Volume 4, Issue 11, November 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

An Implementation of 4 Bit Image Steganography for Data Security in Clouds

Vaishali, Ankur Goyal
Yagyalakya Institute of Technology, Jaipur
Rajasthan Technical University, Kota, Rajasthan, India

Abstract– In this article the accomplishment and study of image steganographic procedure is carried out to insure security of the data on cloud. LSB encoding technique is utilized to hide the data in last 4 LSBs of 24 bit RGB pixels of colored cover image. The intermediate cover image containing the data (stegano image) shows a good correlation with the original cover image meaning a good amount of hiding of data. If the size of the data file to be sent is more than the capacity of the cover image then multiple copies of cover image are used to embed the data. The correlation of recovered data file with the original data file comes out to be 1 meaning a perfect recovery.

Keyword- RGB, LSB

I. INTRODUCTION

Cloud computing is a growing technology example that shifts the technological and computing concepts into utility-like solutions. This technology provides a large pool of services without any initial investment on set up. The cloud computing has a wast possibilities and exciting features but also comes with the threats of the data security as the data is available on the cloud so some secure mechanism should be introduced to make sure that the data which belongs to the specific client must be explored to the respective client only, data security and resource sharing responsibility between cloud provider and its customer for security and privacy and Providing secure multi-client environment (safe and sound and well-organized partitioning of virtualized and shared transportation among various clients). Steganographic techniques can be used to provide a perfect tool for data security and encryption, to permit network attacks or secret communication among secret parties. The aim of these techniques is to hide secret data (stegano grams) in the in some carrier e.g. in normal transmissions of clients. In ideal situation hidden data exchange cannot be detected by third parties. The best carrier for steganograms must possess two features: it should be popular i.e. usage such carrier should not be considered as an anomaly itself and modification of the carrier related to inserting the steganogram should not be "visible" to third party not aware of the steganographic procedure.

In recent times, the significance of ensuring the remote data integrity has been highlighted by the following research works [1–5]. These techniques, while can be useful to ensure the storage correctness without having users possessing data, cannot address all the security threats in cloud data storage, since they are all focusing on single server scenario and most of them do not consider dynamic data operations. As an complementary approach, researchers have also proposed distributed protocols [6–8] for ensuring storage correctness across multiple servers or peers. Again, none of these distributed schemes is aware of dynamic data operations. As a result, their applicability in cloud data storage can be drastically limited. Steganography is the art and science of writing hidden messages in such a way that no one apart from the intended recipient knows of the existence of the message [9]. Due to growing need for security of data image steganography is gaining popularity [10]. The main goal of steganography is to communicate securely in a completely undetectable manner [11] and to avoid drawing suspicion to the transmission of a hidden data [12]. This idea of data hiding is not a novelty, it has been used for centuries all across the world under different regimes but to date it is still unknown to most people – is a tool for hiding information so that it does not even appear to exist. However Steganography operates at a more complex level as detection is dependent on recognizing the underlying hidden data. The work focuses on characterization of information hiding possibilities in Cloud Computing via image steganographic process. The four least significant bits of some or all of the bytes inside an image is changed to a bit of the secret message. Digital images are mainly of two types (i) 24 bit images and (ii) 8 bit images. In 24 bit images we can embed three bytes of information in two pixels, four in LSB positions of the three eight bit values. Increasing or decreasing the value by changing the LSB does not change the appearance of the image; much so the resultant stego image looks almost same as the cover image. In 8 bit images, four bits of information can be hidden. If the size of the data file to be sent is more than the capacity of the cover image then multiple copies of cover image are used to embed the data. The correlation of recovered data file with the original data file comes out to be 1, meaning a perfect recovery.

II. LSB TECHNIQUE FOR HIDING THE DATA

In current work we embedded the data in 4-lsb bits of the RGB components of every pixel. Thus every pixel can accommodate 12 bits of data, we ready two consecutive pixels and embed 24 bits (3 bytes) into RGB components of these pixels. Let the image be of size n*m (where n is number of columns of pixels and m is number of rows of pixels),

thus size of file that can be embedded into the image is Floor ((n*m*3)/2). Let the data stream to be embedded is, "PM **may** visit Jaipur on Sunday". Assume that we have to embed 'may' underlined in the above sentence. Let the RGB component of the two pixels (say Pixel(i, j) and Pixel(i+1, j)) where it will embedded are as follows:

Red (Pixel(i, j)) → $(134)_{10}$ →$(10000110)_2$
Green (Pixel(i, j))→ $(14)_{10}$ →$(00001110)_2$
Blue (Pixel(i, j)) → $(108)_{10}$ →$(01101100)_2$
Red (Pixel(i+1, j)) → $(132)_{10}$ →$(10000100)_2$
Green (Pixel(i+1, j)) → $(18)_{10}$ →$(00010010)_2$
Blue (Pixel(i+1, j)) → $(110)_{10}$ →$(01101110)_2$

The ASCII codes for "may" which is to be embedded are

	Decimal	Binary	2's Complement
m	: $(109)_{10}$	→ $(01101101)_2$	→ $(10010011)_2$
a	: $(97)_{10}$	→ $(01100001)_2$	→ $(10011111)_2$
y	: $(121)_{10}$	→ $(01111001)_2$	→ $(10000111)_2$

The 4 MSB of 'm' (marked red) will be embedded into Red component of Pixel(i,j) while 4 LSB of 'm' will be embedded into Green component of Pixel(i,j). same for others, thus after the embedding process the contents of the RGB components of the Pixel(i, j) and Pixel(i+1, j) shall be as shown below:

Red (Pixel(i, j)) → $(131)_{10}$ →$(10001001)_2$
Green (Pixel(i, j)) → $(3)_{10}$ →$(00000011)_2$
Blue (Pixel(i, j)) → $(105)_{10}$ →$(01101001)_2$
Red (Pixel(i+1, j)) → $(143)_{10}$ →$(10001111)_2$
Green (Pixel(i+1, j)) →$(24)_{10}$ →$(00011000)_2$
Blue (Pixel(i+1, j)) → $(103)_{10}$ →$(01100111)_2$

The algorithm used can be summarized as like we consider a cover image Image_1 of size 120*140, i.e. width of image is 120 pixels and height is 140 pixels, and a data file data.txt of size 200 KB. Here total number of pixels in the image = 120 * 140 = 16800. In every pixel 12-bits of data can be embedded, last four bit at least significant bit of RGB component of pixel, so total data bits that can be embedded in the cover image = 16800 * 12 = 201600 bits or 25200 bytes. Total byte of the data file = 200 * 1024 = 204800. Here number of bytes in the data file is greater than byte that can be accommodated in the cover image. Thus we need multiple copies of the cover image to embed the data file, number of Ceiling (204800 / 25200) = 33. So 33 copies of cover image will be created rather denying the client as size of data file is larger than the number of bytes that can be accommodated in the cover image. Now the data file will also be broken down in 33 small chunks by the process, where first 32 chunks will of size 6300 bytes each and last chunk will be of 3200 bytes (204800 – 6300 * 32). Now iterative process will be called to embed every bit of data file into the 33 copies of the cover image to generate the stegno images which shall then be uploaded over the cloud. In such way the files having a large size can also be uploaded by parts practically having no limit over the data size to be uploaded. The Fig 2 below shows the process in a sequential manner

a) Encoding Process :

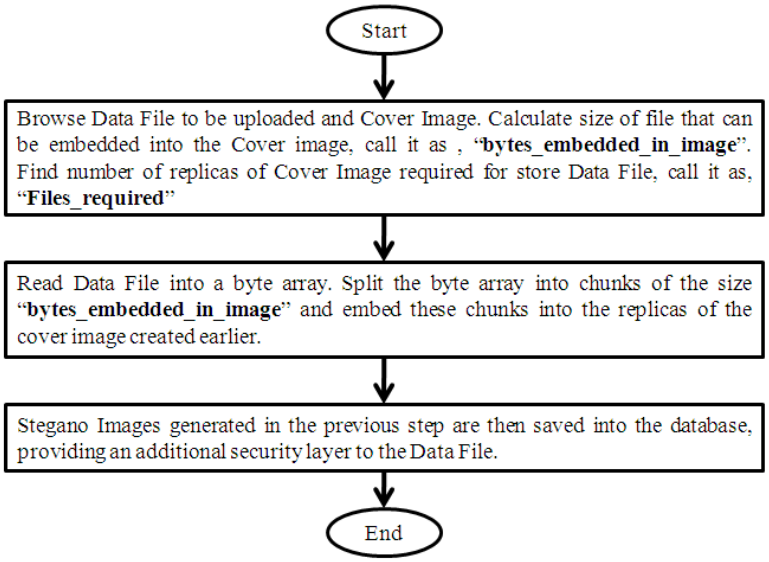

Fig.1 Flowchart of data up load on cloud

b) Decoding Process:

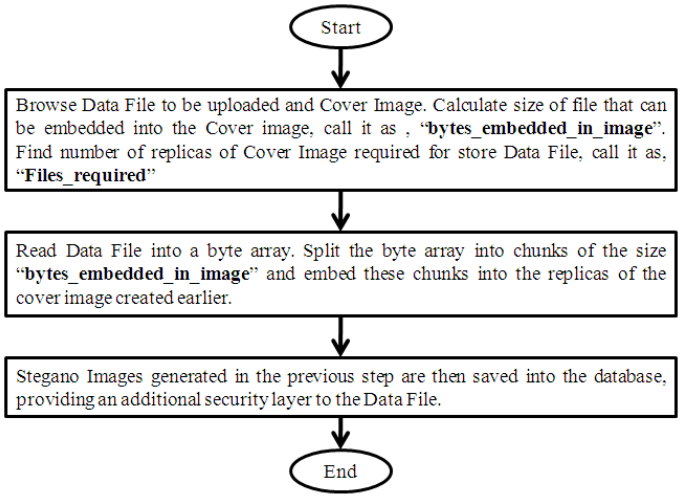

Fig.2 Flowchart of data down load on cloud

.

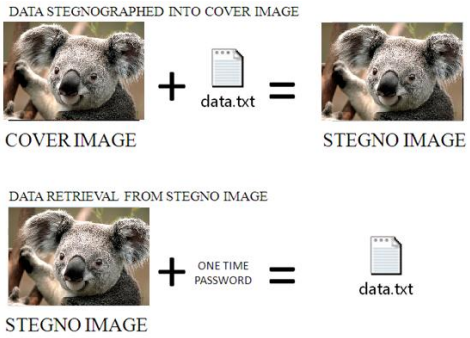

Fig.3 The process of image steganography

The complete process can be summarized as above. In this article an example is taken in which a text data file is hidden in an image. The text file is having a larger size then the capacity of the cover image so multiple copies of the cover image are created to hide the complete data.

III. ANALYSIS OF THE DESIGNED SYSTEM

To analyze the performance of the designed system for securing the data in the cover image and to insure the efficiency of the hiding procedure a correlation method is utilized and the results basis on the correlation method are summarized for a particular example. The cover image is distorted due to hiding of message are hiding the data by changing the pixel values of the image, this distortion or change in the cover image can be recognized and used for analyzing the system performance. In this section an example is taken in which a text data file is hidden in an image. The text file is having a larger size then the capacity of the cover image so multiple copies of the cover image are created to hide the complete data. The difference images and stegano images at each and every step are analyzed. The data matrices of each image are produced in order to justify the process results and the graphs are drawn for each image RGB components. The correlation between each stegano image is calculated.

The text file size used is having a size of 1920B. The cover image taken is having 3x12x16= 576 pixels. To hide one byte of data 2 pixels are required so the given image can handle 576 / 2 = 288 bytes of data. According to the capacity of the cover image seven replicas of cover image are needed to encrypt the complete date. Six Stegano images will be produced to hide the complete data. The Stegano image 1 produced is shown in Fig. 5.2 (a). It is the enlarged copy of the actual Stegano image produced. The difference image is also created by taking the difference of the cover image and the Stegano 1 image. The difference image is almost black showing no significant difference in the pixel values of the cover image. This makes the hiding of the data at a good level.

(a) (b)

Fig. 5.2 (a) Stegano Image 1 (b) Difference from cover image

Similarly six different Stegano images and respective difference images are obtained and the respective matrices are obtained for the comparison purpose rest of the Stegano images and respective matrices are added in appendix 1. Each stegano image seems to be an perfect copy of the cover image having almost no difference, which can be justified by all the difference images are almost black. Here R1, G1,B1 are matrices for Red, Green, Blue component of the Stegano-Image(1) respectively.

Here to estimate the similarity of the cover image with the stegano image the correlation of the RGB component is taken with respective components of each stegano image and the results of the correlation are summarized below: Correlation Coefficient shows amount of similarity between two matrices. Here the calculated value Correlation Coefficients of stegano-images with respect to Cover Image are:

Table 1 correlation coefficients of all image components

corr2(R, R1) = 0.9979
corr2(G, G1) = 0.9979
corr2(B, B1) = 0.9975
corr2(R, R2) = 0.9975
corr2(G, G2) = 0.9978
corr2(B, B2) = 0.9978
corr2(R, R3) = 0.9979
corr2(G, G3) = 0.9978
corr2(B, B3) = 0.9980
corr2(R, R4) = 0.9978
corr2(G, G4) = 0.9981
corr2(B, B4) = 0.9978
corr2(R, R5) = 0.9978
corr2(G, G5) = 0.9979
corr2(B, B5) = 0.9978
corr2(R, R6) = 0.9978
corr2(G, G6) = 0.9977
corr2(B, B6) = 0.9978
corr2(R, R7) = 0.9980
corr2(G, G7) = 0.9984
corr2(B, B7) = 0.9936

Here R, G, B are matrices for Red, Green, Blue component of the Cover Image respectively, R1, G1,B1 are matrices for Red, Green, Blue component of the Stegno-Image(1) respectively. To extend the possibility of mathematical verification of security of the method the entropy of the cover image and the different stegano images are calculated and the percent difference between the entropy of the cover image and stegano images are calculated

Table 2 entropy of various stegano images compared with the original image

		Percentage of variation in Entropy of Cover-Image and Stegno-images
Entropy of Cover-Image	7.2350	
Entropy of Stegno-Image(1)	6.4703	10.5695
Entropy of Stegno-Image(2)	6.4683	10.5971
Entropy of Stegno-Image(3)	6.5008	10.1479
Entropy of Stegno-Image(4)	6.4690	10.5874
Entropy of Stegno-Image(5)	6.5188	9.8991
Entropy of Stegno-Image(6)	6.4957	10.2184
Entropy of Stegno-Image(7)	6.7909	4.9549

From the above table of the entropies it very clear that the difference between entropies the cover image and the respective stegano images is very less also the Percentage of variation in Entropy of Cover-Image and Stegano –images is coming very less that is a very less of fraction showing a good amount matching of the images making the difference delectability very less.

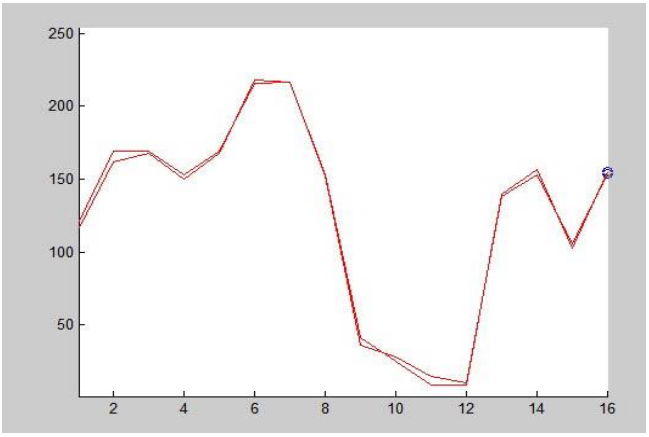

Fig.4 Plotting pixel values of R1 v/s R

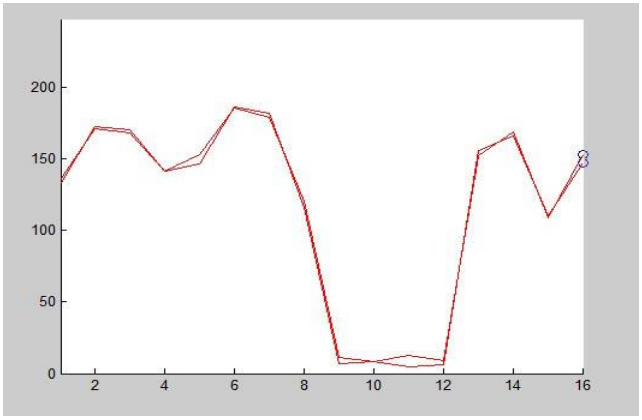

Fig. 5 Plotting pixel values of G1 v/s G

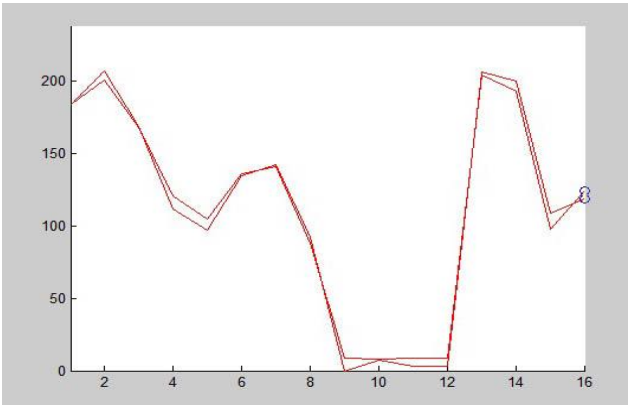

Fig. 6 Plotting of B1 v/s B

IV. CONCLUSIONS

An image steganographic process using the LSB encryption technique is used to secure the data over the clouds in such scheme the data to be transferred is hidden in the LSBs of pixels of the cover image so that he data or information to be sent is in the form of LSBs of the cover image which cannot be detected without any prior information of availability of data in that image. The conclusions of the work carried out are summarized below

- An automatic system is designed to upload to data on the clouds the data as soon as uploaded on the cloud are encrypted in the LSB,s of a cover image and the data is no longer available as the uploaded format but in the encrypted format now.
- The data containing capacity of the cover image is finite so if the data exceeds the capacity of the cover image a multiple copies of the cover image are created the data file is broken in the comestible size pieces and assembled back at the time of recovery.
- An stegano image is created with respect to each cover image embedded with data and the correlation of that stegano image comes out to be 1 in each case, proving that no one can detect the difference in stegano image and cover image.
- The entropy of each stegano image is having a very less difference with the entropy of the cover image again justifying the a good hiding possibility of data in images.

REFERENCES

[1] Juels and J. Burton S. Kaliski, "PORs: Proofs of Retrievability for Large Files," *Proc. of CCS '07*, pp. 584–597, 2007.

[2] H. Shacham and B. Waters, "Compact Proofs of Retrievability," *Proc. of Asiacrypt '08*, Dec. 2008.

[3] K. D. Bowers, A. Juels, and A. Oprea, "Proofs of Retrievability: Theory and Implementation," Cryptology ePrint Archive, Report 2008/175, 2008, http://eprint.iacr.org/.

[4] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores," *Proc. Of CCS '07*, pp. 598–609, 2007.

[5] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Data Possession," *Proc. of SecureComm '08*, pp. 1–10, 2008.

[6] T. S. J. Schwarz and E. L. Miller, "Store, Forget, and Check: Using Algebraic Signatures to Check Remotely Administered Storage," *Proc. of ICDCS '06*, pp. 12–12, 2006.

[7] M. Lillibridge, S. Elnikety, A. Birrell, M. Burrows, and M. Isard, "A Cooperative Internet Backup Scheme," *Proc. of the 2003 USENIX Annual Technical Conference (General Track)*, pp. 29–41, 2003.

[8] K. D. Bowers, A. Juels, and A. Oprea, "HAIL: A High-Availability and Integrity Layer for Cloud Storage," Cryptology ePrint Archive, Repor 2008/489, 2008, http://eprint.iacr.org/.

[9] R. Chandramouli and N. Memon,"Analysis of LSB based Image Steganography", IEEE ICIP, pp. 1022-1022, Oct. 2001.

[10] R.J. Anderson, F.A.P. Petitcolas, "On The Limits of Steganography", IEEE Journal ofSelected Area in Communicafions, pp. 474-481, May 1998.

[11] N.F. Johnson, S. Jajodia, "Staganalysis: The Investigation of Hiding Information", IEEE, pp. 113-116, 1998.

[12] H. Hastur, Mandelsteg, ftp://idea.sec.dsi.unimi.it/pub/security/crypt/code/

