algorithm is based on a simple principle of number theory. When any natural number is added or subtracted by 1 then the least significant bit of its binary equivalent gets flipped. Also when a natural number is added or subtracted by 2 then the second least significant bit of its binary equivalent gets flipped. We can extract k_1 and k_2 from **K** by performing modulo operation on **K**. Once we know k_1 and K_2 then they can be compared with c_1 and c_2 and accordingly action may be taken. If they are same no action is required. Else for flipping k_2 we need to subtract 2 from **K** and for flipping k_1 we have to subtract 1 from **K**.

Although subtraction will serve our purpose but this will shift the RGB intensity of the pixels in only one direction (negative side). Which means the new RGB value for the pixel P(i,j) = {R(i,j),G(i,j),B(i,j)} after embedding the data can be one of these R(i,j), R(i,j)-1, R(i,j)-2, R(i,j)-3 (Possible R values); G(i,j), G(i,j)-1, G(i,j)-2 , G(i,j)-3 (Possible G values) and B(i,j), B(i,j)-1, B(i,j)-2 , B(i,j)-3 (Possible B values). This negative shift of R, G and B levels may make the stego-image relatively easier to be identified by the steganalyst. So for harboring the benefit of obscurity by randomization we should simultaneously use addition also. The combination of addition and subtraction will more randomize the image and make it more difficult to steganalyse. By using combination of addition and subtraction the possible RGB values for Pixel P(i,j) are R(i,j)-1, R(i,j)-2, R(i,j)-3, R(i,j), R(i,j)+1, R(i,j)+2, R(i,j)+3 (Possible R values); G(i,j)-1, G(i,j)-2, G(i,j)-3, G(i,j), G(i,j)+1, G(i,j)+2, G(i,j)+3 (Possible G values) and B(i,j)-1, B(i,j)-2, B(i,j)-3, B(i,j), B(i,j)+1, B(i,j)+2, B(i,j)+3 (Possible B values).

Also the addition in the intensity level of one color and simultaneous subtraction of the intensity level of another color at the same pixel will conserve the brightness of the pixel. This is because the gray scale intensity of the colored image is obtained by weighted average of its R, G and B values. As we know that for gray scale conversion the weight of G is highest among R, G, B so we will get the best results if we use subtraction in R and B and addition in G.

But imprudent use of addition or subtraction for LSB replacement may cause error in certain bright or dark pixels by either raising the color intensity levels above 255 or reducing it below 0 respectively. To overcome such limitations an algorithm is developed which performs addition by 1 when k_1 is 0 and addition by 2 when k_2 is 0. On the other hand it subtracts by 1 when k_1 is 1 and subtracts by 2 when k_2 is 1. The pseudo-code of this algorithm for LSB replacement is given in the boxes below. It is based on two IF-ELSE control structures. Box 1 explains how c_2 replaces k_2 and Box 2 explains how c_1 replaces k_1.

```
IF (k2 = = 1)
        IF (c2 = = 0)
                K = K-2;
        ENDIF

ELSE
        IF (c2 = = 1)
                K = K+2;
        ENDIF
ENDIF
```

Box1 LSB Replacement of c_2

```
IF (k1 = =1)
        IF (c1 = = 0)
                K = K-1;
        ENDIF

ELSE
        IF (c1 = = 1)
                K = K+1;
        ENDIF
ENDIF
```

Box2 LSB Replacement of c_1

V. Performance Evaluation

Based on above algorithm, software was developed in Matlab. It was compared with three other free steganography softwares available on the Internet (Our Secret, Eureka Steganographer and Quick Stego). Then all four of them were examined using StegSpy 2.1 (the only steganalytic tool easily and feely available on the net downloaded from location http://www.spy-hunter.com/stegspydownload.htm). It claimed to identify signatures of well acclaimed stego-tools like Hiderman, JPHideandSeek, Masker, JPegX and Invisible Secrets. But it

was unable to identify any data in the stego-images generated by all the four test software. On the contrary it found a hiderman's signature in one of the cover image itself (a world map obtained from internet). The preliminary comparison results are summarized in Table 4. This table also contains the web source from where these softwares were downloaded.

Among all the four softwares the Eureka steganographer behaved in most weird manner. As mentioned in Table 4 using it we can embed 25 characters in a 100 pixel image but whenever more than 19 characters were inserted it produced an unrecognized file format (from which we can retrieve data). But this number (19 characters) was not fixed and varied with every trial. In certain trials it produced an unrecognizable stego-file even when 1 character was inserted in the same 100 pixel image.

5.1. Susceptibility to Steganalysis

As mentioned in the Introduction the Steganalysis is a difficult task but still we cannot completely rule out its probability. Probabilistically every stego-image is susceptible to such an attack. To minimize this probability we have to keep differences between the cover-image and stego-image very smooth and as small as possible.

For determining the susceptibility to steganalysis these four stego-tools were evaluated by three different techniques. In all these three test runs the difference between the cover-image and stego-image was calculated using various statistical parameters. In first two modes our same old 11 x 8 pixel BMP image was used for embedding the data. In first case only 4 characters (*abcd*) were inserted in it whereas in the second case the stego-softwares were evaluated at full capacity. I.e. maximum possible characters which these tools allowed to insert in this 11 x 8 pixel image. In third case a large image (300 x 300 pixel BMP image) was used for embedding 600 characters (*abc...z1234 in repetition 20 times*).

5.1.1 Parameters for Measurement of Susceptibility to Steganalysis

The various parameters used for calculating the difference between the cover-image and stego-image are Mean Difference (MD), Mean Absolute Difference (MAD), Standard Deviation of the Difference (STD) and Range of Difference ($Z = Z_{upper} - Z_{lower}$), no of pixels changed (P) and change in the size of the image (in terms of memory occupied) K. These parameters are explained below.

Let cover-image be represented by C and stego-image be represented by S. Then cover image C and stego-image S consists of three RGB components C_R, C_G, C_B and S_R, S_G, S_B respectively. Thus the difference image D = C – S also consists of three RGB components $D_R = C_R – S_R$; $D_G = C_G – S_G$ and $D_B = C_B – S_B$. Similarly all these statistical parameters have three components corresponding to R, G and B.

Three components of Mean Difference MD given as MD_R, MD_G and MD_B are arithmetic means of D_R, D_G and D_B respectively. But this parameter does not gives us the exact idea of the difference between the two images as the negative and positive values in the matrix D_R ,D_G and D_B cancel each other. To overcome this drawback another parameter Mean Absolute Difference (MAD) is introduced. For three color components it is represented as MAD_R, MAD_G and MAD_B and corresponds to arithmetic mean of the absolute values of quantities in D_R, D_G and D_B respectively. But in order to determine the degree of abrupt variations in RGB components of cover-image C and stego-image S the Standard Deviation (STD) of the difference matrices D_R, D_G and D_B is

Name of the Steganography Software	**Algorithm (Section 4)**	**Our Secret**	**Eureka Steganographer (Stegan2)**	**QuickStego by Quick Crypto**
Image Type	BMP 24 bit	All file types(even doc, text & media files) except .TIFF	All common image types except .TIFF	BMP , JPEG , JPG and GIF
Interface	CUI	GUI	GUI	GUI
Minimum Image Size required for storing a single character (For BMP Image)	2 Pixel Image or 62 Bytes	1 Pixel Image or 58 Bytes.	59 Pixels or 230 Bytes	75 Pixels or 282 Bytes

Number of character stored in 24 Bit BMP image of		lower case char	Upper case char	Case of the character makes no difference	Case of the character makes no difference	Case of the character makes no difference
	100 Pixels	99	49	Infinite characters	25 characters	9 characters
	1000 Pixels	999	499	Infinite characters	607 characters	331 characters
	10000 Pixel	9999	4999	Infinite characters	6411 characters	3516 characters
Insertion Algorithm		2 bit LSB insertion (SDT)		Uses file systems.		1 Bit LSB Insertion (SDT)
Other Features				Other then characters it can embed all types of files also. Supports data encryption.	Supports data encryption.	
Comments				It can store large file or thousands of characters in 1 pixel by increasing the memory size of the pixel. Crashes after few runs.	If information compared to image is very long (Eg: In 100 pixel image if more than 19 characters inserted) then the stego-image generated is of unrecognized format and is not readable.	Crashes after few runs.
Steganalysis by StegSpy 2.1		Could not detect		Could not detect	Could not detect	Could not detect
Web Source		Not Uploaded Yet		http://www.securekit.net/oursecret.htm	http://www.brothersoft.com/eureka-steganographer-v2-266233.html	http://quickcrypto.com/free-steganography-software.html

Table 4 Preliminary Comparative Study of Different Stego Tools

computed and represented as STD_R, STD_G and STD_B. The STD is measure of sudden sharp changes between C and S (Spikes or Grains in difference image D = C - S).

In addition we have used other parameters like Range of difference in D given by Z_R, Z_G and Z_B where $Z_R = Z_{R\ upper} - Z_{R\ lower}$. $Z_{R\ upper\ is}$ maximum value in D_R and $Z_{R\ lower}$ is minimum value in D_R. Similarly we can compute Z_G and Z_B. We have incorporated one more parameter to measure the spread of the difference and it is given by no of pixels changed in R, G and B components of the image and given as P_R, P_G and P_B. Parameter K is used for determining the difference between the memory size of C and S.

5.1.2 Results of three Trials

These parameters as outcomes of the three trials are illustrated in Table 5.a, Table 5.b and Table 5.c. From these tables it can be very clearly deduced that for N character input algorithm (Section 4) modifies N+1 pixels, Eureka steganographer tentatively modifies 0.3353N + 1.8096 pixels whereas Quick Stego modifies 1.534N +

39.5963 pixels. Further it's worth mentioning that Oursecret's algorithm does not modify any pixel but it increases the size (in terms of memory) of cover-image file. Thus we can see that Eureka concentrates the entire information in few pixels where as Quick Stego and algorithm (Section 4) evenly distributes the information in larger number of pixels. This explains why Eureka steganographer has highest Standard Deviation STD and Range Z. As a result of this high STD and Z, the stego-images generated by Eureka has grains and spikes as shown in Figure 7.

For all other three softwares (Algorithm mentioned in Section 4, Quick Stego and Our Secrets) the difference between Cover Image and Stego Image is so low that it is visually imperceptible to find any change. So the difference between the two can only be established using the statistical parameters mentioned in section 5.1.1 Further the range Z of Quick Stego is always fixed between -1 and 1. This implies that it uses 1 bit LSB insertion. Since the algorithm mentioned in section 4 uses last 2 bit LSB insertion so the overall range lies is between -3 and 3. Further we had calculated (Section 3) that the maximum degradation which may occur in the image after 2 bit LSB insertion will be 1.5625%. But the real error was much lower. The average of the Mean Absolute Diffrences (MAD) for the algorithm mentioned in section 4 in all the three tables i.e. Table 5.a, Table 5.b and Table 5.c came out to be 0.565244 intensity levels out of 256 possible intensity levels. In percentage

Software		**MD**	**MAD**	**STD**	**Z_{lower}**	**Z_{upper}**	**Z**	**P**	**K**
Algorithm (Section 4)	**Red**	0.1364	0.1364	0.6285	0	3	3	5 Pixels	0
	Green	0.1250	0.1477	0.6399	-1	3	4	5 Pixels	
	Blue	0.0682	0.1818	0.7078	-3	3	6	5 Pixels	
Our Secret	**Red**	0	0	0	0	0	0	0 Pixels	212 Bytes
	Green	0	0	0	0	0	0	0 Pixels	
	Blue	0	0	0	0	0	0	0 Pixels	
Eureka Steganographer (Stegan 2)	**Red**	4.7273	4.7273	28.1750	0	217	217	3 Pixels	0
	Green	- 0.2614	3.7841	21.8519	-118	155	273	3 Pixels	
	Blue	5.8409	5.8409	31.4134	0	193	193	3 Pixels	
QuickStego by Quick Crypto	**Red**	0.3068	0.5114	0.6496	-1	1	2	45 Pixels	0
	Green	- 0.0682	0.4773	0.6914	-1	1	2	42 Pixels	
	Blue	-0.0795	0.3523	0.5915	-1	1	2	31 Pixels	

Table 5.a 4 Characters Inserted in 11 x 8 Pixel Image

Software		**MD**	**MAD**	**STD**	**Z_{lower}**	**Z_{upper}**	**Z**	**P**	
Algorithm (Section 4) (87 Characters)	**Red**	0.9886	1.6023	1.7053	-3	3	6	88 Pixels	0
	Green	0.1818	1.5227	1.8417	-3	3	6	88 Pixels	
	Blue	0.0909	1.4318	1.8044	-3	3	6	88 Pixels	
Our Secret (300 Characters)	**Red**	0	0	0	0	0	0	0 Pixels	252Bytes
	Green	0	0	0	0	0	0	0 Pixels	
	Blue	0	0	0	0	0	0	0 Pixels	
Eureka Steganographer (Stegan 2) (18 Characters)	**Red**	4.0568	10.0114	40.2893	-162	217	379	8 Pixels	0
	Green	1.3977	10.1705	39.6492	-207	193	400	8 Pixels	
	Blue	4.3409	8.9091	37.3867	-110	246	356	7 Pixels	
QuickStego by Quick Crypto (5 characters)	**Red**	0.3182	0.5455	0.6703	-1	1	2	48 Pixels	0
	Green	-0.0795	0.5114	0.7147	-1	1	2	45 Pixels	
	Blue	-0.0909	0.3636	0.5995	-1	1	2	32 Pixels	

Table 5.b Full Capacity Evaluation of Softwares Using 11 x 8 Pixel Image

Software		MD	MAD	STD	Z_{lower}	Z_{upper}	Z	P	
Algorithm (Section 4)	**Red**	0.0112	0.0139	0.1737	-2	3	5	601 Pixels	0
	Green	0.0057	0.0132	0.1664	-3	3	6	601 Pixels	
	Blue	0.0032	0.0131	0.1677	-3	3	6	601 Pixels	
Our Secret	**Red**	0	0	0	0	0	0	0 Pixels	252 Bytes
	Green	0	0	0	0	0	0	0 Pixels	
	Blue	0	0	0	0	0	0	0 Pixels	
Eureka Steganographer (Stegan 2)	**Red**	-0.1508	0.1994	5.2088	-240	198	438	201 Pixels	0
	Green	-0.0972	0.1880	4.8514	-239	193	432	203 Pixels	
	Blue	-0.2655	0.2818	7.1111	-253	149	402	202 Pixels	
QuickStego by Quick Crypto	**Red**	0.0038	0.0103	0.1012	-1	1	2	923 Pixels	0
	Green	-0.0020	0.0103	0.1013	-1	1	2	924 Pixels	
	Blue	-0.0043	0.0074	0.0860	-1	1	2	668 Pixels	

Table 5.c 600 Characters Inserted in 300 x 300 Pixel Image

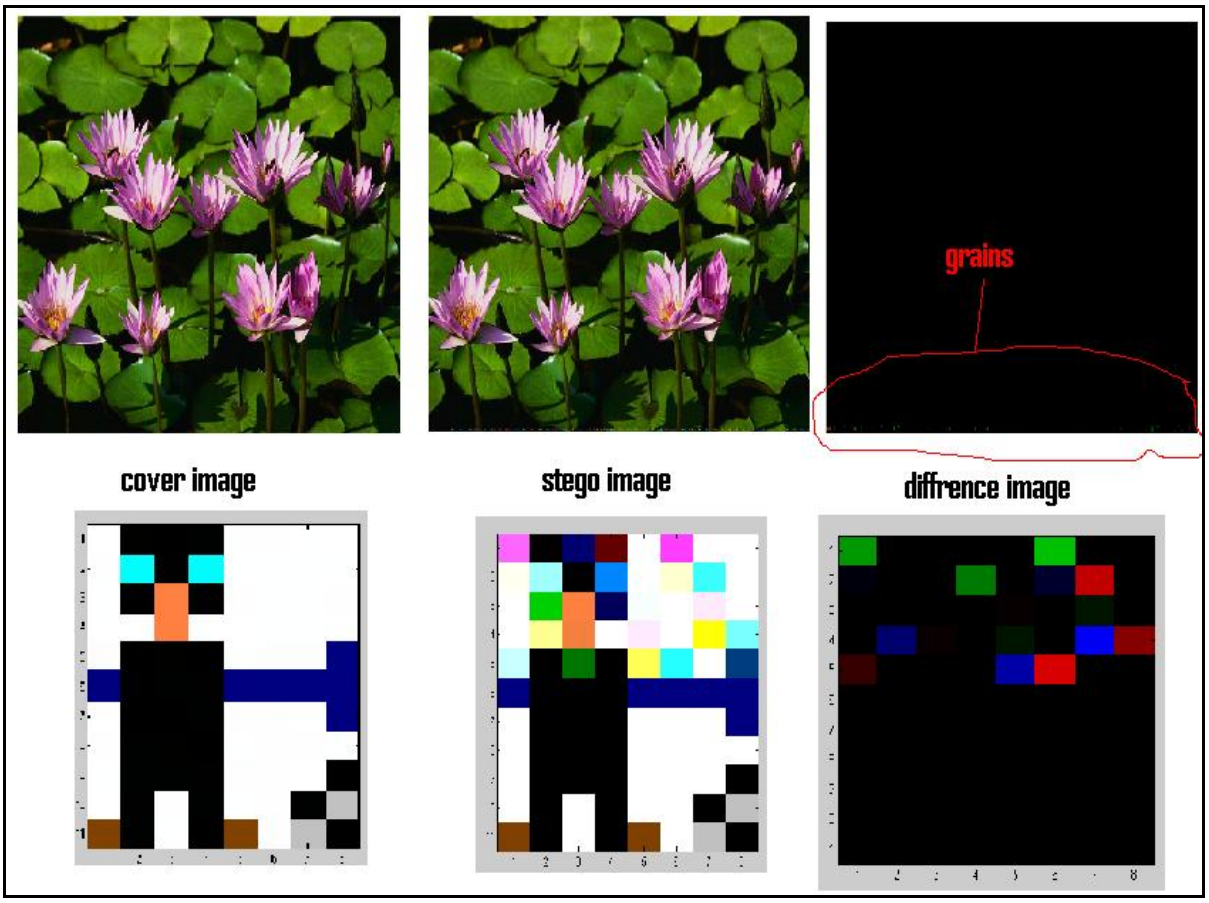

Fig 7 Grains in the Stego-Image and Difference-Image using Eureka Steganographer

terms this difference was just 0.22701% which is much lower than 1.5625%. Similarly for Eureka Steganographer average MAD was found out to be 4.901389 intensity levels or 1.9146% difference whereas for Quick Stego it was just 0.30994 Intensity Levels or 0.12854% diffrence.

Thus we see that Quick Stego has less data embedding capacity and it is least susceptible to general steganlytic attack among all four. Our Secret does not modify the image because it changes the memory allocation in the file system but it is susceptible to steganalytic attack as the file size increases. Algorithm mentioned in section 4 has large information carrying capacity but it is more susceptible to attacks then Quick Stego. Eureka is always an easy prey for most steganalysts.

5.2 Steganlysis

Before we begin with steganalysis we should know various communication channels and techniques through which these stego-images are being used by terrorist organizations and secondly the reasons why they are relying on it so much.

5.2.1 Channels and Techniques used by Terrorist organizations

As mentioned in the introduction, for any communication channel meeting CIA Compliance is must. To achieve it terrorists are using stego-images for one to one communication via e-mails. On the other hand these stego-images are used as Electronic Dead Drops [7][8][9][10]. Dead drop is a Cold War-era slang connoting a place (like a common letter box or a common book in a public library) where spies left information. It helped the two communicating parties to avoid direct meeting or knowing each other. Thus it provided advantages of anonymity. On similar lines the terrorists upload stego-images in a public forum (eg Online Auction site E-Bay) and thus keep communicating without knowing each other. [7][11][12][13]. There does not exist any coordination, meetings, E-mail or instant messaging between them. All that exists is a picture posted to a public forum, and then downloaded by anyone sufficiently enticed by the subject line of the image (even third parties can download the image). But images posted in the forums not only serve as electronic dead drops but can also be used for broadcasting the secret message globally to all the agents operating in different geographical regions. Thus it facilitates completely anonymous and asynchronous communication. Also these stego-images are used by terrorists for safely carrying and storing secret information hidden in normal appearing videos and Images in their personal computers and memory cards[1][2].

5.2.2 Advantages offered by Steganography

A legitimate company never needs to use dead drops. But the terrorists need it. Thus the very aspect of secrecy makes it a potent tool in terrorists' hand. The main reason for using steganography is the cost of inserting the data in the image is far lower than the cost of detecting even the presence of data in the same-stego image. In some cases it may even be impossible to detect any presence of data. For instance, a one bit, "yes" or "no" message embedded in a big image file would be nearly impossible to find [14]. Moreover these image files could be easily converted into video files and collages and can make detection even more difficult.

The smaller the size of information, the harder it is to find. This can be numerically established by comparing MAD and STD fields of Table 5.a with Table 5.b. The average Standard Deviation of the Difference (STD) for all three softwares (our secret is not considered because it has different insertion mechanism) is 9.48332 intensity levels when only 4 characters are embedded (Table 5.a) whereas it is 13.85123 intensity levels when all the three softwares were evaluated at full capacity (inserting 36.67 characters on an average) (Table 5.b). For the same two sets of observation the average Mean Absolute Difference (MAD) is 1.795467 intensity levels and 3.896478 Intensity Levels respectively.

Further the lack of knowledge of insertion algorithm makes detection even difficult. Even after detection the extraction of data is another herculean task. For instance algorithm mentioned in section 4 inserts a character at every pixel by using 6 bits in each pixel. But use of 6 bits for representing a character is not a standard ASCII method for denoting a character. Further this algorithm reads the pixel continuously in row major order. Some other algorithm may do it by combining Row Major, Column Major and even diagonal order and may read pixels in any other pattern.

Most alarming is the fact that this algorithm (section 4) is based on one of the simplest image steganographic techniques (Spatial Domain Steganography based on changing the pixels through LSB Insertion) and still it can have so many variants. But the real steganographic algorithm used by terrorists is much more complex and will be using combination of multiple steganographic techniques like Transformation Domain Steganography[15] (Discrete Cosine Transforms as used in JPEG Images, Discrete Wavelet Transform or even Fast Fourier Transform for embedding information in the transform domain of the image signal), Spread Spectrum Steganography [16] (evenly spreading the information over entire image i.e. reducing the Steganographic Signal to Noise Ratio to minimum possible levels) or may use a different color space itself (Example RGB may be converted to YCbCr) and various other techniques [17].

Another difficulty in image steganography is analysis of the image. This requires various mathematical and statistical operations to be performed on large three dimensional image matrices. This consumes very high computation power and is almost impossible to be performed for every image in the cyber space.

5.2.3 Steganalysis

Identifying a stego-image using visual clues is almost impossible. Among all the four test softwares only the Eureka software produced visually perceptible grains. That too we could locate only after subtracting the stego-image from cover-image. Also in section 5 we saw that steganlysis software Steg-spy 2.1 was unable to detect any presence of information in any of the trials. This was because it was designed for detecting particular steganographic signatures which was considerably different from the one used in any of the four softwares. Unfortunately we have plethora of steganographic softwares [18] and algorithms [17] and thus detection becomes even more difficult.

Most steganalysis algorithms can be broadly classified in to three types. Passive Steganalysis (only detection of steganographic signatures), Active Steganalysis (detection followed by extraction of data from the stego-image) and Active Warden (detection followed by destruction of the hidden information or embedding counter-information) [19].

As mentioned in section 5.2.1 these stego-images are either being uploaded in the Public forums or they are being used for one to one communication through e-mails and by carrying them personally. If public forums are used then web crawlers [20] can be used for associating every image on the Internet with its vital statistical and mathematical parameters. So if any of those parameters change then the crawler will alert the steganalyst. But the terrorists are equally smart. For every part of the secret message they must be using a different and every time a new image. So these crawlers have to be combined with image processing, statistical and mathematical tools for actively steganalysing every image on the net [13][21]. For the images used in one to one communication either through emails or obtained from any other suspicious source reliance can only be on active steganalysis based on probabilistic, statistical and mathematical models.

For analysis of such suspicious images we can use various methods like Brute Force Attacks, Pattern Matching, Stir Mark Attack [7], Statistical Attack [21], Transform Domain Attacks and many more. Brute force method uses pixel by pixel analysis of the image and hence consumes very high computational power and time. But still it is highly unreliable. On the contrary use of statistical method like Kurtosis Analysis, Histogram Analysis and Neighbourhood Analysis of the pixels though marginally speeds up the steganalysis process but leads to frequent False Alarms and hence though speeds up the steganalysis process but reliability is still not achieved. Further the probabilistic methods rely on the fact that Image Matrices and Hidden Information are not purely random numbers (in any natural image the pixels do not change abruptly and colour of any pixel is dependent on the colour of the neighboring pixels thus they are auto-correlated moreover the data inserted is never random and modifies the image in a fixed pattern) and hence pattern matching algorithms can detect the presence of data in the image [22]. Further one of the best and most successful attack technique Stir Mark Attack [7] uses a series of mathematical transformation (assuming the data and image is not random) and thus amplifies the defects in the image. In the case of innocent image though it suffers degradations but still its edges are visually perceptible. Whereas the stego-image (which already had initial distortions) gets completely degraded to noise and becomes unidentifiable.

Also some steganalysis techniques use parameters like changes in size, file format, last modified timestamp and changes in the color palette of the image. Since every digital image has a unique digital file signature (Hex Signature and ASCII Signature) and any modification in the image will change these signatures. StegySpy 2.1(software used for steganalysis in section 5) relies on this method [23] but we have seen that it failed to identify any stego-signatures in all the four test softwares used.

In order to understand the difficulties associated with these steganalytic techniques, one such technique based on Row Gradient Profile of the image (Fig 8) is discussed in brief. It is a combination of pattern matching and statistical process and is a type of Neighborhood Analysis of the pixel.

It analyzes the inter row difference of the intensity levels of the pixel in the image also known as Row Gradient Profile or Simply RG Profile of the image. The image used here is of 60 x 130 and 1080 characters are stored in it using algorithm developed in Section 4. Since each character is stored in 1 pixel so first 18 rows of the image contain data. This inter-row difference or Row Gradient of the image is plotted.

As we can see that parameters of Cover-Image (Upper Graph) and Stego Image (Lower Graph) are clearly very different on the left of the black line (till 18^{th} Row of Image) and exactly identical on the right of the black line. But in real life scenario we will have only Stego Image from which we can obtain only lower graph. By using this lower graph only, it will be very difficult to conclusively say that image is a malicious stego image.

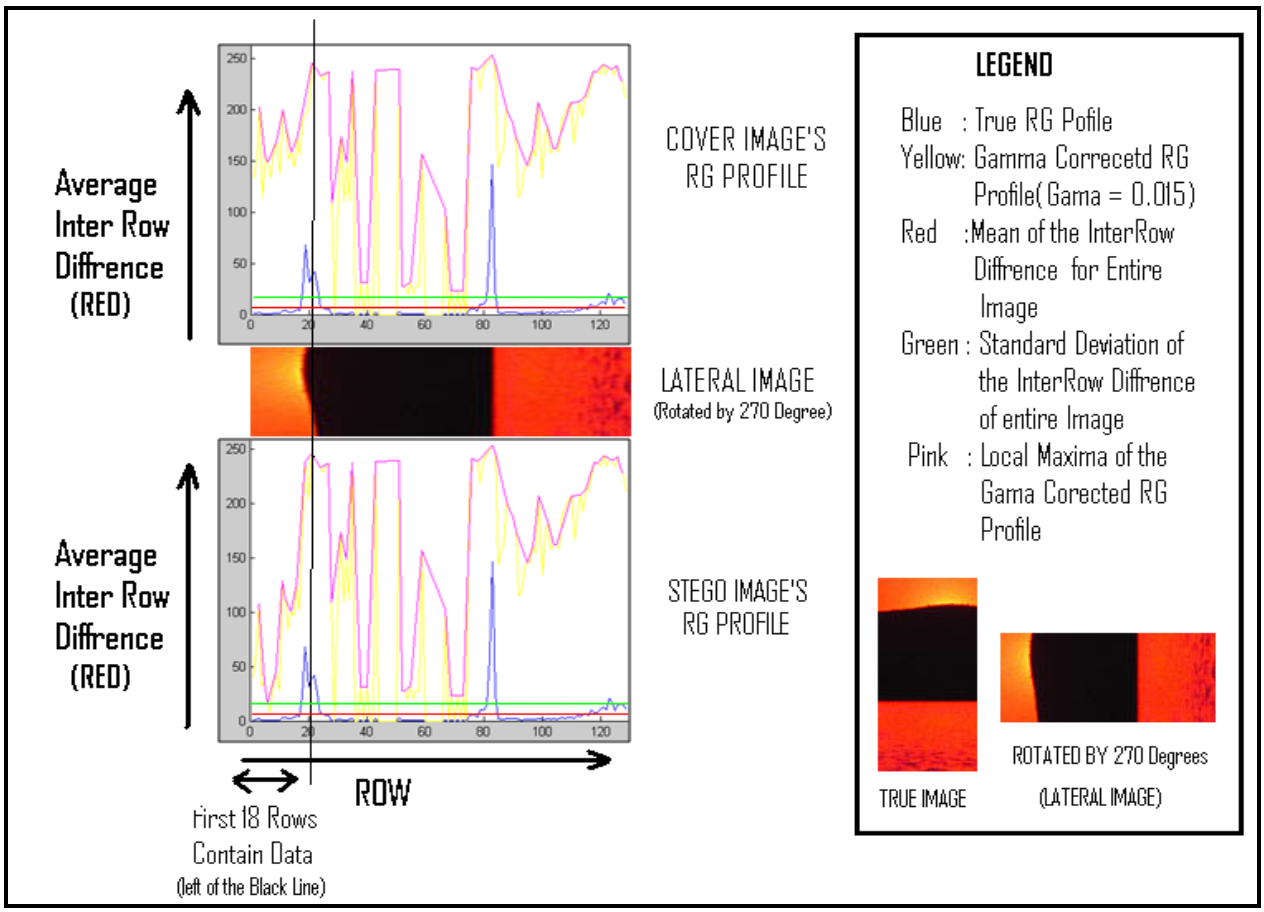

Fig 8 Row Gradient Profile of 60 x 130 Images

VI. Conclusion

This paper provides a macro view of Steganography and Steganalysis and analyses the methods and reasons the terrorists are relying on it. Even a simple steganographic algorithm can hide data so efficiently that steganalysis become very difficult on various accords of time, computation power and money.

But the real world terrorism uses more complex and innumerable steganographic algorithms. Till date we do not have any foolproof method of steganalysis. Considering the volume and size of information transferred over the web and the cost and time for steganalysis, it becomes virtually impossible to keep track of every file. Moreover most of the countries which are either centers of terrorism or its victims are second and third world countries (Fig 9) and do not have any dedicated and efficient system for policing such mammoth volumes of data flow. Interestingly some of these countries (mainly located in South East Asia and Middle East) have immense technical potential and vast Information Technology base but still most of them do not have any dedicated cyber security agency.

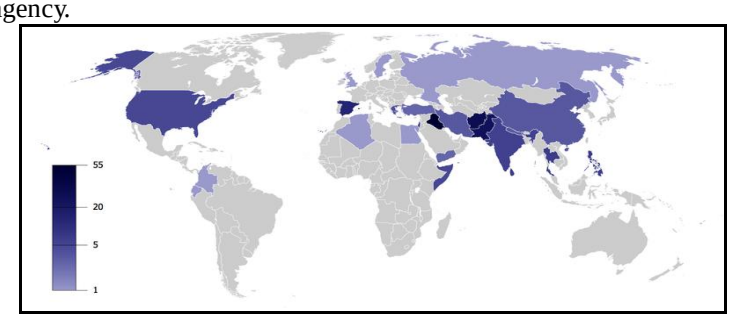

Fig 9 Centers of Terrorism (of 2009 from Wikipedia)

Although Police Forces in most of the developing nations have started cyber cells for handling various cyber crimes and most of the strategic departments in emerging countries have established base, command and ministry level Information Technology Offices but their sole purpose is maintenance of information security in

their own respective departments. None of them is dedicated for monitoring the internet traffic.This inherent weakness gives the terrorist leader the luxury to safely hide in countries where cyber-security is lackadaisical and inadequate.From such countries he keeps uploading malevolent images containing terrorist information and his Agents worldwide visit those concerned websites and keep getting his orders by just one click on the image. On the one hand this technique hides the location of his agents and on the other hand it also provides the advantages of dead-dropping, anonymity and secure broadcasting worldwide. In this regard it is noteworthy to mention that soon after establishment of US Cyber Command (USCYBERCOM) the countries like China, Britain, NorthKorea and South Korea have already begun preparing their cyber force against international cyber space (including the ones of the terrorists as well as those of their enemies). But the situation is relatively grim in most other countries in general and developing countries in particular not just because of lack of technical knowhow, man power or money but also due to poor foresight. Hence the need of the hour is to take enduring and vigorous measures against Cyber Terrorism and at the same time experts from multiple disciplines and countries must join hands to fight this menace to make world a safer place.

Notes and References

[1]. Infosecurity Magazine article dated 02 May 2012 reports that Al-Qaeda uses Steganography to hide documents.
http://www.infosecurity-magazine.com/view/25524/alqaeda-uses-steganography-documents-hidden-in-porn-videos-found-on-memory-stick
[2] Daily Mail Online, UK article dated 01 May 2012 reported that a Treasure trove of Intelligence was embedded in porn video.
http://www.dailymail.co.uk/news/article-2137848/Porn-video-reveals-Al-Qaeda-planns-hijack-cruise-ships-execute-passengers.html#ixzz1uIgxpire
[3].The New York Times article dated Oct 30, 2001 with title "Veiled Messages of Terror May Lurk in Cyberspace" claims 9/11 attacks planned using Steganography.
[4]Wired article dated 02nd July, 2001 nicknamed Bin Laden as "the Steganography Master"
http://www.wired.com/politics/law/news/2001/02/41658?currentPage=all
[5] In image processing the image matrix is represented as Columns by Row. Thus as per strict computer terminology this image will be called as 8 x 11 pixels image. But since this paper is meant for the readers of all backgrounds so we are following conventions of mathematics and hence we denote the image as Row by Column.
[6] William Stallings- *Cryptography and Network Security, Principles and Practices (5th Edition, Section 2.5,Pg 53).*
[7] Jonathan Watkins, *Steganography - Messages Hidden in Bits (15th December,2001)*
http://mms.ecs.soton.ac.uk/mms2002/papers/6.pdf
[8] krypt3ia article dated 13th March 2010 with the title "Al-Qaida goes "Old School" With Tradecraft and Steganography" .
http://krypt3ia.wordpress.com/2010/03/13/al-qaida-goes-old-school-with-tradecraft-and-steganography/
[9]Fabien A. P. Petitcolas,*Information Hiding 5th International WorkshopOct 2002,Netherlands (Pg 20, Volue V)* mentions FBI Agent Robert Hanssen using
steganography as electronic dead drop tool.
http://books.google.co.in/books/about/Information_Hiding.html?id=jCKL2tSwwVMC&redir_esc=y
[10] Peter Wayner - *Disappearing Cryptography: Information Hiding :* Steganography *& Watermarking Second Edition (Pg 413)* also mentions use of steganography as electronic dead drop tool.
http://dl.acm.org/citation.cfm?id=560589
[11] Ebay Online Auctions (www.ebay.com)
[12] Johnson, Neil.F and Jajodia, Sushil. *Steganalysis: The Investigation of Hidden Information, IEEE Information Technology Conference, Syracuse, New York, USA, September 1998*
http://akachuckie.com/text_files/SORTED/Hacking_and_Security_Guides/stega.pdf
[13] Niels Provos and Peter Honeyman, Center for Information Technology Integration University of Michigan, *Detecting Steganographic Content on the Internet.* The study of more than two million images downloaded from eBay auctions shows evidence that terrorists are using the images to hide encoded messages.
http://www.citi.umich.edu/techreports/reports/citi-tr-01-11.pdf
[14] SANS Institute InfoSec Reading Room
http://www.sans.org/reading_room/
[15] K B Shiva Kumar, K B Raja, Sabyasachi Pattnaik paper titled "Hybrid Domain in LSB Steganography" International Journal of Computer Applications, Volume 19– No.7, April 2011
[16] Lisa M. Marvel, Charles G. Boncelet, Jr. and Charles T. Retter, *Spread Spectrum Image Steganography, IEEE Transactions on Image Processing, Vol. 8, No. 8, August 1999*
http://www.fim.uni-linz.ac.at/lva/Rechtliche_Aspekte/2001SS/Stegano/leseecke/spread%20spectrum%20image%20steganography.pdf
[17] Stefan Katzenbeisser, Fabien A. P. Petitcolas, *Information hiding techniques for steganography and digital watermarking chapter 3 Survey of Steganographic Techniques*.
http://www.jjtc.com/pub/book99_ih.htm
[18] Pedram Hayati1, Vidyasagar Potdar2, and Elizabeth Chang *A Survey of Steganographic and Steganalytic Tools for the Digital Forensic Investigator*
http://www.pedramhayati.com/images/docs/survey_of_steganography_and_steganalytic_tools.pdf
[19] R. Chandramouli, *A Mathematical Approach to Steganalysis*
http://www.ece.stevens-tech.edu/~mouli/spiesteg02.pdf
[20] A Web crawler is a computer program that browses the World Wide Web in a methodical, automated manner or in an orderly fashion. They are mainly used by serach engines for creating a copy of pages for later processing by a search engine that will index the pages to provide fast searches.http://en.wikipedia.org/wiki/Web_crawler.
[21] Angela D. Orebaugh George Mason University, *Steganalysis: A Steganography Intrusion Detection System*
http://www.securityknox.com/Steg_project.pdf
[22] J.R. Krenn, *Steganography and Steganalysis January 2004*
http://www.krenn.nl/univ/cry/steg/article.pdf
[23] Michael T Raggo, *Identifying and Cracking Steganography Programs, Session 65*.
http://www.spy-hunter.com/contact_information