International Journal of Electronics, Electrical and Computational System
IJEECS
ISSN 2348-117X
Volume 6, Issue 6
June 2017

Academic Science

Implementation of Share Based Image Encryption Method

Fiona Rozario
Dept. of Electronic & Telecommunication, Dr. D. Y. Patil School of Engineering & Technology, Pune, India

Dr. Mukund G. Wani
Dept. of Electronic & Telecommunication, Dr. D. Y. Patil School of Engineering & Technology, Pune, India

ABSTRACT

Encryption being an integral part of any communication today, it is important that the encryption algorithm be simple, convenient and yet effective without demanding overheads of storage and large computational times. This paper presents the results of implementing the share-based method of image encryption, which fulfills the said requirements.

Keywords: ***Image, Encryption, Image encryption, Pixel Sieve.***

I. INTRODUCTION

Cryptography is "the study of mathematical techniques related to aspects of information security such as confidentiality, data integrity, entity authentication, and data origin authentication. [1]"

Encryption refers to the process of altering the data, which is meant to be communicated (plaintext), into a form, which cannot be understood by anyone without authorized access to the data. This coded data is called ciphertext and a key, which encrypts and decrypts the data, provides the authorization.

Encryption algorithms are of two types: (1) symmetric and (2) asymmetric. Symmetric algorithms use the same key to encrypt and decrypt data. The challenge of symmetric algorithms is communicating the key to the receiver end since the confidentiality of the key is critical to the security of the data. Asymmetric algorithms use two mathematically linked keys – public and private. Multiple users can share public keys while a single user uses the private key.

Encryption may be applied to text and images albeit differently. Image encryption is different from textual encryption since an image is nothing but a matrix or a block of pixel values, whereas textual data is a one-dimensional stream of characters. Being a two-dimensional matrix, image data is also heavier than textual data. Also, losses in textual encryption cannot be tolerated whereas losses in image encryption can be tolerated within limits [2].

Image encryption techniques may be classified as (i) key dependent techniques like modified AES for images, vector quantization, chaos based algorithms and others and (ii) keyless techniques like the share-based algorithm proposed by Moni Naor and Adi Shamir.

While the key dependent schemes face the challenges of efficient key management, limited key space and high computational times, keyless schemes are simpler to implement without any requirement of key management and yet effective in their basic purpose of securing the data.

Academic Science

Moni Naor and Adi Shamir proposed one such method of securing data without the use of encryption keys. They proposed dividing the data to be shared into n parts. To be able to decrypt or reconstruct the data, the receiver must have the knowledge of at least k parts out of the n parts; else the decryption is not possible [8].

Section II of this paper reviews the related work in the field of image encryption. Section III presents the proposed technique of share based algorithm, namely the Sieve, Shuffle, Divide & Shuffle technique, which is based on the technique proposed in [8] and [9]. Section IV presents the experimental results and Section V concludes the paper.

II. REVIEW OF RELATED WORK

The authors in [2] have designed an algorithm to decompose the image into a set of vectors and then quantize the set of vectors as per a designed codebook, such as to achieve least distortion between the quantized and the actual value. To transmit, either the codebook indices are encrypted while codebook is sent as plaintext or the codebook is encrypted and indices are sent as plaintext. While the advantage of this method is compression in the file size, its disadvantage is that the design of the codebook for quantization is computationally complex.

In [3], the authors propose to calculate the digital signature of the image using any one of the standard algorithms and add it to the encoded image. A digital signature is a bit pattern that depends on the message and uses some information unique to the sender [4]. Some of the standard digital signature algorithms are MD2, MD4, MD5 and SHA. This method was however proven to be weak in [5] as a brute force attack can break the encrypted image.

Hash based algorithm in [6], proposes to calculate a fixed length hash value, H from a variable length message [7]. Hash functions, when applied to a large number of input messages, produce outputs that are all equiprobable and random. The proposed method produces a uniform histogram and low correlations between encrypted and original images, however needs to enforce some encryption key management scheme.

The authors in [8] refer to the scheme proposed by them as the ‘threshold scheme’. To share a secret data, the data must be divided into a total of n parts. For anyone to be able to retrieve the data in totality, knowledge of at least k parts is a must. Knowledge of any less than k parts renders the decryption incomplete. This is the (k, n) threshold scheme.

The threshold scheme is best suited for data which is sensitive and when the application environment involves individuals with conflicting interests. The k parts of the data are saved with k individuals each. Thus to retrieve the data, a user will require the parts saved with the other users as well. Besides securing data, the threshold scheme may also be used in managing the cryptographic keys of an encryption system.

In [9], the authors have proposed a (z, z) threshold scheme where an image is divided into z parts and all z parts are required to recover the original image. The algorithm is called the SDS algorithm where SDS stands for Sieve Division Shuffling.

The image is first sieved into the RGB components; each component is then randomly divided into z parts or shares (R_1, R_2…R_z; G_1, G_2…G_z; B_1, B_2…B_z) [9]. Each of these shares is then shuffled within itself and

International Journal of Electronics, Electrical and Computational System
IJEECS
ISSN 2348-117X
Volume 6, Issue 6
June 2017

lastly, the shuffled shares are combined. The size of the shares is kept constant which is equal to the size of the secret image. Information regarding the encryption scheme is unavailable in [9].

III. PROPOSED TECHNIQUE

Since the application of image encryption is intended for use on computer/TV screens, the colour model selected is the additive colour model. The additive colour model mixes red, green and blue in varying proportions to produce the different colours [10]. For example, green and red mixed together produce yellow. In contrast, the subtractive colour model subtracts certain colours to create other colours. It is the subtractive model that helps us in seeing different colours on objects around us. The print media uses this colour model. The proposed technique is Sieve, Shuffle, Divide & Shuffle:

(i) the plaintext image is sieved into its RGB components

(ii) the individual R, G and B components are shuffled

(iii) three shares are created from each of the components (R_1, R_2, R_3, G_1, G_2, G_3, B_1, B_2 and B_3)

(iv) each of the shares is encrypted i.e. the pixel value in each share is encrypted

(v) each share is shuffled after encryption

(vi) R_1, G_1, B_1 are concatenated to form the first encrypted image share, R_2, G_2, B_2 are concatenated to form the second encrypted image share and R_3, G_3, B_3 form the third encrypted image share.

There are two shuffling algorithms used. The shuffling algorithm used to shuffle the RGB components in step (ii) is in synch with the shuffling algorithm used in [9] i.e., the current component or share is shuffled on the basis of whether the corresponding pixel value in the next component share is odd or even.

The shuffling algorithm used in step (v) is based on the modulus operator. New row (x value) and column (y value) indices are calculated for each pixel from the following:

$$newx = (x + increment) \bmod 256 \quad (1)$$

$$newy = (y + increment) \bmod 256 \quad (2)$$

The value is being reduced to 256 using modulus since $2^8 = 256$ is the maximum number of values that are possible in each of the R, G and B components in a 24-bit image.

The increment is the first prime number within a chosen range of the height and width of the image. Thus if the size of the image is 20 x 10 pixels, the prime number for x will be the first prime number in the range of [20 to (20+20)] and increment for y will be the first prime number in the range of [10 to (10+10)]. The range i.e. the difference between the upper and lower bounds (20-10 and 40-20) can be altered as per the user or kept constant. However, the prime number selected for the increment will definitely change if the dimensions of the image change.

Value at shuffled_share (newx, newy) = original (x, y)

The encryption algorithm in step (iv) also uses the modulus operator. The pixel value, b at (i, j) is replaced by its modulo inverse, b-1 given in [11] as:

$$bb^{-1} \equiv 1 \bmod m \qquad (3)$$

The chosen value for m is 256 since $2^8 = 256$ is the maximum number of values that are possible. This has been derived from the basic linear congruence equation:

$$ax \equiv b \bmod m \qquad (4)$$

If d is the gcd of a and m, (4) has solutions only if d divides b. Equation (4) has a unique solution iff gcd (a, m) is 1 i.e. d is 1[12].

IV. RESULTS

Most digital cameras today support a minimum pixel depth of 24 bits. Hence, the proposed technique has been implemented on 24 bit pixel depth images (jpg and png file formats). The pixel depth gives the number of pixels used to define each of the R, G and B pixels. In case of a 24-bit pixel depth, 8 bits are used for R, G and B components each. The results and figures below are for a jpg image of dimensions 100 x 100 although the scheme has been tested for images of various dimensions as well.

The number of shares created is three. Unlike in [9], the size of the shares is not constant and is a function of the original size of the image.

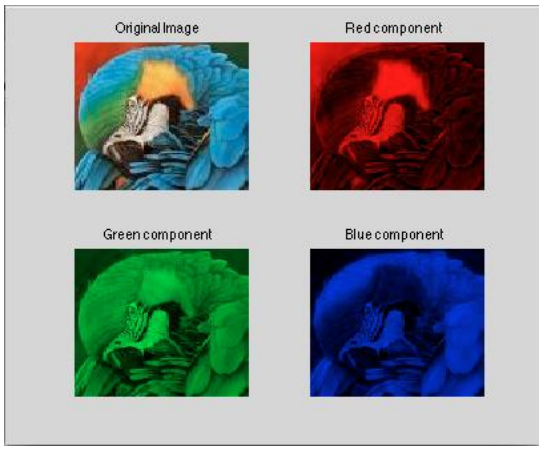

Figure 1: RGB components of the original image

Figure 1 shows the RGB components of the original image after sieving and figure 2 shows the shuffled RGB components.

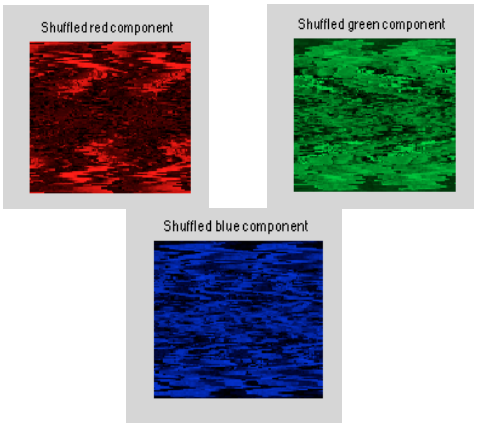

Figure 2: Shuffled R, G and B components

Figure 3 is the result after steps (iii), (iv) and (v). Figure 4 shows the three encrypted images (or shares of the original image) that are to be transmitted to the receiver. First image share is created from encrypted and shuffled R_1, G_1, B_1; second from encrypted and shuffled R_2, G_2, B_2 and third from encrypted and shuffled R_3, G_3, B_3. For decryption at the receiver end, three images are selected and the decryption algorithm followed.

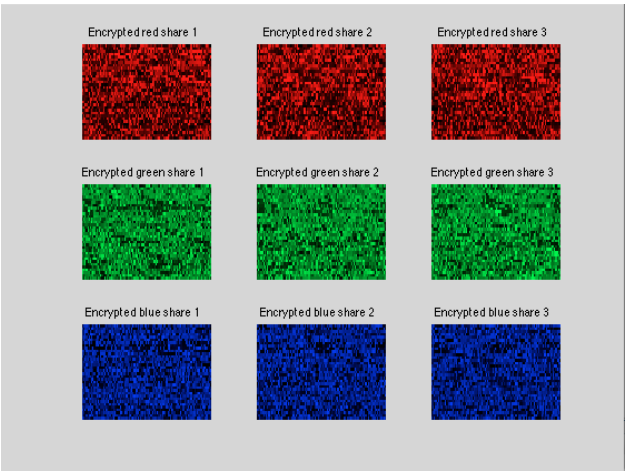

Figure 3: Encrypted shares of the RGB components

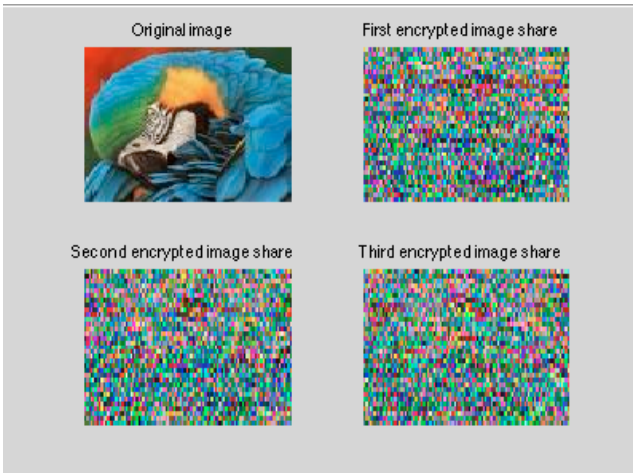

Figure 4: Encrypted image shares for transmission

The encrypted shares have been evaluated on the metrics of correlation with the plaintext shares, entropy and Peak Signal to Noise Ratio (PSNR).

Correlation gives an idea about the relationship between two entities. It calculates the degree of similarity between the two entities. A good cryptosystem should hide all attributes of the original image and randomize the pixel values in the encrypted image. This would give a very low, close to zero correlation coefficient. A value of 1 implies that the two images are in perfect relation or in other words, are identical [13].

The correlation coefficients obtained for the three encrypted image shares are -0.0202, 0.0228 and 0.0329, which shows that the encrypted images bear very little resemblance with the original image shares.

Information entropy is another parameter that evaluates a cryptographic system. Information entropy gives a measure of the uncertainty in the communication system, or in our case, the uncertainty produced by the cryptographic system [13]. For a system that uses 8 bits for a symbol, the ideal value for entropy is 8, which implies that the system is perfectly random in producing symbols. However, truly random systems do not exist. Hence the entropy should be close to the ideal value. Entropy is calculated as:

$$H(m) = \sum_{i=0}^{2^N-1} p(m_i) \times \log_2 \frac{1}{p(m_i)} \quad (5)$$

where $p(m_i)$ denotes the probability of the i^{th} message using N bits. The entropy calculated for the proposed scheme for the three image shares are 7.3959, 7.4335 and 7.3839.

PSNR indicates the change in pixel values between two images (plaintext and cipher image or plaintext and deciphered image) and is calculated from the Mean Square Error (MSE). MSE measures the average of the squares of deviations between two images and is given by:

$$MSE = \frac{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (P(i,j) - C(i,j))^2}{M \times N} \quad (6)$$

where M and N are the width and height of the image respectively, P(i,j) is the pixel value of the plaintext image at (i,j) and C(i,j) is the pixel value of the ciphertext image at (i,j). Thus, MSE between the cipher image

International Journal of Electronics, Electrical and Computational System
IJEECS
ISSN 2348-117X
Volume 6, Issue 6
June 2017

and plaintext image should be large, whereas it should tend to zero for deciphered image and plaintext image. Mathematically, PSNR is given by:

$$\text{PSNR} = 10 \log_{10} \left[\frac{\text{L}^2}{\text{MSE}}\right] \quad (7)$$

where L is the maximum value a pixel can take. In case of a 24-bit depth image, L is 2^8-1 = 255. From (7), for a plaintext image and cipher image, since MSE should be large, PSNR should be low, indicating good encryption quality. On the other hand, between a deciphered image and plaintext image, since MSE tends to 0, PSNR tends to infinity. Excellent recovery of image in wireless transmission requires PSNR of 30 to 50 dB [14]. The values of PSNR obtained for the three ciphered shares are 12.0311, 12.0176, 12.1151.

To be able to decrypt the images, all three shares of the image are required. Hence, breaking the data into shares and distributing the shares among more than one person protects the sensitivity of the data.

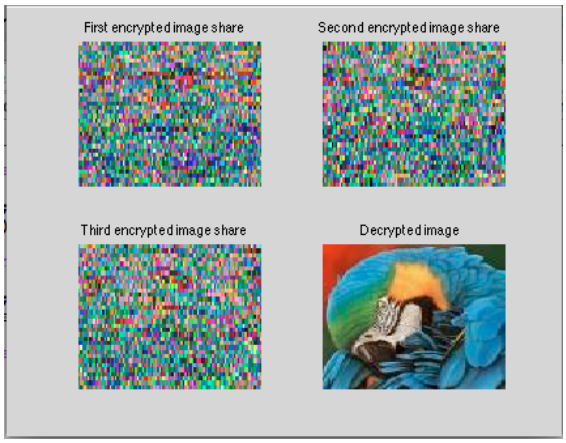

Figure 5: Decryption from image shares

The scheme does not require any cryptographic key and hence is free from key management issues. Also, there is no loss of data due to any quantization or increase in file size from addition of redundant bits. Hence, this scheme is suited for applications with bandwidth and storage constraints.

Figure 5 shows the decrypted image from the individual shares. The correlation between the decrypted and original image is 1; hence the decrypted image is lossless.

The results for three other test images are as tabulated below:

Figure 6: rainbow.jpeg

Figure 7: vegetable.png

Figure 8: smoke.png

TABLE I: RESULTS

Image	Dimension	Correlation	Entropy	PSNR
rainbow.jpeg	379 x 113	0.0451 0.0247 0.0357	7.4672 7.4748 7.4817	11.5897 11.4604 11.5193
smoke.png	180 x 180	-0.0415 -0.0194 -0.0218	7.2725 7.2971 7.2327	9.8526 9.9686 9.9601
vegetable.png	259 x 194	0.0330 0.0291 0.0101	7.6562 7.6633 7.6649	10.3195 10.3039 10.2460

V. CONCLUSION

The image encryption scheme proposed is based on the idea of dividing the data into smaller 'shares' and encrypting and transmitting these shares individually as the shares cannot provide any information 'stand alone'.

Following are the merits of the proposed scheme:

(i) there is no expansion in file size which suits applications with bandwidth and storage constraints.

(ii) there are no key management issues since it uses no keys,

(iii) the encryption and shuffling scheme use modulus operation, which produce results in a finite field while following algebraic properties

(iv) sensitive data is protected by sharing different parts of the secret data among multiple users. Thus all shares will need to be hacked into to access the data.

(v) it provides low values of correlation and PSNR between cipher and plaintext image shares and high entropy values.

The scheme is suitable for environments where multiple users have access to sensitive information, for e.g. the military and crime branch or critical bank account picture passwords. The picture password may be sent to the different authorized users as shares, each share transmitted via a different means. Thus, if the communication link is hacked into while transmitting one of the secret shares, the picture password will not be accessible without all shares combined.

REFERENCES

[1] Menezes, P. Van Oorschot, and S. Vanstone, "Chapter 1 - Overview of Cryptography," in Handbook of Applied Cryptography.: CRC Press, Inc., 1997.

[2] Chin-Chen Chang, Min-Shian Hwang, and Tung-Shou Chen, "A new encryption algorithm for image crytosystems," The Journal of Systems and Software.

[3] Aloka Sinha and Kehar Singh, "A technique for image encryption using digital signature," Optics Communications, 2003.

[4] William Stallings, Cryptography and Network Security - Principles and Practice, 5th ed.: Prentice Hall, 2006, Chapter 13

[5] L. Hernandez Encinas and A. Peinado Domınguez, "Comment on 'A technique for image encryption using digital signature'," Optics Communications, vol. 268, no. 2, pp. 261-265, December 2006.

[6] Seyed Mohammad Seyedzade, Reza Ebrahimi Atani, and Sattar Mirzakuchaki, "A Novel Image Encryption Algorithm Based on Hash Function," 2010.

[7] William Stallings, Cryptography and Network Security - Principles and Practice, 5th ed.: Prentice Hall, 2006, Chapter 11.

[8] R. Rivest and Adi Shamir, "How to Share a Secret," Communications of the ACM, vol. 22, no. 11, November 1979.

[9] Siddharth Malik, Anjali Sardana, and Jaya, "A Keyless Approach to Image Encryption," in International Conference on Communication Systems and Network Technologies, 2012.

[10] Rafael C. Gonzalez and Richard E. Woods, Digital Image Processing. Indian Subcontinent version: Pearson, 2014.

[11] Eric Weisstein. Wolfram Mathworld. [Online]. "http://mathworld.wolfram.com/ModularInverse.html" http://mathworld.wolfram.com/ModularInverse.html

[12] Victor Adamchik. Carnegie Mellon University School of Computer Science. [Online]. "https://www.cs.cmu.edu/~adamchik/21-127/lectures/congruences_print.pdf" https://www.cs.cmu.edu/~adamchik/21-127/lectures/congruences_print.pdf

[13] Ahmad, Jawad Ahmed and Fawad, "Efficiency Analysis and Security Evaluation of Image Encryption Schemes," International Journal of Video & Image Processing and Network Security, vol. 12, no. 4, August 2012.

[14] Emanuele Colucci. (2011, April) emanuelecolucci.com. [Online]. "http://emanuelecolucci.com/2011/04/image-and-video-quality-assessment-part-one-mse-psnr/" http://emanuelecolucci.com/2011/04/image-and-video-quality-assessment-part-one-mse-psnr/