Steganography With Data Integrity

Deepali
Department of Computer Science and Engineering
PEC University of Technology
, Chandigarh

Abstract

Steganography is the technique of hiding private or sensitive information within something that appears to be nothing out of the usual. Steganography is often confused with cryptology because the two are similar in the way that they both are used to protect important information. The difference between the two is that Steganography involves hiding information so it appears that no information is hidden at all. In this paper, we describe method of Steganography based on embedding encrypted message bits using RSA Algorithm in the 1st least significant (LSB Technique) and last 4 significant bits (Modulus 4 bit technique) of the pixel of image. Here we also provide integrity using MD5 hash algorithm. The analysis shows that the PSNR is improved in the case of LSB technique. Use of hash algorithm provides data integrity.

Keywords: Data integrity, LSB technique, MD5 Hash Algorithm, Modulus 4 bit algorithm, PSNR, RSA Algorithm, Steganography

1. Introduction:

Steganography is the technique of writing hidden messages in such a way that no one apart from the intended recipient knows of the existence of the hidden message. It is taken from Greek word “STEGANOS” which means “Covered” and “GRAPHIE “which mean “Writing”. So, Steganography is a method of covering important information behind an image. Steganography ancient origins can be traced back to 440 BC, from the Histories of Herodotus. Demeratus sent a warning about a forthcoming attack to Greece by writing it on a wooden panel and covering it in wax. During World War 2 invisible ink was used to write information on pieces of paper so that the paper appeared to the average person as just being blank pieces of paper. Liquids such as milk, vinegar and fruit juices were used, because when each one of these substances are heated they darken and become visible to the human eye It is not a rule that we must hide data in image files only; we can also hide data in MP3 and Video files too. When hiding information inside images the LSB (Least Significant Byte) method is usually used. When hiding information inside Audio files the technique usually used is low bit encoding which is somewhat similar to LSB that is generally used in Images. The problem with low bit encoding is that it is usually noticeable to the human ear, so it is a rather risky method for someone to use if they are trying to mask information inside of an audio file. Spread Spectrum is another method used to conceal information inside of an audio file. This method works by adding random noises to the signal, the information is conceal inside a carrier and spread across the frequency spectrum. When information is hidden inside video the program or person hiding the information will usually use the DCT (Discrete Cosine Transform) method. Steganography in Videos is similar to that of Steganography in Images, apart from information is hidden in each frame of video. When only a small amount of information is hidden inside of video it generally isn’t noticeable at all, however the more information that is hidden the more noticeable it becomes. So Steganography in Images is preferred.

2. Related Work:

2.1 Lsb Insertion Method

The least significant bit insertion method is probably the most well known image Stenography technique. It is a common, simple approach to embed information in a graphical image file. Unfortunately, it is extremely vulnerable to attacks, such as image manipulation. A simple conversion from a GIF or BMP format to a lossy compression format such as JPEG can destroy the hidden information in the image. When applying 4LSB techniques to each bytes of a 8-bit image, one bit can be encoded to each pixel. Any changes in the pixel bits will be indiscernible to the human eye. The main advantage of 4LSB insertion is that data can be hidden in the last four least significant bits of pixel and still the human eye would be unable to notice it. Care needs to be taken in the selection of the cover image, so that changes to the data will not be visible in the stego-image.

2.2 RSA Algorithm:

Encrypting using RSA, we encrypt our data that is hidden in an image. Hackers can not identify hidden data in images easily and at most they can get encrypted data from images which will not reveal any confidential information.. Care should be taken during the selection of prime numbers, so that hacker will not able to reveal key to decrypt.

2.3 MD5 Algorithm*:*

MD5 algorithm provides data integrity. Message digest is sent along encrypted data which is hidden in image. At receiver side, receiver first get data from image, decrypt it and then find message digest using same algorithm and compare it with original message digest. If they are same, data isn't tempered. Hence data integrity is maintained.

3. Implementation:

The proposed work provides data integrity using hash algorithm MD5. We create message digest that is sent along encrypted data. This digest is hidden in image. At receiver side, receiver first get data from image, decrypt it and then find message digest using same algorithm and match with original message digest. The challenge in this work was to find a way to camouflage a secret message in an image without perceptible degrading the image quality and to provide confidentiality and data integrity that make man-in-middle attack difficult. Therefore, we applied a encryption using RSA algorithm and MD5 hash algorithm.The main idea of this method is to utilize least significant bit or modulus 4 significant bits of a cover image to hide message bits. This approach is illustrated in details in the following four steps (algorithm):

• Step 1: Applying RSA encryption using sender's private key

Here sender and receiver generate their public and private keys using RSA algorithm and save in file. Then message bits are encrypted with sender's private key using RSA encryption $c = m^e$ (modulo n) . We do this encryption to provide authentication that data is sent by intended user because intended user know his private key.

• Step 2: Applying RSA encryption using receiver's public key

Here encrypted message is again encrypted with receiver's public key using RSA encryption ***$c = m^e$ (modulo n)***. We do this encryption to provide confidentiality that data is not read by any intruder without knowing private key.

• Step 3: Applying hash algorithm

Here original message is hashed with MD5 algorithm to create message digest that is sent along encrypted message to provide data integrity.

• Step 4: Embedding encrypted message bits and message digest

In this step, encrypted message bits are embedded one by one in image either at least significant bit or last four bits. In the first row of image we hide the size of our message that has to hide so, that we receiver can easily recover the message by knowing his private key. And from 2nd row data is hidden and a stego image is produced.

At receiver end data is extracted from stego image and decrypted by receiver's private key and then with sender's public key and after that message digest is created. If message digest matches with original message digest then data is not tempered and accepted.

4. Analysis

The obtained results of the experiments are summarized in the following Table 1.1 which shows PSNR of the different image categories (mountains, pokemon, dog, monkey, laptop, tree) that conceal same message. Table shows that for all type of images LSB technique has high PSNR value than modulus 4 bit technique.It means LSB causes less degradation in cover image than modulus 4 bit. From images shown below original and stego images cant be differentiated but magnified histogram of images shows that LSB is more closer to original image than modulus 4 bit and it ensures data integrity.

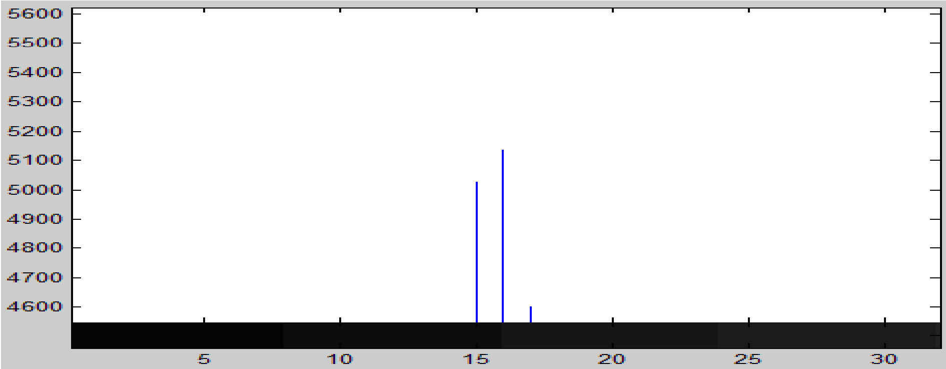
TABLE 1.1

IMAGE TYPE	LSB (PSNR Value)	MODULUS 4 BIT (PSNR Value)
Mountain	78.5481	64.0353
Pokemon	66.5931	52.0803
Dog	66.7186	52.0888
Monkey	66.7001	52.5743
Laptop	66.7514	50.3696
Tree	73.8528	58.8528

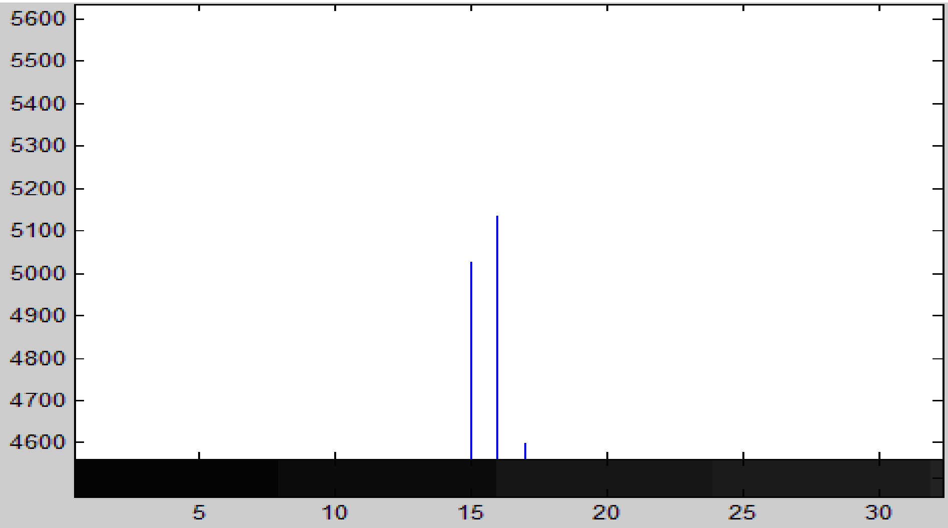
LSB Image

Modulus 4 Image

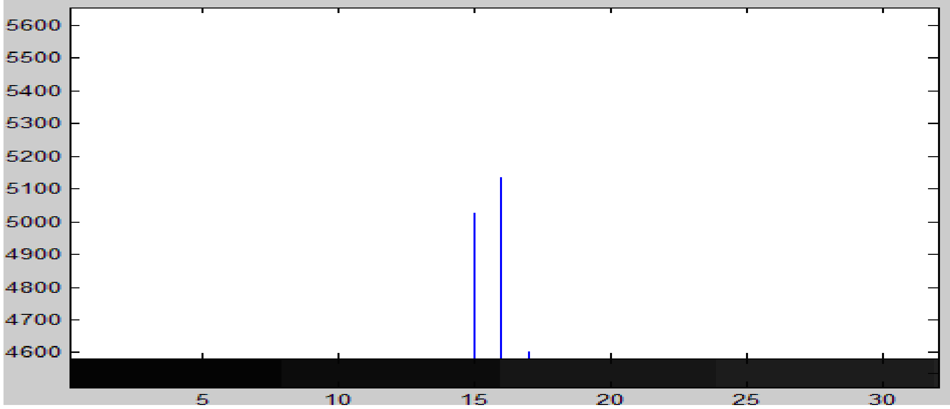
Original Image



Modulus 4 Histogram



LSB Histogram



Original Histogram

References

[1] Beenish Mehboob and Rashid Aziz Faruqui "A StegnographyImplementation" in 2008 IEEE

[2] Nedal M. S. Kafri1 and Hani Y. Suleiman Bit-4 of Frequency Domain-DCT Steganography Technique in 2009 IEEE

[3] Ismail Avcibas N.M. and B. Sankur, "Steganalysis using image quality metrics", In *IEEE Transactions on Image Processing*, vol. 12, No. 2., February 2003.

[4] M. **S.** Sutaone, M.V. Khandare "Image Based Steganography Using LSB Insertion Technique"

[5] Swati Tiwari1, R. P. Mahajan2 "A Secure Image Based Steganographic Model Using RSA Algorithm and LSB Insertion " in International Journal of Electronics Communication and Computer Engineering
Volume 3, Issue 1, ISSN 2249 –071X