Image and Text Hiding using RSA & Blowfish Algorithms with Hash-Lsb Technique

M.RAJKAMAL[1], B.S.E. ZORAIDA[2]

[1]Research Scholar,
School of Computer Science and Engineering, Bharathidasan University, Tamil Nadu, India

[2]Assistant Professor
School of Computer Science and Engineering, Bharathidasan University, Tamil Nadu, India

Abstract

Steganography is a method of hiding secret messages in a cover object while communication takes place between sender and receiver. Security of secret or important information has always been a major issue from the past times to the current time. It has always been the interested topic for researchers to develop secure techniques to send data without revealing it to anyone other than the receiver. Therefore from day to day researchers have developed many techniques to fulfill secure transfer of data and steganography is one of them. In this thesis we have developed a new technique of image steganography inside the embedding the encrypted Data file or message using Hash-LSB with RSA algorithm for providing more security to data as well as our data hiding method. The developed technique uses a hash function to generate a pattern for hiding data bits into LSB of RGB pixel values of the carry image. This technique makes sure that the data has been encrypted before embedding it into a carry image. Embedded-text in images usually carries important messages about the content. if the any cases third party get the message so many ways, so prevent this action this thesis implements hash table encryption to the message then hide into the image reason is to provide more secure way to transfer data. In this work, a new way of hiding information in an image with less variation in image bits have been created, which makes our technique secure and more efficient. In this technique also applied a cryptographic method. Second level is to encrypt and decrypt steganography image using blowfish algorithm, this action used to manage another cycle of security process implementation.

Keywords: *cryptography, steganography, hash-lsb merging, blowfish encryption & decryption, rsa encryption & decryption.*

1. Introduction

Cryptography or cryptology from "hidden secret"; and graphein, "writing" or “study", is the practice and study of techniques for secure communication in the presence of third parties is called adversaries. Due to the rapid growth of digital communication and electronic data exchange, information security has become a crucial issue in industry, business, and administration. Modern cryptography provides essential techniques for securing information and protecting data.

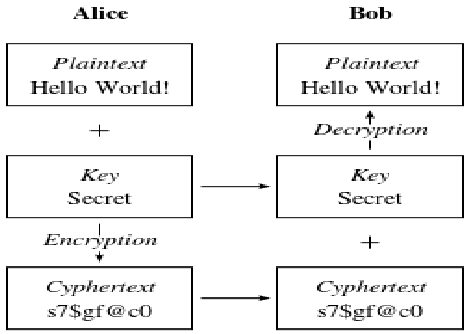


Fig. 1 Cryptography process flow

1.1 Steganography

Steganography is a technique used to transmit a secret message from a sender to a receiver in a way such that potential intruder does not suspect the existence of the message. Generally this can be done by embedding the secret message within another digital medium such as text, image, audio or video. The word steganography is of Greek origin and means “concealed writing” from the Greek words steganos meaning “covered or protected”, and graphie meaning “writing”. The first recorded use of the term was in 1499 by Johannes Trithemius in his Stegano-graphia, a treatise on cryptography and steganography disguised as a book on magic and classically, the hidden message may be in invisible ink between the visible lines of a private letter. It is a high security technique for long data transmission. There are various methods of steganography:

1 .Least significant bit (LSB) method
2. Transform domain techniques
3. Statistical methods
4. Distortion techniques

Image steganography

Hiding information inside images is a popular technique nowadays. An image with a secret message inside can easily be spread over the World Wide Web or in newsgroups. who created a scanning cluster which detects the presence of hidden messages inside images that were posted on the net. However, after checking one million images, no hidden messages were found, so the practical use of steganography still seems to be limited. To hide a message inside an image without changing its visible properties, the cover source can be altered in ”noisy” areas with many color variations, so less attention will be drawn to the modifications. The most common methods to make these alterations involve the usage of the least-significant bit or LSB, masking, filtering and transformations on the cover image. These techniques can be used with varying degrees of success on different types of image files.

A more complex way of hiding a secret inside an image comes with the use and modifications of discrete cosine transformations. Discrete cosine transformations (DST)), are used by the JPEG compression algorithm to transform successive 8 x 8 pixel blocks of the image, into 64 DCT coefficients each. Each DCT coefficient *F(u, v)* of an 8 x 8 block of image pixels *f(x, y)* is given by:

$$F(u,v)=\frac{1}{4}C(u)C(v)\left[\sum_{x=0}^{7}\sum_{y=0}^{7}f(x,y)*\cos\frac{(2x+1)u\pi}{16}\cos\frac{(2y+1)v\pi}{16}\right],\ C(x)=\begin{cases}\frac{1}{\sqrt{2}} & x=0\\ 1 & else.\end{cases}$$

After calculating the coefficients, the following quantizing operation is performed:

$$F^{Q}(u,v)=\left\lfloor \frac{F(u,v)}{Q(u,v)} \right\rfloor$$

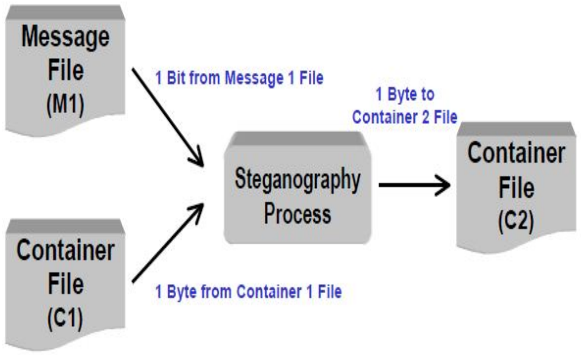

Message File (M1)
1 Bit from Message 1 File
1 Byte to Container 2 File
Steganography Process
Container File (C2)
Container File (C1)
1 Byte from Container 1 File

Fig. 2 image steganography process flow

2. Related work

There are many steganography techniques which are capable of hiding data within an image. These techniques can be classified into two categories based on their algorithms: 1.spatial domain based techniques;2. transform domain based techniques . The spatial domain based steganography technique use either the LSB or Bit Plane Complexity Segmentation (BPCS) algorithm The most widely used technique to hide data is the usage of the LSB . The existing techniques are mainly based on LSB (Least Significant Bit) where LSBs of the cover file are directly changed with message bits. A significant number of methods have been proposed for LSB steganography. Masud et al. has proposed a LSB technique for RGB true color image by enhancing the existing LSB substitution techniques to improve the security level of hidden information. In] designing of robust and secure image steganography based on LSB insertion and RSA encryption technique has been used. In proposed a LSB matching revisited image steganography and edge adaptive scheme which can select the embedding regions according to the size of secret message and the difference between two consecutive pixels in the cover image. Mohmmad A.Ahmed et al. proposed a method in which a message hidden inside an image by using the Least Significant Bit technique and after creation of the hidden message, the image will pass it in hash function to obtain hashing value using the MD5 technique. In two steganography technique proposed for hiding image in an image using LSB method for 24 bit color images. In a hash based approach proposed for secure keyless steganography in lossless RGB images that an improved steganography approach for hiding text messages in lossless RGB images.. In a security analysis on spatial domain steganography for JPEG decompressed images has been presented. Anderson and Petitcolas posed many of the open problems resolved in this article regarding to steganography. In particular, they pointed out that it was unclear how to prove the security of a steganography protocol. They also posed the open question of bounding the bandwidth that can be securely achieved over a given cover channel. Video steganography of late has also gained quite significance for researchers. Various techniques of LSB exist to implement video steganography . In a hash based least significant bit technique for video steganography has been proposed. Where the secret information is embedded in the LSB of the cover frames and a hash function is used to select the position of insertion in LSB bits. blowfish is a variable-length key block cipher. It is suitable for applications where the key does not change often, like a communications link or an automatic file encryptor. It is significantly faster than most encryption algorithms when implemented on 32-bit microprocessors with large data caches.

3. Existing techniques used

3.1. hash-lsb technique
3.2. Rsa algorithm
3.3 Blowfish algorithm
3.4 DCT-Discrete Cosine Transform

3.1. H-LSB based technique

The Hash Based Least Significant Bit Technique For image Steganography deals with hiding secret message or information within a image. Steganography is nothing but the covered writing it includes process that conceals information within other data and also conceals the fact that a secret message is being sent. Steganography is the art of secret communication or the science of invisible communication. Hash based least significant bit technique for image steganography has been proposed whose main goal is to embed a secret information or file in a particular image file and then extract it using a stego key or password. In this Least Significant Bit insertion method is used for steganography so as to embed data in cover image with change in the lower bit. This LSB insertion is not visible.

Image Steganography Technique has been proposed in which it perform encoding and decoding for hiding message and extracting message respectively. First of all message file will be embedded within the cover image file by using the steganography like a LSB Techniques. This steganography file is again applied to steganography tool to extract embedded data. A cover image consists of collection of pixels and the secret data is embedded in these frames as payload. Data hiding in image by encoding and retrieving data by decoding is explained below.

Encoding Process

For encoding first a image file is selected then information about the cover free pixel (LSB) will be collected. These pixels of image are separated from each other then in this image a secret message is hidden using hash based least significant bit technique. As hash code is generated then it helps to embed data within the frame. Then it will find 4 LSB position in the pixel in which the secret message will be embedded. Stego pixel combine with other pixel and then Stegno image is formed. This Stegno image will be transmitted to the intended receiver. This encoding process used to hide data.

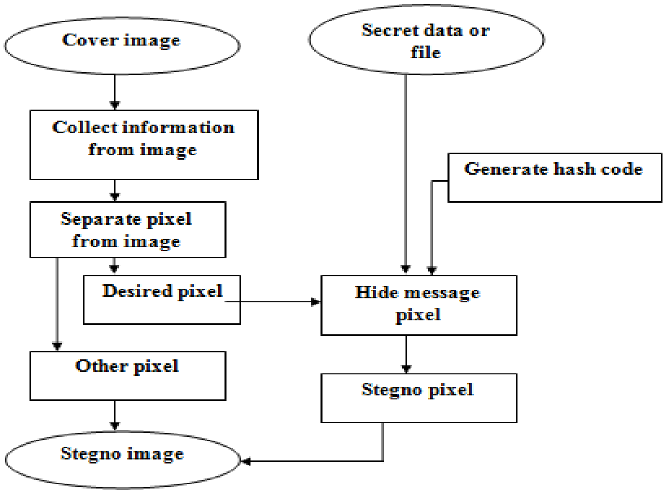


Fig. 3 Image Encoding Techniques

Decoding Process

For decoding a Stegno image is taken and to extract the secret data or information from image all information about image is collected. This Stego image pixel will be applied to the de steganography tool to decode data. From this pixel hidden information is taken out. The password will be used to decode the data as it is known to intended receiver. Here password also known as Stego key.

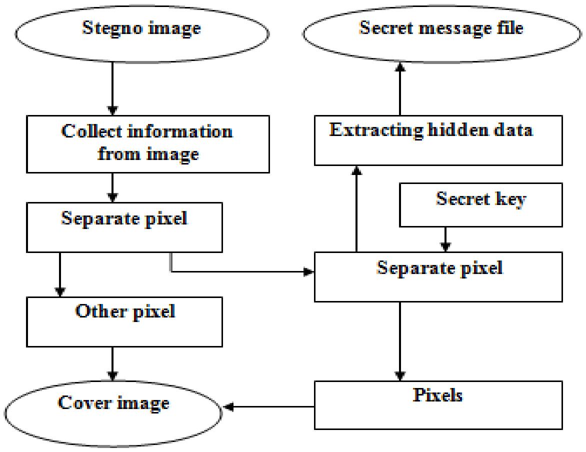


Fig.4 Image Decoding Techniques

Hash Algorithm Flow

Hash based least significant bit technique which produces hash function. This hash function deals with the LSB bit position within the pixel and the position of each hidden image pixel and also with the number of bits of LSB. Hash value takes a variable size of input and returns a fixed size of digital string as output. Hash function also used for detecting duplicated record in large files.

Hash function generally given by
$x= y \% z$

Where, x is LSB bit position within the pixel, y represents the position of each hidden image pixel and z is number of bits of LSB.

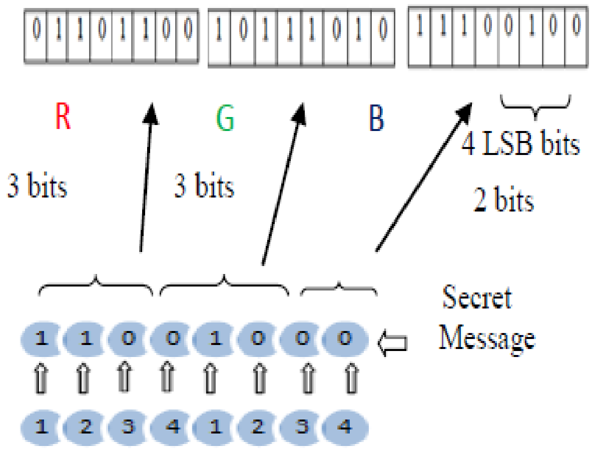


Fig.5 Distribution of Secret Message bits

3.2 RSA algorithm

You must understand the following mathematical principles to understand this algorithm and if you don't understand these principles, look them up first (I had to look up the last one, the Euler totient function, as I had never heard of it):

- Exponentials
- Prime numbers
- Prime factorization
- Greatest Common Denominator (GCD)
- Modular arithmetic
- Euler totient function

This is also going to have development in mind, so you maybe should also understand: binary, char, bits, ascii, UTF-8, etc..

It can be quite annoying for me when it shows algorithms using one character variables. This may be the mathematical way but I prefer to use a developer style where variables are named clearly. I need to make sure I understand how RSA works so I am going to write about it.

Here is an example of how they use just one character:

The RSA algorithm uses two keys, d and e, which work in pairs, for decryption and encryption, respectively. A plaintext message P is encrypted to cipher text C by

$C = P^{e} \bmod n$

The plaintext is recovered by

$P = C^{d} \bmod n$

Because of symmetry in modular arithmetic, encryption and decryption are mutual inverses and commutative. Therefore,

$P = C^{d} \bmod n = (P^{e})^{d} \bmod n = (P^{d})^{e} \bmod n$

This relationship means that one can apply the encrypting transformation and then the decrypting one, or the one followed by the encrypting one.[1]

I would never write code this way and looking at this, it might leave one who is not an expert wondering what do the variables P, C, d, e, n represent again? And is there a reason P, C is capitalized and d, e, n are lower case? Lets rewrite these with nice developer variable names where the name comments itself based on the what it really is. In the quoted text above each variable is defined clearly except what "mod n" really represents, I had to read on to determine this. Also, where to get the values for each variable is not defined, again, I had to read on to determine this, and this led to more equations to add to the list. These are the equations, in order

Equation List

1. ProductOfPrime1Prime2 = Prime1 * Prime2
2. Totient = (Prime1 – 1) * (Prime2 -1)
3. (Totient * Any Integer) + 1 = 1 mod Totient
4. Encrypt Prime * Decrypt Prime = 1 mod Totient
5. Encrypt Prime * Decrypt Prime = (Totient * Any Integer) + 1 where (Totient * Any Integer) + 1 has exactly prime factors
6. Cipher Text = PlainText$^{\text{EncryptPrime}}$ mod ProductOfPrime1Prime2
7. Plain Text = Ciphertext$^{\text{DecryptPrime}}$ mod ProductOfPrime1Prime2
8. Plain Text = Ciphertext$^{\text{DecryptPrime}}$ mod ProductOfPrime1Prime2 = (PlainText$^{\text{EncryptPrime}}$)$^{\text{Decrypt Prime}}$ mod ProductOfPrime1Prime2 = (PlainText$^{\text{DecryptPrime}}$)$^{\text{Encrypt Prime}}$ mod n

Some of the values above you get to "choose" or if you were writing this algorithm in code, you would probably not "choose" so much as generate the value at random. So if we get to choose, then lets learn how to choose.

3.3. Blowfish Algorithm

Blowfish algorithm uses a Feistel network for data encryption which iterates the function 16 times. Each round includes a key dependent permutation and data dependent substitution.

Process of data encryption and the different steps in encryption are described below:

1.Split the 64 bit block into two equal blocks having 32 bits size each (XL and XR).The left block XL is XOR'd with first element of P-block, and thus obtained result is fed to the F function. 1 P

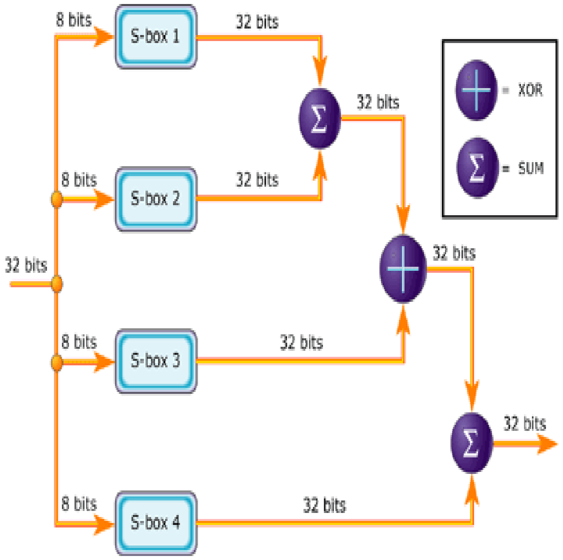


Fig.6 blowfish flow

2. In the F function block, substitution operation is carried out where the given 32 bit input is transformed into another 32 bit output.

3. The output from F block is XOR'd with right half XR and the results obtained are swapped in the Fig.after completing each round successfully, the so formed right half become the new left half or vice versa.

These steps are continued up to 16 rounds. .

4.The final left and right halves are not swapped but XOR'd with seventeenth and eighteenth P box elements.So obtained result is the cipher text which is non understandable to outsiders and attackers.

3.4 DCT-Discrete Cosine Transform

DCT is like a encoder and decoder. The first stage of image compression is DCT. Encoder. It consists of FDCT, quantize, and entropy encoder. The second stage is DCT decoder. It consists of entropy decoder, debutanizer and inverse mapped.

1. The input image is N by M;
2. F (i, j) is the intensity of the pixel in row i and column j;
3. F (u, v) is the DCT coefficient in row k1 and column K2 of the DCT matrix.
4. For most images, much of the signal energy lies at low frequencies; these appear in the upper left corner of the DCT.

$$C(u,v) = \frac{\alpha(v)}{2} \cdot \frac{\alpha(u)}{2} \sum_{y=0}^{7} \sum_{x=0}^{7} f(y,x) \cdot Cos\left[\frac{\Pi(2x+1)u}{16}\right] \cdot Cos\left[\frac{\Pi(2y+1)v}{16}\right]$$

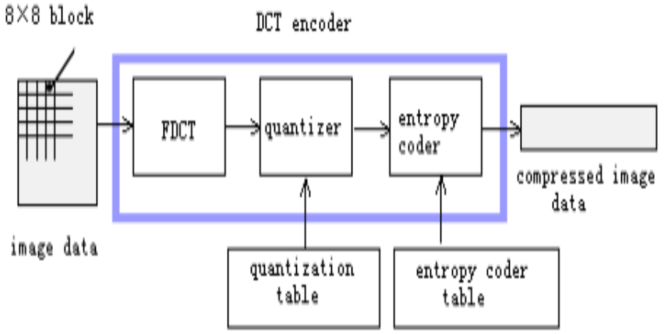
$$\text{For } u, v = 0,1,2,\ldots 7$$

$$f(x,y) = \sum_{y=0}^{7} \frac{\alpha(v)}{2} \sum_{x=0}^{7} \frac{\alpha(u)}{2} \cdot C(v,u) \cdot Cos\left[\frac{\Pi(2x+1)u}{16}\right] Cos\left[\frac{\Pi(2y+1)v}{16}\right]$$

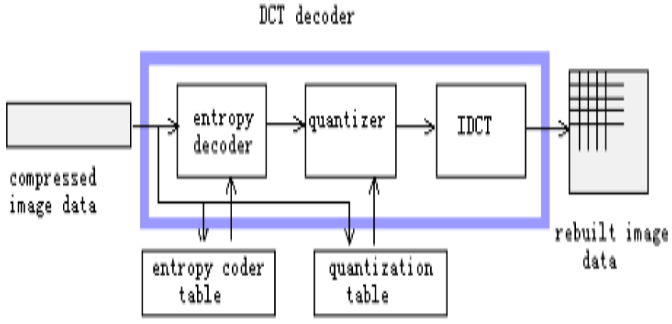
$$\text{for } x, y = 0,1,2,\ldots 7$$

$$\alpha = \begin{bmatrix} 1/\sqrt{2} & \text{if } u = 0 \\ 1 & \text{if } u \diamond 0 \end{bmatrix}$$

5. Compression is achieved since the lower right values represent higher frequencies, and are often small - small enough to be neglected with little visible distortion.
6. The DCT input is an 8 by 8 array of integers. This array contains each pixel's gray scale level;
7. 8 bit pixels have levels from 0 to 255.



(a) Compressed steps



(b) Decompressed steps

Fig.2 The basic operation of the DCT is as follows

4. Proposed work

In this proposed work we have implement an image steganography using an H-LSB encoding and decoding, RSA message or file encryption and decryption, and Blowfish used an image encryption decryption algorithm.

Image, text and document type of data's are hide to inside the image. The over all process are create a dual encryption. I.e. our secret data's encryption on rsa and blowfish algorithms

First encryption

Rsa algorithm only encrypts the secret data, then merges to cover image using hash-lsb techniques. Now create a stegno image (viewable stegno image)this image view our human eyes.

Second encryption

Blowfish algorithm encrypts the overall stegno images. in this level can't view the stegno image(not viewable stegno image).This stegno image sends to receiver. Receiver access the reverse process gets the secret data's (image, text or document) .this overall combine works create high security data transaction.
Generally, now a day's hackers easily find out what type of data's hide in cover image. But this new implementation process create highly and confident information protection. Because, incase third person hack our stegno data that time only view on encryption format data's. Receiver also view the original data must on two times decrypt the stegno data based on the keys (rsa & blowfish.)

Sender side process

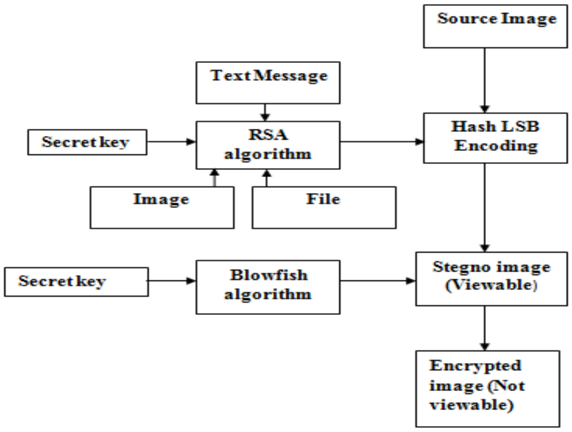


Fig.7 Sender side process view using RSA, H-LSB, Blowfish

Step1: The secret message or file is encrypted first using RSA algorithm with secret key (private key) value.

Step2: Encrypted message or file (cipher text) is embedding in to the cover image using H-LSB. The Stegno image created successfully.

Step3: Stegno image encrypted using Blowfish algorithm with secret key (private key), then got an encrypted cipher format images.

Step4: The cipher format image transferred to the recipient.

Receiver side process

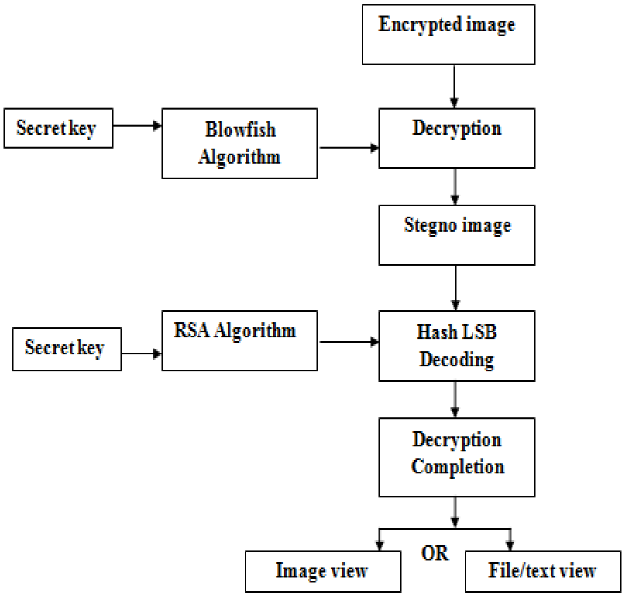


Fig.8 Receiver side process view using, Blowfish, H-LSB, RSA
:

Step1: The recipient receives the encrypted message and decrypts the image using blowfish secret key (Private Key) value. Then got an viewable Stegno image

Step2: Then decode the Stegno image using RSA secret key value (Private Key).then got the real image file and secret message or file also.

Three level processes use this work in secure data transmission

1. Secrete message or file Encryption decryption using RSA algorithm.
2. Encrypted and decrypted message or file embedding and retrieving cover image using H-LSB Technique.
3. Encrypt and Decrypt steganography image using Blowfish Algorithm.

In image technology, secret communication is achieved to embed a message or file into cover image (used as the carrier

to embed message into) and generate a Stegno-image (generated image which is carrying a hidden message). In this paper we have critically analyzed various steganography techniques and also have covered steganography.

Generally image steganography is categorized in following aspects shows the best Steganography measures.

High Capacity: Maximum size of information can be embedded into image.

Perceptual Transparency: After hiding process into cover image, perceptual quality will be degraded into Stegno-image as compare to cover-image.

Robustness: After embedding, data should stay intact if stegno-image goes into some transformation such as cropping, scaling, filtering and addition of noise.

Temper Resistance: It should be difficult to alter the message once it has been embedded into Stego-image.

Computation Complexity: How much expensive it is computationally for embedding and extracting a hidden message?

5. Conclusion and future scope

A Secure Image Steganography Based on RSA Algorithm and Hash-LSB Technique has been implemented. An efficient encryption technique RSA used to encode the secret information or data file, and then hide into the cover image file using DCT (Discrete Cosine Transform) and Hash-LSB technique secret hidden process done without disturb image viewing option. After encrypt the covered image using blowfish algorithm using secret key value reason is provide user authentication. The dual encryption process done using RSA and Blowfish algorithm. Then the dual encryption process completion processed cover file transfer recipient. The reverse process will be done retrieving the original data according to secret key value for both RSA and blowfish algorithm.

In future work develop a same process after completion the image format will be changing like (jpg, png, gif, bmp).this thesis work will be continuing in future using video file also like (3gp,mp4) and cryptography algorithm also possible to implement higher level(ex AES,MSB). method for embedding secret messages into cover images without producing any major changes has been accomplished through Hash-LSB method. In this work, a new way of hiding information in an image with less variation in image bits have been created, which makes our technique secure and more efficient. This technique also applies a cryptographic method i.e.RSA algorithm to secure the secret message so that it is not easy to break the encryption without the key. RSA algorithm itself is very secure that's why we used in this technique to increase the security of the secret message. A specified embedding technique uses hash function and also provide encryption of data uses RSA algorithm; makes our technique a very much usable and trustworthy to send information over any unsecure channel or internet. The H-LSB technique have been applied to .tiff images; however it can work with any other formats with minor procedural modification like for compressed images. Performance analysis of the developed technique have been evaluated by comparing it with simple LSB technique, which have resulted a very good MSE and PSNR values for the stego images. The future scope for the proposed method might be the development of an enhanced steganography that can have the authentication module along with encryption and decryption. Meanwhile the work can be enhanced for other data files like video, audio, text. Similarly the steganography technique can be developed for 3D images. The further work may contain combination of this method to message digesting algorithms.

Reference

[1] Anil Kumar, Rohini Sharma, "*A Secure Image Steganography Based on RSA Algorithm and Hash-LSB Technique*", International Journal of Advanced Research in Computer Science and Software Engineering,
Vol. 3, Issue No. 7, July 2013

[2] Aishwary Kulshreshta , Ankur Goyal"*Image Steganography Using Dynamic LSB with Blowfish Algorithm*" International Journal of Computer & Organization Trends, Vol 3 Issue No 7, August 2013.

[3] Mamta Juneja, Parvinder Singh Sandhu, "*Designing of Robust Image Steganography Technique Based on LSB Insertion and Encryption*", International Conference on Advances in Recent Technologies in Communication and Computing, Pages No. 302 – 305, 27-28 Oct., 2009.

[4] Swati Tiwari, R. P. Mahajan, "*A Secure Image Based Steganographic Model Using RSA Algorithm and LSB Insertion*", International Journal of Electronics Communication and Computer Engineering (IJECCE), Vol. 3, Issue No. 1, 2012.

[5] N. F. Johnson, S. Jajodia, "*Steganography: seeing the unseen*", IEEE Computer, Vol. 31, Issue No. 2, Pages No. 26 - 34, Feb., 1998.

[6] Wien Hong, Tung-Shou Chen, "*A Novel Data Embedding Method Using Adaptive Pixel Pair Matching*", IEEE Transactions on Information Forensics and Security, Vol. 7, Issue No. 1, Pages No. 176 - 184, Feb., 2012.

[7] Komal Patel, Sumit Utareja, Hitesh Gupta"Information Hiding using Least Significant Bit Steganography and Blowfish Algorithm" International Journal of Computer Applications, Vol. 63, Issue No.13, February 2013.

[8] R. Chandramouli, N. Memon, "*Analysis of LSB based image Steganography techniques*", International Conference on Image Processing, Vol. 3, Pages No. 1019 – 1022, 07 Oct 2001-10 Oct, 2001.

[9] Weiqi Luo, Fangjun Huang, Jiwu Huang, "*Edge Adaptive Image Steganography Based on LSB Matching Revisited*", IEEE Transactions on Information Forensics and Security, Vol. 5, Issue No. 2, Pages No. 201 – 214, June, 2010.

[10] Ross J. Anderson, Fabien A. P. Petitcolas, "*On the Limits of Steganography*", IEEE Journal on Selected Areas in Communications, Vol. 16, Issue No. 4, Pages No. 474 – 481, May, 1998.

[11] Min-Wen Chao, Chao-hung Lin, Cheng-Wei Yu, Tong-Yee Lee, "*A High Capacity 3D Steganography Algorithm*", IEEE Transactions on Visualization and Computer Graphics,Vol. 15, Issue No. 2, Pages No. 274 – 284, March-April, 2009.

[12] Nicholas Hopper, Luis von Ahn, John Langford, "*Provably Secure Steganography*", IEEE Transactions on Computers, Vol. 58, Issue No. 5, Pages No. 662 – 676, May, 2009.