Hiding Data into Reserve Space before Image Encryption using Blowfish Algorithm

Pradnya P. Mandlik
Department of Computer Engg.(B.E)
Dr.DY.Patil Institute of Engg and Technology
Ambi,Talegaon Dabhade,Pune.

Samruddhi S. Mhatre
Department of Computer Engg.(B.E)
Dr.DY.Patil Institute of Engg and Technology
Ambi,Talegaon Dabhade,Pune.

Samiksha M. Niwal
Department of Computer Engg.(B.E)
Dr.DY.Patil Institute of Engg and Technology
Ambi,Talegaon Dabhade,Pune.

Priti Mithari
Department of Computer Engg.
Dr.DY.Patil Institute of Engg and Technology
Ambi,TalegaonDabhade,Pune.

ABSTRACT

Nowadays, with the fast development of information technology more images and data are available on the internet. Hence there is a need to have some kind of authentication to that datawith increase in technology. This paper is about encryption and decryption of images using a private-key block cipher called 64-bits. Blowfish designed to increase security and to improve performance. This algorithm will be used as a variable key size up to 448 bits. The reversible data hiding (RDH) in encrypted images, is most powerful technology than others. Reversible Data Hiding (RDH) maintains the tremendous property that is the original content can be losslessly recover after embedding data. Reversible Data hiding phenomena is defined as a scheme that allows complete and visionless restoration of the original host data. All existing method of embedded data by reversible vacating room from the encrypted images which may have some faults on the data extraction or image restorations. The reversible data hiding is also known as the new watermarking technique which is used to derive an image by embedding some data on it as a watermark. A novel method is proposed by reserving room for embedding data before encryption of an image takes place with the available RDH algorithm and methods. Now the authentic person can secrete the data definitely on the image to provide authentication. Reversible technique is in the sense extraction of the original input image.

General Terms

Statistical analysis, Security, Algorithm

Keywords

Block Cipher, Cryptography, Feistel Network, Cryptography, Data Hiding, Steganography, Watermarking.

1. INTRODUCTION

Data is elicitated from and image by using Reversible Data Hiding techniques while data, is extracting from image after we get image as it is without looking its original features for data hiding we use stenography also, but in most case some distortion occur and if cannot be renovated back to its original media. Reversible Data Hiding (RDH) provides the exact image after extraction of data and improved that image into original form. If user wants to send an image through internet them he has to encrypt that image before sending through net. Reversible data implating can be used as information's. As the difference between the embedded image and original image is almost invisible from human eyes, reversible data embedding could be presumed as a hidden communication channel. By embedding its message verification code, reversible data embedding provides a real identity verification order, without the use of metadata. Reversible Data embedding it is also Called Lossless Data Embedding. The purpose of reversible data embedding is misrepresentation free data embedding.In most cases of data hiding, the cover images will experience some distortion due to data hiding and cannot be reversed back to the original form. Reversible Data Hiding is widely used in applications like medical, military and law forensic fields, misrepresentation of cover images does not allowed.

2. LITERATURE SURVEY

While embedding data some distortion is get created so this optimal balance between the amount of an information and the induced distortion get studied in [2] in this RDH technique use with the help of RDH we can remove the distortion of data. We prove the some result on capacity of reversible data hiding scheme. A reported a novel approach based on reversible data hiding, the particular encrypted image is divided into three blocks. The half portion of each block in reversed by three LSB's, room can be vacated for the embedded bit. The data extraction and image recovery keep by finding which part has been reversed in one block [5]. This process can be achieved with the help of spatial link in decrypted image. A framework of protection system for secret data communication through encrypted data cover - up in encrypted images. A reversible data hiding method based on difference expansion. In his work, the cover image was divided into a series of non-overlapping, neighboring pixel pairs, and the difference of each pixel pair was doubled [4]. Then, the doubled difference was either kept reserved or modified according to the parity of the embedding secret bit. On the receiver side, the embedded private data can be extracted easily from the least significant bit (LSB) of the differences of the pixel pairs in the stego image [5]. But the additional information of the location map was needed to solve the underflow and overflow problems. The lossless data embedding has the property that the distortion due to embedding can be completely removed fromthe watermarked image without procuring any side channel [4]. This can be a very important property whenever serious points over the image quality and artifacts visibility occurs, such as for medical images, due to legal reasons, for military images used as vital evidence in court that may be viewed after enhancement and zooming.

3. EXISITING SYSTEM

The losslessly vacating room from the encrypted images is relatively difficult and sometimes inefficient. the RDH tasks in encrypted images would be more natural and much easier which leads us to the novel framework, "reserving room before encryption (RRBE)".In figure 1, the transmitters have to give image as an input. This image is encrypted with the help of encryption key, after that with the help of data hiding key data is embedded by vacating space and then transmission of that image is take place over secure channel. Once the image is received by the receiver on receiver side, he has to extract the data with the help of secret data hiding key. So in this way receiver extract the secret data with the help of secret data hiding key for getting original image, he have to recover the image with the help of encryption key.

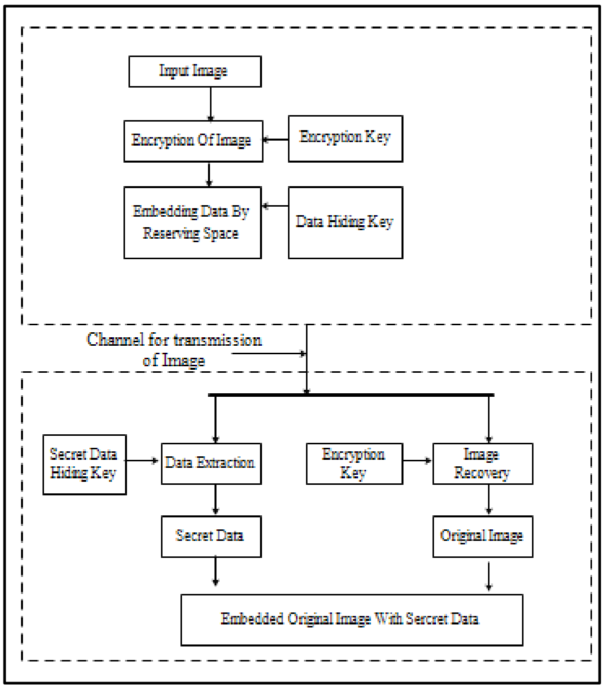

Fig1:- Reserving Room after Encryption

the help of above concept that is vacating image after encryption for RDH.

4. PROPOSED SYSTEM

In the proposed system, the content owner first reserves the enough space on original image into its encrypted version with encrypted key. Next the data is encrypted using Blowfish algorithm and embedding process starts the RDH tasks in encrypted images would be more usual and are much simpler which leads us to the framework, **"Reserving Room Before Encryption (RRBE)".**

4.1 Different Steps of RDH:

4.1.1. Generation Of Encrypted Images:

We have to divide that image into two parts the LSB, s of A are reversibly embedded into B with a standard RDH algorithm.

4.1.2. Data Hiding In Encrypted Image

Once the data hider obtains the encrypted image. He can embedded data into it, he does not have an access to original image. If anyone who does not possess the data hiding key could not remove the additional data.

4.1.3. Data Extraction and Image Recovery

Extracting Data From Encrypted Images

4.1.3.1. Extracting Data From Encrypted Images:

In order to renew and achieve personal information of images that are encrypted for protecting clients' privacy, an substandard database manager may only get access to the data hiding key and have to manipulate data in encrypted domain. The order ofdata extraction before image decryption assurancesthefeasibility of our work in this case.

4.1.3.2. Extracting Data From Decrypted Images:

In the previous case both inserting and removal of the data are manipulated in encrypted domain. While there exists a different situation where the user wants to decrypt the image first and remove the data from the decrypted image when it is needed. Alice out-sourced her images to a cloud server is one of the most important example of such application. images are encrypted to save from harm their contents.In Figure 4.1, the transmitter has to give input as an image. Before performing any operation on data he has to reserve room or space for image. Encrypt the image with the help of encryption key. Data is embedded with that image with the help of data hiding key. That embedded image get transferred over a secure channel. Once the receiver receives that image data get extracted with the help of secret data hiding key, then by using encryption key recover the original image. So in this way receiver retrieve the secret data and original image with the help of above concept i.e. vacating space before encryption for RDH channel. Once the image is received by the receiver on receiver side, he has to extract the data with the help of secret data hiding key.

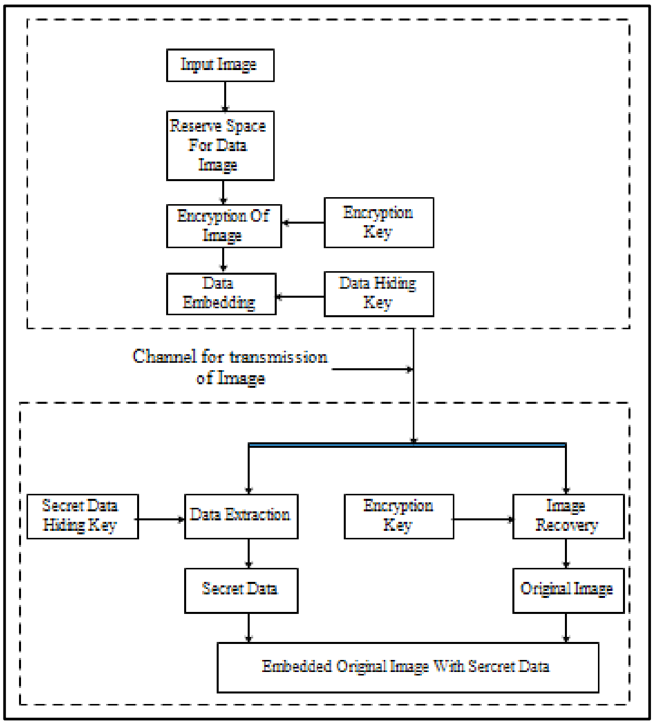

Fig 2:- Reserving Room before Encryption

So in this way receiver extract the secret data with the help of secret data hiding key. For getting original image, he has to recover the image with the help of encryption key.Thus we canretrieve secret data and original, image with the help of above concept that is vacating image after encryption for RDH.

5. SYSTEM ARCHITECTURE

We can use the security key while encrypting an image, but the security of that key is very less so to improve the security of key we can use linear feedback shift register (LFSR).Now we can expand the concept of "Reserving Room Before Encryption" (RRBE) using LFSR, which contain mainly four stages such as a)Generation of encrypted image, b)data hiding in encrypted image c) data extraction d) image recovery.The sender first have to reserve some space for data hiding and encrypt that image using encryption key ,for generating LSFR code data hider use LSFR .In LSFR Pseudo random no generate. The random no generated through LSFR code repeat itself 2n-1 .In this case we have to use two keys that is encryption key and data hiding key. Though data hider doesn't know the content of original image then also he can added the extra information into encrypted image with the help of data hiding key. In this case receiver must know the encryption key as well as the data hiding key. Content decryption is nothing but the extraction of data .If receiver having a data hiding key but he doesn't know about the encryption key, then he does not to extract the knowledge from the encrypted image, which content extra idea added. For creating a space data hider have to compress the Least Significant Bits (LSB). of encrypted image. With the help of LSFR we can generate good key stream.

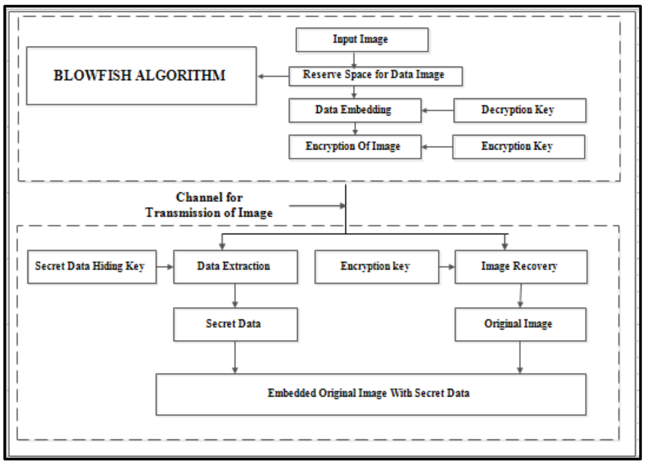

Fig 3:-System Architecture

Though **losslessly** vacating room from the encrypted images is bit difficult and sometimes inefficient. Before encryption owner have to reserve room first .Reversible data hiding provide exact image after decryptions. In this system owner first have to occupy some space for image i.e. he have to reserve room for the encryption. Then encrypted that image with the help of encryption key. For encrypting any image first we have to divide that image into 3 parts i.e. deviation of image, self-reversible embedding, encryption of image.so in that dividation of image divide that image into 2 parts. The **LSB** of first part get embedded with **LSB** of second part, by using **RDH** technique. After that encrypt that image and produce the final version. After that owner have to reserves room before encryption .In self-reversible embedding the LSB of second image is embedded into **LSB** of first image by using data hiding technique. After that image is encrypted with th help of encryption key. If third person wants to get the content of image then without encryption key he cannot able to get it.thus we can provide privacy to our data or image. After that data get hide inside that image. If anybody cannot have the data hiding key then he is not able to get the access for addition of extra data. When anybody wishes to extract that data then first he have to decrypt the **LSB** planes. So in this way we can secure our data by using method of encryption along with the concept of reserving room.

5.1 Blowfish Algorithm

Bruce Schneier designed blowfish in 1993 by Bruce Schneier as a fast, free alternative to existing encryption algorithms. *Blowfish* is unpatented and license-free, and is available free for all uses. Blowfish Algorithmis a Feistel Network, iterating a simple encryption function 16 times. The block size is 64 bits, and the key can be any length up to 448 bits.*Blowfish* is a variable-length key block cipher.The elementary operators of Blowfish algorithm include table lookup, addition and XOR. The table includes four S-boxes and a P-array. Blowfish is a cipher based on Feistel rounds, and the design of the F-function used amounts to a simplification of the principles used in DES to provide the same security with greater speed and efficiency in software. Blowfish is a 64 bit block cipher and is suggested as a replacement for DES. Blowfish is a fast algorithm and can encrypt data on 32-bit microprocessors. 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.

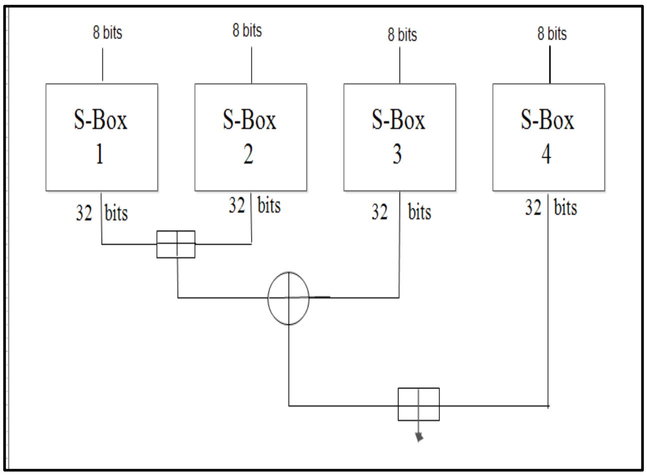

Fig 4:-Blowfish Algorithm

Blowfish consists of three parts:

1. Encryption algorithm
2. Key-expansion
3. Decryption algorithm

5.1.1 Encryption Algorithm

There are the P-arrays, which has eighteen 32-bit boxes, and the S-boxes, which are four 32-bit arrays with 256 entries each. All of these boxes are initialized with a fixed string, the hexadecimal digits of pi [10]. After the string initialization, the first 32 bits of the key are XOR with P1 (the first 32-bit box in the P-array). The second 32 bits of the key are XOR with P2, and so on, until all 448, or fewer, key bits have been XOR. Cycle through the key bits is completed by returning to the beginning of the key, until the entire P-array has been XOR with thekey [12].

Blowfishhas 16 rounds.

The input is a 64-bit data element, x.
Divide x into two 32-bit halves: xL, xR.
Then, for i = 1 to 16:
xL = xL XOR Pi
xR = F(xL) XOR xR
Swap xL and xR
After the sixteenth round, swap xL and xR again to undo the last swap.
Then, xR = xR XOR P17 and xL = xL XOR P18.
Finally, recombine xL and xR to get the ciphertext.

5.1.2 Key Expansion

A symmetric Encryption key is used for this application, which means the same key is shared for both Encryption and decryption.

5.1.3 Decryption Algorithm

The encrypted image is divided into the same block length of Blowfish algorithm from top to bottom.The first block is entered to the decryption function and the same encryption key is used to decrypt the image but the application of sub keys is reversed. The process of decryption is continued with other blocks of the image from top to bottom.

The basic algorithm for Blowfish is illustrated as follows:

Divide X into two 32-bit halves XL and XR

For i=1 to 16:

XL = XL Pi

XR = F (XL) XR

Swap XL and XR

End for

Swap XL and XR

XR = XR P17

XL = XL P18

Recombine XL and XR

Output X (64-bit data block: cipher text)

For decryption, the same process is applied, except that the sub-keys Pi must be supplied in reverse order. The nature of the Feistel network ensures that every half is swapped for the next round.

Decryption is the same as encryption, except the P-arrays are used in reverse. Hence, Blowfish encrypts by splitting half the block (32 bits) into 8-bit chunks (quarters) and inputting this into the S-box. The result from S-boxes then are added and XOR. The S-boxes and P-boxes are initialized with values from hex digits of pi. The variable length user-input key is then XOR with P-entries. Then a block of zeros is encrypted, and this result is used for P1 and P2 entries. The cipher text resulting from the encryption of a zero block is then encrypted again and use for P3 and P4. This process continues until every P-box entry and S-box entry has been replaced, resulting in 521 successive key generations. This involves about 4KB of data processing. This relatively complex key schedule makes Blowfish an effective and durable cryptographic algorithm .Blowfish is among the fastest block ciphers available and yet remains cryptographically secure.

5.1.4Mathematical Module:

1. Let S be a system that describes BLOWFISH Algorithm
S= {...}
p= {p0, p1, - - - -, p17}
Where p the Elements in the Array.
2. Identify input as I
S ={I,..}
Let I =i1, i2, i3, id
The input will be Encryption Key, Decryption key, Original Image
3. Identify output as O
S = {I, O}
Output will be encrypted Data, Decrypted Data.
4. Identify the process as P
S = {I, O, P…}
P= {S, X)
F_R = {0, 1, - - - - - , 15}
K= { K_0, K_1…, Kn}

Where,
Let F be the round function.
Process will be S-Box Process and XOR Process to Input Bits.
K be the sub-keys

4. Identify Failure as F
5. S={I,O,P,F,..}s

F= Failure occurs when the system is not Decrypt a authorized data.
6. Identify success as s
S= {I,O,P,F,s, ..}
s= when the system Decryptan authorized data.
7. Identify the initial condition as Ic
S= {I, O,P,F,s,Ic}
Ic= Key is required for Encryption & Decryption.
8. Split the plaintext block into X_L, $X_{R,}$ into two 32 bits
xL = xL XOR Pi
xR = F(xL) XOR xR
Swap xL and xR
After the sixteenth round, swap xL and xR again to undo the last swap.
Then, xR = xR XOR P17 and
xL = xL XOR P18.
Finally, recombine xL and xR to get the ciphertext.

6. CONCLUSION

The Reversible Data Hiding(RDH) determines that the data and image can be rebuild back to its original state using image expansion. Since we are not changing any original pixels, the image can be restored without loss of quality. **Reversible Data Hiding (RDH)** having ability to recapture the cover without any alteration less. We are using simple **LSB** method in which the security of data is very rarer. The Novel method can advance dispersed data extraction, real reversibility and significantly enhancement on the quality of obvious decrypted images. We divide an image into pixel values, select foldout difference numbers for difference expansion and embed a payload which includes an authentication hash.Blowfish cannot be broken until an attacker tries 28r+1 combinations where r is the number of rounds. Hence if the number of rounds are been increased then the blowfish algorithm becomes stronger. Since Blowfish has not any known security weak points so far it can be considered as an excellent standard encryption

7. FUTURE SCOPE

In the system, if the amount of data to be inserted into the image is very large then the size of expanded image will also be very large. As a result overhead for transferring the image will increase. To improved data hiding with maximum length and equate the embedded rate with image and video.

8. ACKNOWLEDGMENTS

This Paper present a concept of image encryption by using Reserving Room Before Encryption of image. It uses the reversible data hiding technique. People will get benefit for securing their data by using these concepts. The content in this paper is the quantribution of many peoples. Here we acknowledge and give special thanks to those who contributed the most.

We convey our sincere thanks to Prof. Priti Mithari.
Despite the assistance of so many people and our best efforts, there are still likely to be some unintentional omissions or errors. We would therefore highly appreciate any useful suggestions and corrections form the reader.

9. REFERENCES

[1] PradnyaMandlik, SamruddhiMhatre, Samiksha Niwal, “Reversible data hiding using encryption and data hiding Technique” IEEE Trans. On information forensic and security,September 2015.

[2] Kede Ma, Weiming Zhang, XeianfengZhao,”Reversible data hiding in encrypted images by reserving room before encryption” IEEE Trans. On information forensic and security,March 2013.

[3] T. Kalker and F.M.Willems, “Capacity bounds and code constructions for reversible data-hiding,” Digital Signal Processing(2002), 2002.

[4] W. Zhang, B. Chen, and N. Yu, Improving various reversible data hiding schemes via optimal codes for binary covers," IEEE Trans. Image Process, 2012.

[5] J. Fridrich and M. Goljan, “Lossless data embedding for all image formats,” in Proc. SPIE Proc. Photonics West, Electronic Imaging, Security and Watermarking of Multimedia Contents, San Jose, CA, USA, Jan. 2002, vol. 4675, pp. 572–583.

[6] W. Zhang, B. Chen, and N. Yu, Capacity-approaching codes for reversible data hiding," in Proc 13th Information Hiding (IH'2011), LNCS 6958, 2011, pp. 255-269, Springer-Verlag.

[7] L.Luoet al., “Reversible image watermarking using interpolation technique,” IEEE Trans. Inf. Forensics Security, vol. 5, no. 1, pp. 187–193, Mar. 2010.

[8] D.M. Thodi and J. J. Rodriguez, “Expansion embedding techniques for reversible watermarking,” IEEE Trans. Image Process., vol. 16, no. 3, pp. 721–730, Mar. 2007.

[9] Xinpeng Zhang, “*Reversible Data Hiding in Encrypted Image*”, IEEE Signal Processing Letters, vol.18, No.4, April 2011.

[10] Z. Ni, Y. Shi, N. Ansari, and S. Wei, “Reversible data hiding,” IEEE Trans. Circuits Syst. Video Technol., vol. 16, no. 3, pp. 354–362, Mar.2006.

[11] Xiaolong Li, Bin Yang and Tie yongZeng, “Efficient Reversible Watermarking Based on Adaptive Prediction-Error Expansion and Pixel Selection”, IEEE Transaction on Image Processing, Dec 2011.