Advanced Diffusion for Image Authentication

Rupali Kamalkishor Lakhotiya
M.E, 2nd Year: Information Technology Department
Siddhant College of Engineering, Pune, India
Email-id: rupalilakhotiya@gmail.com

Prof. Rashmi Deshpande
Assistant Professor, Information Technology Department
Siddhant College of Engineering, Pune, India
E-mail-id: rashmi2810@gmail.com

***Abstract*— Information hiding techniques gain much more attention from the researchers and from cryptography communities. Until now cryptographic algorithms are only applicable for encoding and decoding of data. In this phenomenon we are achieving cryptography along with information hiding. Here we are going to use various types of algorithms for encryption. For hiding of information we are going to use stochastic diffusion approach. We are going to encrypt image using the application of LSB (least significant bit). Here we have also used algorithm of watermarking which is responsible for hiding image in single binarized image. The information which we want to share is encrypted first and then it is hide into another cover image. The information hiding again is heterogeneous information where there is no concern whether the information being hidden is a data file or an image file or a video file. At receiver side cover image will be received and then using stegnographic algorithm hided data will be extracted and recovered. After recovering or extracting the data or image, the obtained image will be decrypted using symmetric algorithm used during encryption process. The decrypted data is then used for the intended purpose, may it be a data file or an image or a video file. In the proposed system, we eradicate the limitations of sending or hiding only limited kind of data. Research work in the proposed system includes LSB Data encryption and data hiding techniques for various data.**

Keywords- Encrypted Information Hiding; Stochastic Diffusion; Hidden Codes; watermarking algorithm.

I. INTRODUCTION

In computer era, digital image is a base of many security systems. In old days image is only responsible for storing precious moments. But now a day's image is used in every application for various purposes. Image processing plays an important part in digital world. Image is being used for authentication system. In this digital world or digital era, image is also used for encryption as well as compression methodologies. Now image becomes an efficient and reliable way of sharing data in secret manner. The data which we want to be shared secretly is encrypted first using efficient encryption algorithms. Then this encrypted data is hided into cover image. Cover image is a host image which will get shared on network.

Digital image consist of watermarks. We are hiding data into watermarks. Normally image is a set of x and y pixels and each pixel is having separate color, position and property. Pattern matching algorithm is initially considered as stochastic diffusion algorithm. The algorithms in the category of swarm intelligence and naturally inspired search and optimization algorithms includes this algorithm. The phenomenon of ant colony optimization, genetic algorithms and particle swarm optimization is used for diffusion. We can achieve cryptography using stochastic diffusion. If we encrypt any plain text with same key then it will produce different cipher text. This derivation of key from cipher text is known as diffusion. This is done to prevent the attacker from accessing data. Attacker may inspect the plain and cipher text to guess the key for encryption.

II. LITERATURE SURVEY

The existing image authentication papers fail to achieve all the requirements efficiently. It cannot store lossy format into pixels. And therefore the Results produced by previous techniques lack in efficiency that gives an area for improvement in the image authentication mechanism.

A. Cryptographic Image Authentication:

Reference [1] Uses the digital signature algorithm for image encryption for security purpose. It encrypts host image and share the same on the network. There is no extra mechanism used for image encryption rather than digital signature.

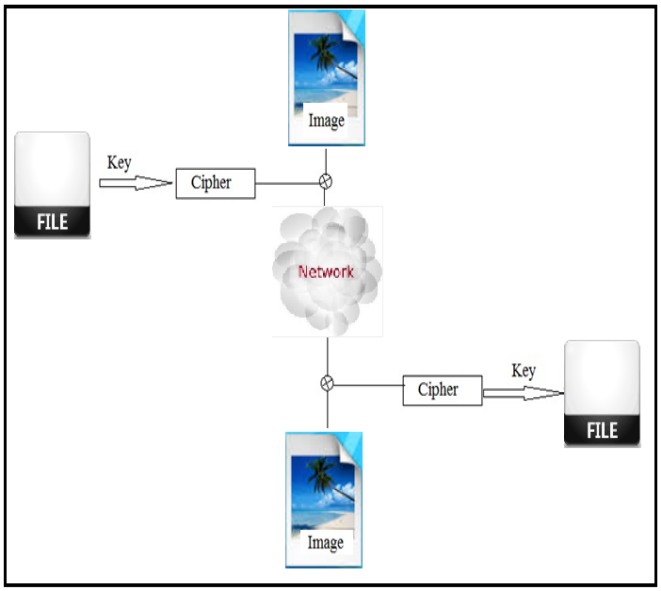

Figure 1. Image Encryption Authentication.

B. Distributed source coding:

It uses Wolf coding projections for authentication of image. For robustness in image coding distributed source

coding algorithm is used. Authentication is done on the basis of contrast, intensity and Brightness of image.

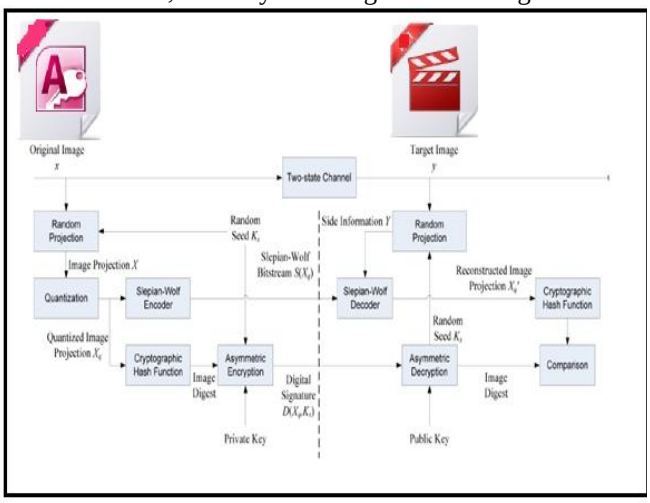

Figure 2. Distributed source coding

C. *Neural networkImage Authentication:*

Authentication key and data are used as neural code reference [3]. This is compared with media information, and small size security parameter.

Then key ad security parameter is shared through secure median and remaining coded data is transmitted over public network. Drawback of this system is that malicious attacker can damage coded data. Then evaluating the original code and the computed code, the result is generated. If they are having small difference then data is secure otherwise, it is tampered.

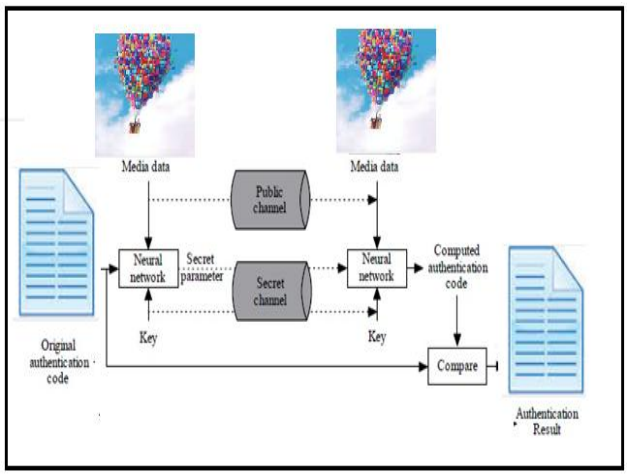

Figure 3. Neural Authentication

D. *DCT with recovery capability:*

[4] Uses DCT method for diffusion with recovery. It detects the tampered parts of the image and apply recovery on that part. It uses two water marking algorithm. One semi-fragile watermark, second watermark is use for making recovery efficient while first is used for the authentication phase.

Watermark generation algorithm:

1. Host image is divided into two part; this is called I.
2. Then convert image to grey level and reduce 127 from each pixel of grey scale of I to force pixel towards [-127,128].
3. Divide I into 8x8 components or block.
4. Calculate 2D-DCT forevery 8x8 block.
5. Retained first sixteen DCT coefficients from each block in zigzag order.
6. Round DCT coefficient to the nearest integer of 7 bit with including sign.
7. Quantized DCT coefficients are using the JPEG quantization matrix having quality factor equal to 50 before being encoded.

III. PROPOSED SYSTEM

In proposed system we are going to encrypt the secret image information and then hide it into cover image. This cover image will transmitted freely over public network. The key and other encryption information will send through secure private network.

A. *Encryption Module:*

The secret information is being encrypted in this module before sending it over network. Encryption is done using consistent key and diffusion. This key is shared using secure network.

B. *Heterogeniety of Data to be Hidden:*

The existing system previously worked only on an input image that may display some information and that image was to be converted into another image so as to hide the important data in the original image. But the image being encrypted is either 8 bit or 24 bit, and it will hardly contain some small amount of data. So to transmit large amount of data the existing system had to use multiple images. But the proposed system focuses on transmitting large amount of data over the single image irrespective of it being 8 bit or 24 bit. We can hide various kind of data like audio file, video file or a data file behind the image and there by further apply LSB steganography to encrypt the image.

C. *Hiding using stochastic diffusion:*

We can achieve uniform diffusion using stochastic diffusion method for encryption. It generates random number which will be used as private key. For encryption of this host image we apply Advance Encryption Standard (AES) on image. Arithmetic compression method will be applied on encrypted image which will stochastically diffuse the image into another. Then data is compressed and converted into the binary string. Then encoded in the image using the bit-plane.

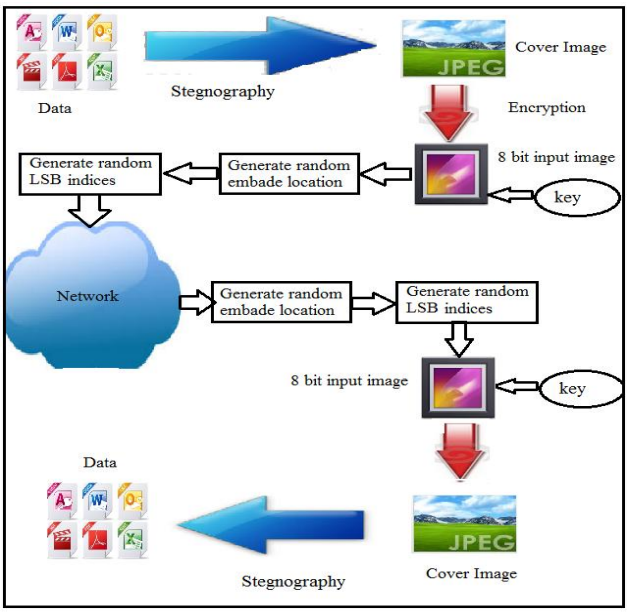

Figure 4. System Architecture.

D. Hiding Codes for binary image watermarking:

In order to increase the security level for the data it uses least significant bit method (LSB). Which will help to improve efficiency of the stochastic diffusion. It uses three algorithms Uniform distributions for generation of hidden codes, Log-normal method, and Gaussian distribution algorithm.

IV. MATHEMATICAL MODEL

The Image cryptography and steganography can be represented mathematically with the help of DFA representation. The Mathematical model of the proposed system can be formulated as follows:

Here the heterogeneous data gets hidden behind the image in an encrypted format. The encryption algorithm used to encrypt the data is AES algorithm.

Consider an image I is given as an input to the data hiding module £, the data may be any kind of data represented by Δ, then the Equation to be formed for data hiding is as follows:

$$H_i = £ (\Delta, I, \alpha, \sum, €) \qquad (1)$$

Where H_i stands for Hidden data image.

Δ - represents data to be hidden.

I – Image which is to be encrypted and diffused.

α – Represent the bitwise operations being performed over the image.

∑ - represents the collections of pixels.

€ - represents the final stochastically diffused image.

The various bitwise operations being performed are explained further in subsection E of the section IV.

The obtained hidden data image H_i *is* then encrypted using LSB Steganography algorithm so as to obtain a new encrypted image. This image is then transmitted through as secure channel.

AES Encryption Algorithm

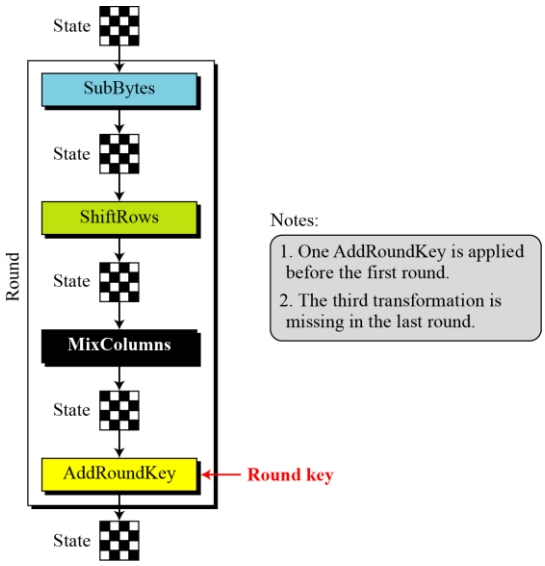

Figure 5. AES algorithm

To provide security, AES uses four types of transformations: substitution, permutation, mixing, and key-adding.

A. Subbytes:

It is used at the encryption side. To substitute a byte, we interpret the byte as two hexadecimal digits.

B. Shift rows:

Transformation is di=one while doing encryption is called ShiftRows.

C. Mix Coloumns:

The MixColumns transformation operates at the column level; it transforms each column of the state to a new column.

D. AddRoundKey

AddRoundKey proceeds one column at a time. AddRoundKey adds a round key word with each state column matrix; the operation in AddRoundKey is matrix addition.

E. LSB Algorithm:

LSB (least significant bit) is the very easiest and most simple steganography type. For encoding the information which is hidden, the one's bit of a byte is used. Expect in the following 8 bytes of a carrier file, we wish to encode the letter A (binary 01000001 or ASCII 65).

01011101 11010000 00011100 10101100
11100111 10000111 01101011 11100011

Becomes,

01011100 11010001 00011100 10101100
11100110 10000110 01101010 11100011

In a single byte, the bits have a rank according to binary standards, the left most bit is recognized as the most significant bit and right most is recognized as least significant. This gives us the trigger to use LSB as steganography medium, if we think of changing some data bits in the image, we focus that it should be as unobtrusive as possible, or even undetectable. Soto the least significant bit of some of the bytes we will apply our changes. In this way a maximum of 1 bit in value, we change each byte.

F. Encoding LSB Steps:

Here is how this logic accomplishes that:

1. *for(int i=0; i<addition.length; ++i)* – loops through each byte of the addition array
2. *int add = addition[i];* – assigns add to be the current byte
3. *for(int bit=7; bit>=0; --bit, ++offset)* – loops over byte's 8 bitswhich stored in add
4. *int b = (add >>> bit) & 1;* – b is assigned the value of the byte add shifted right bit positions AND 1

This can show to be much complex, but the ultimate result is a loop that iteratively assigns variable *'b'* the next one bit value of the data byte '*add'*, either *0*, or *1*.

Let's start with *int b = (add >>> bit);* for Time being consider:
add = 78 = 01001110;

First Iteration: bit = 7:
01001110 = 78
>>> 7
00000000 = 0

Next Iteration, bit = 6:
01001110 = 78
>>> 6
00000001 = 1

Next Iteration, bit = 5:
01001110 = 78
>>> 5
00000010 = 2

Next Iteration, bit = 4:
01001110= 87
>>> 4
00000100 = 4

Next Iteration, bit = 3:
01001110= 87
>>> 3
00001001 = 9

Next Iteration, bit = 2:
01001110= 87
>>> 2
00010011 = 19

… etc.
Note that how the left bits in obtained code are matched by the right bits in byte '*add'*, in a growing number which depends on how many location we shift *add.*

Now we apply the *'& 1'* operation:
First Iteration:
00000000 = 0
00000001 = 1
00000000 = 0 = b

Next:
00000001 = 1
00000001 = 1
00000001 = 1 = b

Next:
00000010 = 2
00000001 = 1
00000000 = 0 = b

Next:
00000100 = 4
00000001 = 1
00000001 = 0 = b

After notify pattern, *b* is appoint the value 1 or 0, according to the final bit of the moved '*add' byte.* By ANDing by '*1'*, we obtain the same, except the last which is left as it was, which clears all bits to 0. In the for loop this indicates that *b's* value shows the bit at position *bit*.

image[offset] = (byte)((image[offset] & 0xFE) | b);

This code works in a similar manner. *0xFE* is the hex, in binary, by *11111110* is represented by it. By this, the first 7 bits will leave it as is, and convert LSB bit to *0*. Then the last bit being *0*, is ORed with *b*, which is either: *00000001 or 00000000*. To match the value previously stored in *b* this set the last bit. As ORing operation with 0s will not affect any of the first 7 bits, and thereforesophisicated0 is the last bit, the value at current position of *b*, is definitely settled to be placed into this location, either be it 1 or 0.
We gradually increment the *offset* value as the loop enhances further also, so the single byte's *8 bits* addition are isolated across the 8 LSB Bits of *8*subsequent and bytes of the image.
Most important of all is that the length is encoded first, i.e. It gets stored in 4 bytes, or first 32 LSBs. Thus we can predict that how many LSBs to read after the length to encoded data.

G. Decoding Embedded LSB Bits:

1. *int offset = 32;* – The length of the data is stored as a 4 byte numeric value, or stored using 32 bits, thus the actual data starts after 32 bytes of image.
2. *for(int i=0; i<32; ++i)* – As discussed earlier the first 32 bytes contain *1* bit each of data length, we have to iterate all first 32 bytes to retrieve the encoded data length.
3. *length = (length << 1) | (image[i] & 1);* – We shift length bits left by *1,* then ORing it with a result of the LSB of the image byte. Further operation *'(& 1)'* will erase all bits, except the last one, that is kept unchanged. Thus as the added bits propogate, they are moved along and placed into the newly created empty LSB slot of length.
4. f*or(int b=0; b<result.length; ++b)* – As now we have retrieved the encoded data length and also created byte array to store the bits, we loop through equal amount or count of image bytes.
5. *for(int i=0; i<8; ++i, ++offset)* – Still we need to iterate through the *8* bits of a byte to get stored.
6. *result[b] = (byte)((result[b] << 1) | (image[offset] & 1));* – Finally the decoded data is retrieved and stored in *result* array

V. Result Analysis

The proposed system implemented so far gives a clear idea that the security mechanism for image encryption will work much better than the existed systems. The below figures show the results obtained so far for the implemented code which weights more than 60% of the entire project.

Figure. 6 Input Image

Figure 7. Encrypted Cover Image

VI. Expected Results

The proposed system not only defines mechanism to hide the secret image data into host image using stochastic diffusion, but also incorporates an additional facility to embed the various kind of data such as multimedia data like images, videos or audio. The data is embedded over the secret image and then the further operations under stochastic diffusion process are performed. Thus the proposed system is expected to securely encrypt the data being hidden and then diffuse the encrypted data or image in the host image using stochastic diffusion.

Conclusion

Thus we come to conclusion that initially watermarking algorithm is only responsible for encryption, and encryption alone is not secure enough for data transmission. So we have designed the robust method of hiding heterogeneous data inside the image. Information is in hidden format and the type of the data being hidden is again a mystery for an attacker, so attacker will not able to decrypt it. This method not only embeds the 8-bit image pixels into 24-bit pixels by using binarization property but also hides huge amount of the data inside the image and further enhance the security with the use of LSB steganography. The LSB compression method provides the high fidelity decrypt which use to avoid the cipher bits loss.

References

[1] Mahimn Pandya Hiren Joshi Ashish Jani. Novel Digital Watermarking Algorithm using Random Matrix Image International Journal of Computer Applications © 2013 by IJCA Journal Volume 61 - Number 2 Year of Publication: 2013 Authors:

[2] Dr. Krishna Mohanta, 2,Dr.V.Khanaa Image Authentication Using Distributed Source Coding International Journal Of Computational Engineering Research (ijceronline.com) Vol. 3 Issue. 1

[3] Shiguo Lian. Image Authentication Based on Neural Networks SAMI Lab, France Telecom R&D Bejing Beijing, P.R China, 100080

[4] El'arbi, M. ; ENIS, Univ. of Sfax, Sfax, Tunisia ; Ben Amar, C Image authentication algorithm with recovery capabilities based on neural networks in the DCT domain. IEEE Volume:8 Issue:11

[5] Fei Shao ; Dept. of Inf. Technol., Jinling Inst. of Technol., Nanjing, China ; Zinan Chang ; Yi Zhang AES Encryption Algorithm Based on the High Performance Computing of GPU IEEE International Conference on Embedded and Ubiquitous Computing (HPCC_EUC), 2013 IEEE 10th International Conference on, On page(s): 504 – 510.s