Performance Review of Image Encryption Methods

Fiona Rozario and Dr. Anup Vibhute

Dept. of Electronics and Telecommunications, Dr. D. Y. Patil School of Engineering and Technology, Pune University, India

Abstract

Images have found widespread use in today’s world from social networking, banking, medical to military uses. Hence, to maintain confidentiality of these images, image encryption becomes an integral part of any system. This paper reviews some of the image encryption techniques and compares the performance of these techniques with respect to different parameters.

Keywords: Image, Encryption, Image encryption

I. INTRODUCTION

Images have found widespread use in today’s world from social networking, banking, medical to military uses. Hence, to maintain confidentiality of these images, image encryption becomes an integral part of any system.

Image encryption is different from encrypting text information and hence the existing encryption methods for text cannot be used as is for images. Firstly, images are larger in size than text and hence image compression is needed. Secondly, text encryption methods are block or stream ciphers which process a one-dimensional array of characters; whereas, images are two-dimensional arrays of pixels. In order to be encrypted, images must be converted to one-dimensional arrays. Lastly, text encryption is lossless. On the other hand, some losses in image encryption and decryption can be tolerated.

Moni Naor and Adi Shamir, the pioneers of visual encryption, designed and proposed the ‘share-based’ visual encryption method at Eurocrypt in 1994. The method divides an image into n parts (or shares) such that the knowledge of a minimum k or more number of shares makes the decryption computable, else the decryption fails [2].

Since the share-based method developed by Moni Naor and Adi Shamir, many more techniques have been developed for image encryption. These techniques can be classified as: (i) techniques requiring keys: vector quantization, chaos based encryption, AES based techniques etc. depend on a secret key for encryption and decryption. (ii) keyless techniques: these techniques divide an image into n number of shares and knowledge of a qualifying number of shares makes the decryption computable, else the decryption fails [2].

Section II of this paper reviews five visual encryption techniques. The techniques are namely: Modified AES algorithm, vector quantization, hash based algorithm, authentication using digital signatures and the pixel sieve algorithm. In section III, the above techniques are compared on the basis of PSNR, SNR, correlation between pixels in encrypted and original image, histograms of encrypted and original image and entropy. One of the variables, which determine the loss in image quality, is Peak Signal To Noise Ratio, PSNR. PSNR measures the magnitude of change in pixel values between the plaintext and ciphertext images. Mathematically [9],

$$PSNR = 10 \times \log_{10} \left( \frac{M \times N \times 255^2}{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (P(i,j) - C(i,j))^2} \right)$$

Higher the PSNR value, better the reconstruction quality [16]. Finally, section IV concludes the paper.

II. REVIEW OF ENCRYPTION TECHNIQUES

A. Modified AES algorithm

The AES algorithm uses plaintext of 16 bytes and a key of length 16, 24 or 32 bytes. Each plaintext block undergoes the functions as shown in Fig. 1.

![Fig. 1 AES Algorithm](image1.png)

The length of the key dictates the number of rounds, Nr. The AddRoundKey is a simple XOR between the plaintext and the round key. SubBytes substitutes bytes from an S box in place of the plaintext (achieving confusion). ShiftRows performs circular left shifts on the rows of the plaintext matrix (achieving diffusion). In MixColumns function, each byte of a column is mapped into a new value that is a function of all four bytes in that column (producing nonlinearity).

To improve the encryption performance, particularly the correlation among the adjacent pixels, histogram of the encrypted image and the entropy of the image, the modified AES algorithm makes use of a key stream generator as shown in Fig. 2.

![Fig. 2 Modified AES Algorithm](image2.png)
The key stream generator may be either the A5/1 or the W7 key stream generator [7].

The A5/1 key stream generator uses three Linear Feedback Shift Registers (LFSR) of lengths 19, 22 and 23 bits. All three LFSRs use majority functions (C1 – bit 8, C2 – bit 10 and C3 – bit 10) and use different tapping bits [13].

After three rounds of initialization, the output at each clock cycle is obtained by an XOR of the last bits of each register. Its limitation is of a small key space of since the key length is 64 bits and it is also vulnerable to a brute force attack given today’s computing speeds. Fig. 3 shows the A5/1 LFSRs.

![Fig. 3 A5/1 LFSRs](image)

The W7 is a stream cipher supporting key lengths of 128 bits. It essentially overcomes the weakness of A5/1 key stream generator by increasing the key length from 64 to 128 bits. It also uses majority functions C0 - C7 as shown in Fig. 4 [10].

![Fig. 4 W7 keystream generator](image)

On performing the statistical analysis and comparison with the AES algorithm, the authors conclude that AES modified with the W7 key generator provides better security in terms of correlation between adjacent pixels and entropy [7].

### B. Vector Quantization

Each pixel in an image has RGB components, which can be organized into a column vector.

\[
z = \begin{bmatrix}
\text{red} \\
\text{green} \\
\text{blue}
\end{bmatrix}
\]

Thus, an RGB image of size M x N can be represented by three component images of size M x N or a total of MN 3-D vectors.

The aim of vector quantization (VQ) is to represent data in a more compact manner by approximating continuous vector values, with discrete values. For example, colour intensities in the range of 10-12 can be substituted by 11. These discrete values are provided by a codebook.

To use VQ [3]: (i) the image is first decomposed into m vectors, \[X_1, X_2, \ldots, X_n\]. (ii) A codebook is then chosen such that there is least distortion between the vector \(X_i\) and the quantized value selected \(A_j\) (this is measured in terms of least Euclidean distance between \(X_i\) and \(A_j\)). (iii) The index, \(i\) of \(A_j\) and the codebook is transmitted using the conventional encryption methods.

There are two ways to transmit the indices and the codebook.

(i) Encrypt the indices by commercial cryptosystems and send the codebook as plaintext (fig. 5)

(ii) Encrypt the codebook and send the indices as plaintext. (fig. 6)

![Fig. 5 VQ by encrypting indices](image)

The advantage of the VQ method is that it compresses the image into a set of indices from the codebook, hence saving storage space and channel BW.

VQ also has some limitations [4]. Selection of the training vectors to design the codebook and the computational complexity in the design of the codebook are the major challenges in using VQ. The design computation of the codebook involves the distribution of the intensity mean and the standard deviation, \(\sigma\) of the vectors in the image.

Codebooks can be classified as adaptive and non-adaptive. Non-adaptive codebooks have uniform vector entries based on the average vector intensities, \(\mu\). Hence, for an 8 bit imaging system, the codebook values are uniformly placed between [0, 255]. The adaptive codebook classifies vectors as edge and non-edge vectors based on a threshold of standard deviation, \(\sigma\).

A comparative study of SNR vs. ICR (Image Compression Ratio) for adaptive and non-adaptive codebooks has been performed by Yong You and Ahmed Desoky, which shows that the experimental and expected SNR values are close and the difference is insignificant in case of the non-adaptive codebook [4].

### C. Image Encryption Using Digital Signatures

This is a simple technique proposed, where the digital signature of the original image is calculated and added to the encoded image. The purpose of a digital signature is to authenticate a message. The ‘signature’ of a message is an entity computed from the message [20]. The digital signature can be computed using any one of the standard algorithms like MD2, MD4, MD5, or the SHA algorithms.

The image is encoded using an appropriate error control code, which adds redundancy bits to the image so that bits corrupted by noise can be recovered. In essence, the digital signature acts as the noise, which is XORed with the image. The authors have used MD5 algorithm to compute the digital signature and Bose-Chaudhuri Hochquenghem (BCH) error control code [5].
The choice of error control code is important as more the number of redundancy bits, more the security; however introducing too many redundancy bits will pose transmission and storage problems due to increase in file size. The BCH class of codes can perform Adaptive Protection, i.e., the redundancy is added as per the current channel characteristics [6].

The performance of BCH code for images with different values of n (code length) and t (number of errors to be corrected), for an SNR value of 9 dB has been studied in [6]. The results show that the PSNR values are high for higher values of n and t.

The authors also studied the correlation between the original and recovered signatures when the (a) image was not tampered and (b) image was not tampered. In the former case, a prominent correlation peak was obtained whereas the latter did not produce any peak. This helps to authenticate the image received.

This technique of encryption (using digital signatures) has however been proved to be weak in Ref. [12] where the authors show that a brute force attack can easily break the encrypted image.

D. Image Encryption Using Hash Functions

A hash function, H, produces a fixed-length hash value, h from a variable-length message M [11]. A desirable property of hash functions is that applied to a large number of input messages, all outputs should be equiprobable and random. Hash functions needed for security applications are called as cryptographic hash functions. For a cryptographic hash functions, it is computationally impossible to have: (i) two message blocks which result in the same hash value (ii) a message block map to a predetermined hash value.

One of the applications of hash functions is to check the integrity of the transmitted data. The hash value or message digest is computed for the original image or data and transmitted along with the image or data. At the receiver, the hash value is computed again from the received data and compared with the transmitted hash value. If the two hash values match, it validates the integrity of the data received else it implies that the data has been tampered with [14].

The proposed technique uses the security hash algorithm, SHA-512 [8]. It takes a message block of length \( \leq 2^{128} \) bits and produces a hash value of length 512 bits. The input message is processed in blocks of 1024 bits. SHA-512 can be broadly seen as composed of two processes – (i) preprocessing: padding the input message to make the length an integer multiple of 1024 bits, initializing the hash buffers and splitting the padded data into 1024 bit blocks (ii) hash computation: using the various constants and functions, repeated n number of times.

The authors follow the following steps:
1. Divide the image into 4 parts, Se.1, Se.2, Se.3 and Se.4.
2. Substitute sub-part Se.x according to the S-box of AES.
3. Calculate the mean of all rows of Se.x, which provides a [128 x 1] matrix. This matrix is concatenated horizontally to get a matrix, MRC [128 x 128].
4. Calculate the mean of all columns of Se.x, which provides a [1 x 128] matrix. This matrix is concatenated vertically to get a matrix, M [128 x 256].

Hash of Se.x and Se.y is calculated: Se.x and Se.y form a [128 x 256] matrix. M. Each row of M is divided into 124-byte long sub-arrays, appended with four 8-bit keys. Each sub-array gives a 64-bit hash value. Thus the hash function on M produces a mask of [128 x 256].

5. The new sub-parts of the images are calculated.

6. This gives us encrypted sub parts. These encrypted subparts are further manipulated as follows:

The histograms of the encrypted image and the original image are different from each other; while the histogram of the original image is non-uniform, the histogram of the encrypted image is nearly uniform. Correlation between adjacent pixels is also reduced (-0.0006 and -0.0030 for horizontal and vertically adjacent pixels respectively in the Lena image [8]). Hence cryptanalysis is made difficult.

E. Pixel Sieve Method of Image Encryption

This technique is inline with the technique suggested by Moni Naor and Adi Shamir [2]. The image is split into shares and only the knowledge of all shares will reveal the original image [2], [15], [17]. The authors have used the Sieve-Division-Shuffling (SDS) method [1].

Sieveing: this process filters the image into the Red, Green and Blue components. Hence, for a 24-bit image, 8 bits are assigned to each of the components.

Division: the filtered RGB components are divided into n shares. Thus, R \( \rightarrow (R_1, R_2, ... R_n) \), G \( \rightarrow (G_1, G_2, ... G_n) \), B \( \rightarrow (B_1, B_2, ... B_n) \). Each element in each share is assigned a random value from \([0, 2^x -1]\), where x is the number of bits used for a colour.

Shuffling: Each element in each of the shares is shuffled, with the LSBs of the next share deciding how a share should be shuffled. For example, elements of R_2 will decide how elements of R_1 will be shuffled; elements of R_3 will decide how elements of R_2 will be shuffled and so on.
The shares are combined to create the encrypted image. One of the shares can be sent on a secured channel while the other shares can be sent on an unsecure channel. Figure 9 shows the SDS method of image encryption.

Some of the advantages of this method are: (a) there is no image expansion due to redundancy bits and thus there is no additional demand for storage space (b) there are no keys used and hence there is no key management issue. (c) the technique can withstand brute force attacks. (d) the cross correlation between the original and the recovered image is 1, in other words, there is no loss in image quality. The Normalized Correlation, NC between the original and the recovered image can be calculated as [1]:

\[ NC = \frac{\sum_{i=1}^{w} \sum_{j=1}^{h} (S_{ij} \oplus R_{ij})}{w \times h} \]

S represents the secret image, R is the recovered image, w and h are the width and height of the image respectively.

### III. COMPARISON OF ENCRYPTION TECHNIQUES

The following table gives a concise comparison of the reviewed encryption methods:
The Pixel Sieve method provides a NC value of 1, which is the maximum value possible. This shows that the recovered and the original image are identical and there is no loss in quality.

In this paper, the performance of various image encryption methods in terms of their response to noise, ability to decrypt without losses, correlation between pixels in the encrypted image, key space available, compression (and hence channel BW available and storage demands), information entropy, cryptanalysis etc. has been reviewed. A given method may not show optimum performance in all these parameters and hence an encryption method must be chosen on a case-to-case basis, as per the requirements of the given system.

REFERENCES

https://www.tutorialspoint.com/cryptography/cryptography_hash_functions.htm