BEST: International Journal of Management, Information Technology and Engineering (BEST: IJMITE)
ISSN 2348-0513
Vol. 3, Issue 4, Apr 2015, 5-12

TEXT HIDING USING RSA AND BLOWFISH ALGORITHMS WITH HASH-BASED LSB TECNIQUE

D.M.S. MADHURI[1], G. ANNAPURNA[2], CH. VENKATARAMANA[3] & G. SWETHA[4]

[1]Assistant Professor, Department of CSE, LIET, Andhra Pradesh, India

[2,3,4]Department of CSE, LIET, Andhra Pradesh, India

ABSTRACT

Every confidential information like bank details, university details, shopping mall details etc., needs security today. There are many techniques like cryptography and steganography to provide security to the important information by hiding the data in some way. The process of hiding may include converting the original data into cipher text or may be hiding the text or image or video by another text or image or video. In this paper we mainly provided the security for information in the form of text by using both cryptography and steganography combinely. Our paper describes the process of converting the text into cipher text using a cryptographic technique called RSA algorithm. This cipher text will be encoded by calculating the pixel positions to embed the cipher text into an image using an insertion technique called Hash-LSB encoding. This encoded image will now be encrypted into non-viewable image using BLOWFISH algorithm. The retrieval of the plain text will also be shown by using decryption algorithms of RSA and BLOWFISH.

KEYWORDS: Cryptography, Steganography, RSA algorithm, HLSB, Blowfish Algorithm

INTRODUCTION

Cryptography is a process of proving security for the important information by converting the text into cipher text using algorithms like RSA, Deffie-hellman, ceaser cipher etc.,

Every algorithm includes a key. The key may be a public key or a private key or a secret key. Public key is a key which is known to everyone, private key is a key which is known to only sender and receiver. The key is used to break the cipher and to retrieve the original text.

Steganography is the process of hiding information with another information. The information may be a text file, audio file or a video file. Steganography literally means “covered writing”.

In our paper we explained about image steganography that is, encrypting the image into non-viewable image using Blowfish encryption algorithm. RSA algorithm deals with the conversion of plain text into cipher text using a key. RSA algorithm is designed by three crypt analysts called Rivest Shamir and Adleman. HLSB is a technique to know the pixel positions in an image into which the text is to be hidden. This is possible by generating a hash function for the each pixel in the image. Blowfish algorithm is a method of encrypting an image into cipher formatted image using a key.

RELATED WORK

Steganography is a process of hiding the data using another data. Our paper works on embedding the text which has already encrypted will be embedded into an image using HLSB encoding technique. In this method the least significant bits of red green and blue components of an image are replaced by the message bits so that the picture quality never decreases. This is sure that by changing the least significant bits of the RGB components never effects the picture clarity

because the lsb bits doesn't show any difference to the human eye. The related work we have done is to insert the message bits in four least significant bits of the RGB components of every pixel of the image. The remaining paper explains about the existing techniques which are,

- RSA algorithm
- HLSB technique
- Blowfish algorithm

RSA ALGORITHM

RSA algorithm includes a key that is private key for both encryption and decryption algorithms. This key will be only known to sender and receiver. This algorithm is based on euler's totient function.RSA algorithm uses two keys called 'd' and 'e' for decryption and encryption respectively. The plain text is taken as 'P' which will be converted to cipher text 'C'.

The cipher text 'C' will be generated by using the formula,

C=P^ e mod n

The plain text is retrieved from the cipher text using the formula,

P=C^ d mod n

Where n is the value calculated by the product of two primes say 'p' and 'q'.

By the rules of symmetry encryption and decryption are commutative to each other,

P=C^ d mod n = (P^ e)^ d mod n =(P ^ d)^e mod n

This relationship explains that one can do encryption and then decryption or viceversa by using the formulae provided.

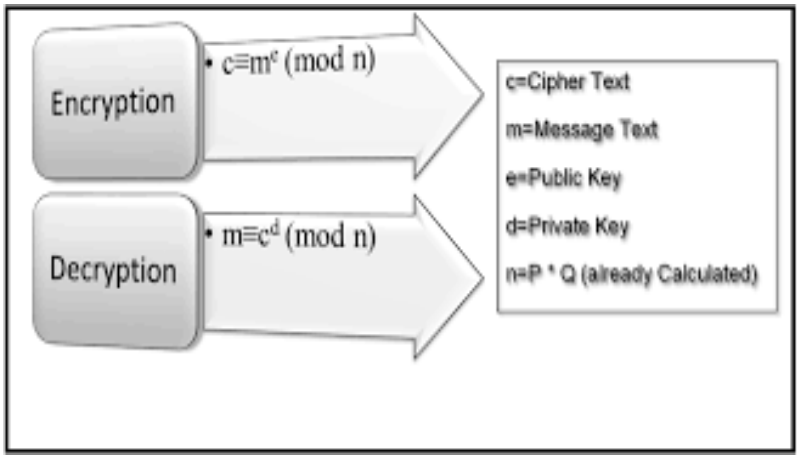

Figure 1: RSA Encryption and Decryption

HASH-BASED LSB TECHNIQUE

As steganography is said as "covered writing" Hash based least significant bit deals with hiding the message bits in an image by knowing the pixel positions using hash function and embedding the message into the calculated least significant bits of known pixel positions with the process of encoding. The file is inserted in to am image in its lower bits

because, the LSB insertion is not visible. The cover image consists of pixels into which the message is embedded into frames as payload.

Encoding Process

First select an image and collect the information about the cover free pixels. After collecting the pixel information divide the remaining pixels from the cover free pixels, and embed the message bits into that pixels at the lower bit values at four LSBs by generating a hash function, this results in stego pixels. Later these stego pixels will be combined with remaining pixels to form a stegno image. This is the process of encoding the message by an image.

Decoding Process

To get back the hidden text the information about the stegno image is collected and sent it through desteganography tool to decode the image. Now the hidden date is retrieved. A password called stego-key may be used to decode the image which is known to intended receiver.

Hash Algorithm Flow

This technique produces the hash function that deals with the least significant bits position within the pixels**.** Hash value takes a variable size of input and returns a fixed size of digital string as output.

Hash function is calculated by the formula,

X=Y%Z

Where X is lsb bit position within the pixel, Y is position of each hidden image pixel, Z is number of lsb bits.

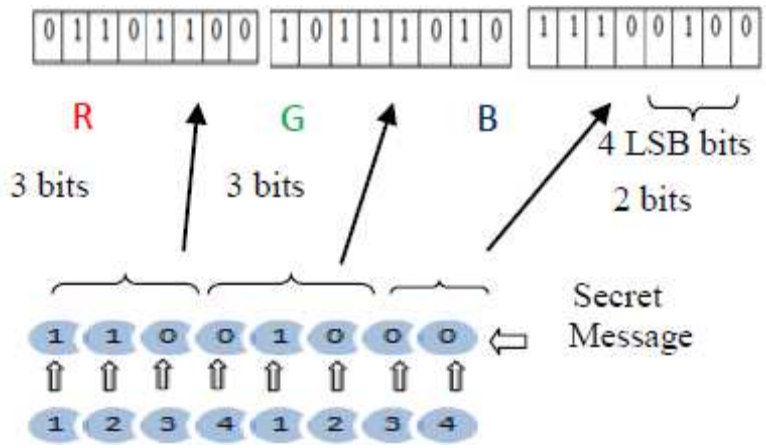

Figure 2: Distribution of Message into Pixels

BLOWFISH ALGORITHM

Blowfish algorithm is a feistel network in which the encryption function iterates for 16 times. In this paper blowfish algorithm is used for encrypting an image into cipher formatted image. To achieve this DCT (discrete cosine transform is used).

DCT (Discrete Cosine Transform)

This is like an encoder and decoder**.** the image will be compressed using DCT encoder and decompressed by DCT decoder.

- The image that is taken as input is N by M
- In f(i,j) –'i' represents the intensity if pixel ith row and jth column
- f(u,v) –DCT coefficient in k1 row and k2 column in the N by M matrix
- The signal energy of low frequency is appeared at the upper left corner of the DCT
- Compression is possible as the lower right values deals with higher frequencies.
- The input of DCT is 8 by 8 array of integers. The array contains gray scale value of each pixel.
- 8 bit pixels have 0to255 levels.

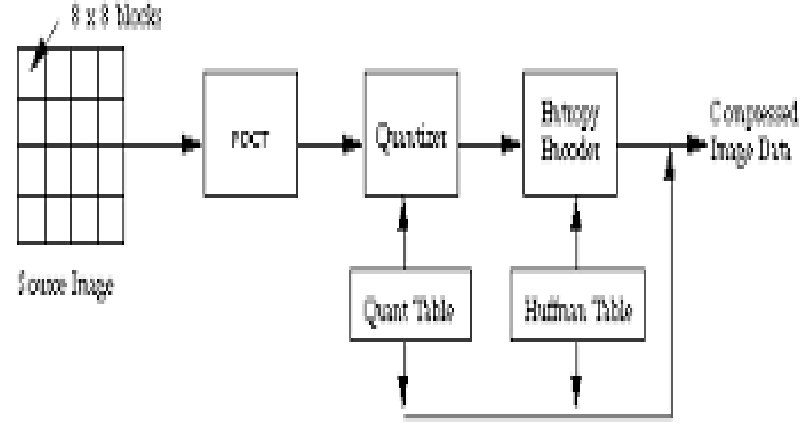

Figure 3: Compression Process

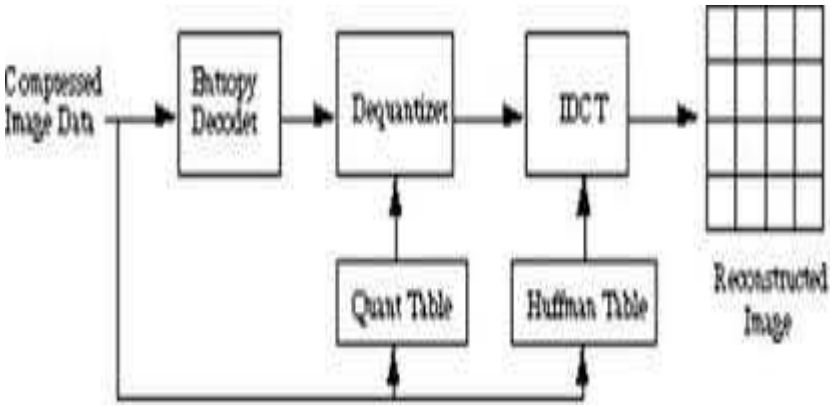

Figure 4: Decompression Process

PROPOSED WORK

In this paper we proposed the work that includes message hiding in an image using Hash-based lsb encoding and decoding technique along with RSA encryption algorithm to convert the original message to cipher text and finally Blowfish algorithm to convert the stegno -image to non viewable image using DCT encoder and decoder.

The overall process includes dual encryption as we used two keys one for RSA and other for Blowfish algorithm.

First encryption includes the conversion of plain text into cipher text using a private key for RSA encryption algorithm and then a stegno image is created by embedding this cipher text into an image by using HLSB encoding technique. These stegno images are viewable to human eyes.

Second encryption includes encrypting the all stegno images using blowfish secret key which is only known to sender and receiver. These encrypted images will not be viewable to the human eyes. The usage of two secret keys for encryption is to provide more security for the confidential and important information. When a third party wants to know the information it will be difficult to break the cipher as he has to know the complete information about the two secret keys. This paper provides better security when compared to other papers which have used only one encryption algorithm.

Algorithm for Encryption at Sender Side

Step 1: Sender types the message

Step 2: The message file is encrypted into cipher text using RSA secret key for encryption algorithm.

Step 3: The encrypted file along with an input image are taken and the cipher text will be embedded into the image pixels using HLSB(4) encoding technique with a stego key as password known to sender and receiver.

Step 4: Now the encoded image is taken as input and encrypted into non-viewable image using a secret key for Blowfish encryption algorithm.

Step 5: The resultant image is the completely encrypted image with data hidden inside it.

Algorithm for Decryption At Receiver Side

Step 1: The non-viewable image is taken as input at receiver side and decrypted into stegno image by using the secret key for Blowfish decryption algorithm.

Step 2: This stegno image along with the stego key will be used for decoding the image to separate the cipher text from the image using HLSB(4) decoding technique.

Step 3: The cipher text is now decrypted into plain text using secret key for RSA decryption algorithm.

Step 4: The receiver will now get the original text that is given by the sender using two secret keys for decryption.

WORKING MODEL

The input may be a text, image or a file that to be given to RSA algorithm using a secret key that is only known to sender and receiver. The output will be a cipher text, image or a file which will be encoded by taking an image as input with HLSB encoding technique which gives stegno image as output. By using Blowfish encryption algorithm with a secret key will converts the stegno image to cipher formatted image (not viewable).

The stegno image will be decoded by collecting the information from image using HLSB decoding technique. At the other side the cipher formatted image will be converted to normal viewable image using secret key of Blowfish decryption algorithm. This image will be decoded and cipher text is retrieved successfully and then converted to plain text using RSA public key decryption algorithm.

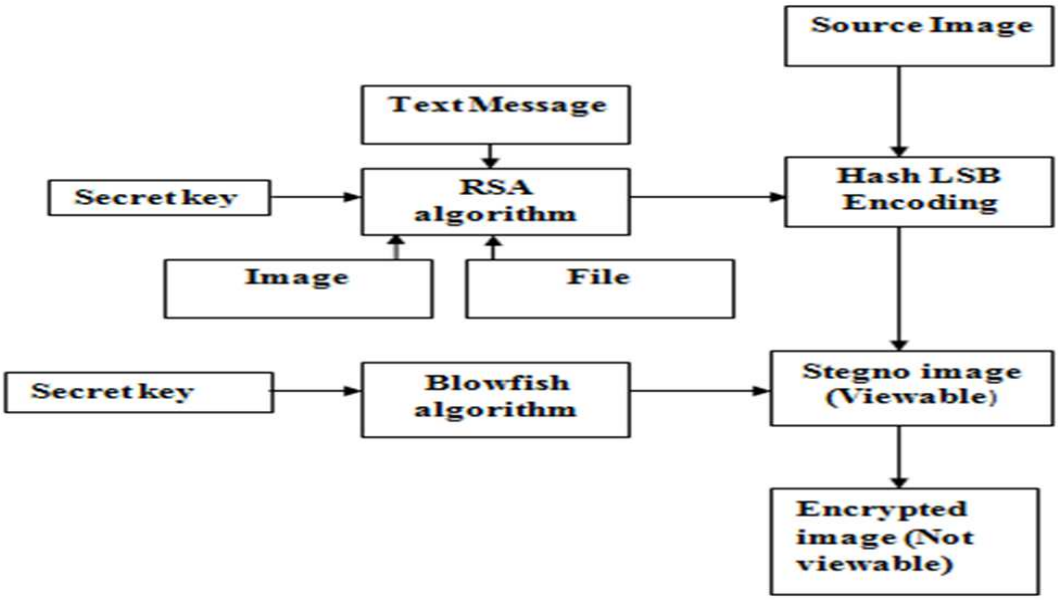

Figure 5: System Architecture

RESULTS

The above working model has been implemented and result screens are shown below. Figure 6 shows the encryption of data and figure 7 shows the decryption of data.

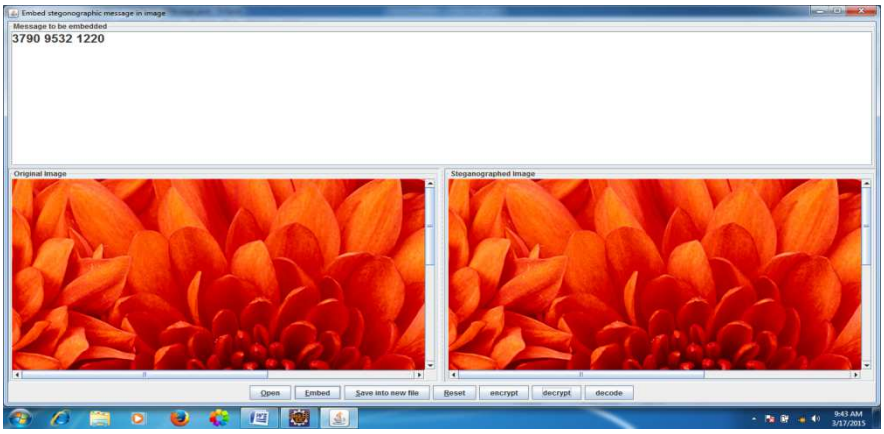

Figure 6: Hides the Data into Image

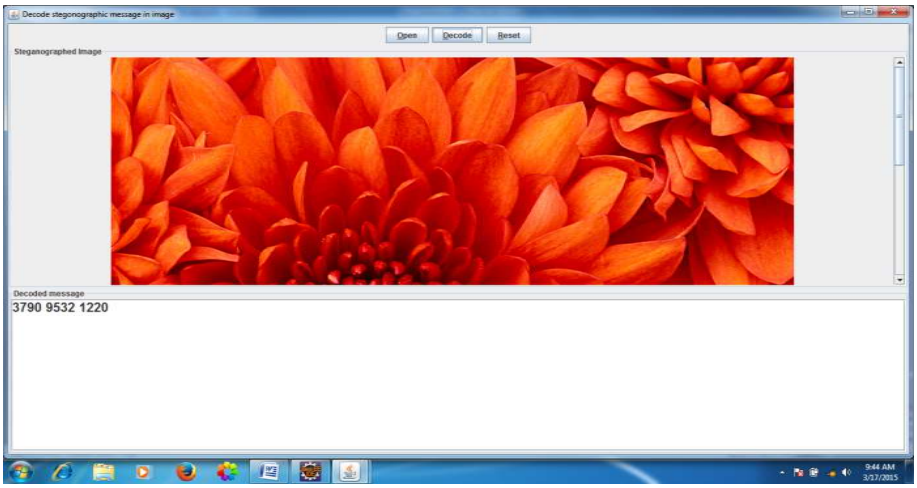

Figure 7: Unhides the Data from the Image

Figure 8: Before Encoding

Figure 9: After Encoding

CONCLUSIONS

Text hiding using RSA and Blowfish algorithms with HLSB technique has been presented. This technique is used to hide the text inside an image by converting it into cipher text using RSA encryption algorithm and then the encoded image is encrypted using Blowfish encryption algorithm.The security aspects of this proposed technique are quite improved as it considers three algorithms which will be difficult for the hacker to break the cipher as we used two secret keys in our project and it becomes difficult to know the pixels where the text is embedded as we used HLSB encoding technique.

ACKNOWLEDGEMENTS

We express our sincere and profound gratitude to our principal Dr. V.V. Rama reddy, our head of the department A.Rama Rao and our guide Mrs.D.M.S.Madhuri for their valuable guidance and support. Our sincere thanks to co-faculty members and our classmates who have supported us in completion our project.

REFERENCES

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.