Int. J. Electron. Commun. (AEÜ) 69 (2015) 562–572

Contents lists available at ScienceDirect

International Journal of Electronics and Communications (AEÜ)

journal homepage: www.elsevier.com/locate/aeue

Pixel scattering matrix formalism for image encryption—A key scheduled substitution and diffusion approach

Padmapriya Praveenkumar, Rengarajan Amirtharajan*, K. Thenmozhi, John Bosco Balaguru Rayappan

School of Electrical & Electronics Engineering, SASTRA University, Thanjavur 613 401, India

ARTICLE INFO

Article history:
Received 23 May 2014
Accepted 14 November 2014

Keywords:
Image encryption
Pseudorandom sequence
Discrete cosine transform (DCT)
Subspace key
Substitution and diffusion

ABSTRACT

The significance of cipher communication against hackers has been strongly felt by the cyber lawmakers. In this direction a yet another double-edged knife cipher methodology employing image encryption (first edge) and multilayered cryptic key (second edge) has been proposed. In this paper, a rapid key encryption procedure employing symmetric key adopting matrix array was practiced. A complex and multilayered key generation scheme employing pseudorandom sequence, DCT, quantization and scrambling was adapted. The operations like diffusion and substitutions were inherently inducted in the proposed scheme to provide faster convergence of cipher image. The effectiveness of the proposed scheme was verified by performing security analysis and also metrics evaluation. Further the present results were compared with the available results and the merits of the same have been highlighted.

1. Introduction

Immense research has been going on with cryptography as the base. It is not something new; it is intelligent communication without others getting to know it. Also steganography [1] and watermarking have been active in their contribution for secure communication. The first term is just masking the digital data within spatial or transform domain coefficient and the second is associated with copyright and endorsement. Encryption has further become popular as it does not give a single data and differs from normal image as it has redundancy and bulk data capacities as well as supporting all types of digital files.

Shreyamsha Kumar and Patil [2] have presented Fuzzy-based PN sequence to modify DCT blocks and use chaotic-based encryption to provide the coded bits. Youssef et al. [3] have studied image encryption by combining pseudorandom code generators with gold code generators. An image encryption [4] presents a scheme in which the digital encryption was achieved via Logistic chaos on RGB encrypted with improved DES twice. Yoon and Kim [5] proposed chaotic maps which are used to generate permutation in a pseudorandom manner. To retain the quality of the image during the encryption and decryption processes DCT-based image models are developed [6–8].

Huang et al. [9] have proposed a time-delay Lorentz chaotic system to provide diffused image encryption scheme. A technique with encrypted steganography was presented by Diaconu and Loukhaoukha [10]. Riad and Hussein [11] have combined Arnold's cat map with modified IDEA techniques to reduce the encryption time as compared to the conventional scheme. Feng and Yu have proposed a 2D image encryption where the keylength is not restricted and provides no information loss [12]. Wang and Jiang proposed an algorithm which is based upon hyperchaotic map, image shuffling and image mixing techniques [13]. Loukhaoukha et al. have presented Rubik's cube algorithm in which two secret keys in rowwise and columnwise are used to encrypt the image with the help of XOR operation [14]. Naik and Pal have proposed a cryptosystem in which the secret image data set is shuffled using the Arnold's cat map and later diffused [15].

In order to resist against attacks, a large key space is required. The algorithms like new chaotic-based shuffling algorithm [16], Ergodic matrix and hybrid key [17] and a new method of selective encryption for JPEG images [18] are used. Paul et al. [19] have presented a complex key generation procedure based on matrix manipulations that could be introduced in symmetric ciphers. Younes and Jantan have presented a key-based encryption utilizing Blowfish and transformation process [20]. Alam and Khan have analyzed the performance and efficiency of different block cipher

* Corresponding author. Tel.: +91 4362 264101–108; fax: +91 4362 264120.
E-mail addresses: padmapriya@ece.sastra.edu (P. Praveenkumar), amir@ece.sastra.edu (R. Amirtharajan), thenmozhik@ece.sastra.edu (K. Thenmozhi), rjbosco@ece.sastra.edu (J.B.B. Rayappan).

http://dx.doi.org/10.1016/j.aeue.2014.11.010
1434-8411/

algorithms, their encryption time, decryption time, throughput and power consumption [21].

Yang and Boussakta have presented a symmetric cryptographic system founded on the concept of Shannon where the S box is replaced by number theoretic transform (NTT) which results in diffusion [22]. Ramesh and Umarani have presented a series of transformations that are in use based on the S-BOX, XOR Gate, and AND Gate to encrypt the message [23]. A technique to show that chosen cipher text attacks of the wavelet encryption system can reduce the problem of solving a set of nonlinear equations over finite fields has been presented in [24]. Anoop and Alakkaran [25] have combined DWT and Stream Ciphers on image encryption schemes in transform domains and the combination of DWT and block cipher for image encryption has been proposed by Dang and Chau [26].

After reviewing the available literature, this paper focuses on symmetric key-based image encryption to provide faster data accessing in today's multimedia world. Then PN sequence was generated to provide the key and to use for manipulating the matrix. This method ensures high security as the key was generated using PN sequence generator and DCT-based scrambling. The proposed encryption algorithm encrypts a 128 bit block and a 128 bit key at once since it is a block cipher. Section 2 provides the proposed methodology and Section 3 presents the results and discussion part and finally Section 4 provides the conclusion part.

2. Proposed methodology

The proposed algorithm is given in Fig. 1 and the algorithm for encryption and decryption is given in Section 2.1. Matrix generation and key scheduling schemes were employed which make use of pseudorandom sequence, DCT, quantization and scrambling. Diffusion and substitution operations were performed finally to provide the cipher image.

2.1. Proposed algorithm

Step 1: Secret key generation (K)
- 1.1) Get the initial value of shift register for PN sequence generator
- 1.2) Generate the PN sequences.
- 1.3) Convert the values into decimal to get K(i), where, K is the secret key and I represents the number of iterations.
- 1.4) Repeat the above steps for 16 iterations.

Step 2: Matrix generation (M).
- 2.1) Initialize a matrix of size 16×256.
- 2.2) Element of matrix depends on the secret key of size 16.
- 2.3) $M(i, j) = integervalueofK(i) + j.$ (1)
  where K is the secret key, i, j are the rows and columns of the matrix. M is the generated matrix.

Step 3: Sub-key generation. (SK1 and SK2)
- 3.1) Initialize sub-keys of size 16×16.
- 3.2) $A1 = K^T$.
- 3.3) A2 = A1 xor K
  (A2 left half) = L,
  (A2 right half) = R
- 3.4) $A3 = L \oplus R$
- 3.5) $A4 = L^T \oplus R^T$.
- 3.6) A5 = concatenate A3 and A4.
- 3.7) S = sum of all the elements of A5.
- 3.8) A1 = S % 23.
- 3.9) A2 = S % 15.
- 3.10) SK1 (i, j) = (i,(A1 + A2 + j))
  SK2 (i, j) = (i,(KS1(i, j)))
- 3.11) $Q = (K(i)\%12) + AS1andR = (K(i)\%10) + AS2$ (2)
- 3.12) Rotate ith column of SS1 by Q times and SS2 by R times.
- 3.13) Divide SS1 and SS2 into 4×4 blocks.
- 3.14) Compute DCT coefficients for each block and apply quantization and find the difference between the adjacent blocks of the quantized value. Then Scramble each block.
- 3.15) recombine all the block to get SK1 and SK2.

where, i, j are the rows and columns of SK1 and SK2.
SK1 is the subspace key 1, SK2 is the subspace key 2.

Step 4: Iterations for substitution and diffusion
- 4.1) Get the input image of size 256×256 and
- 4.2) Divide them in to 4×4 blocks.
- 4.3) For a block iterate the steps for 16 rounds.
  - 4.3.1) Get the image block of size 16 bytes.
  - 4.3.2) C (i) =M (i, (p(i))). I varies from 1 to 16
    P = image block.
  - 4.3.3) C1 = C XOR SK1 (i, j).

where, i, j are the rows and column of subspace key 1.
  - 4.3.4) $C2 = C1^T$.
  - 4.3.5) left half of C2 = L1
  - 4.3.6) right half of C2 = R1
  - 4.3.5) $C3 = L1 \oplus R1$.
  - 4.3.6) $C4 = L1^T \oplus C3$.
  - 4.3.7) C5 = concatenate C3 and C4.
  - 4.3.8) $C6 = C5^T$.
  - 4.3.9) C7 = C6 XOR Sk2 (i, j).

where, i, j are the rows and columns.
  - 4.3.10) C8 = Left half of C7
  - 4.3.11) C9 = Right half of C7.
  - 4.3.12) C10 = Sum of elements of C8.
  - 4.3.13) C11 = C10 MOD 6.
  - 4.3.14) C9 = Rotate right C9 by C11 times.
  - 4.3.15) C12 = sum of elements of C9.
  - 4.3.16) $C13 = C12\%6.$ (3)
  - 4.3.17) C8 = Rotate right C8 by C13 times.
  - 4.3.18) Cipher text = concatenate C9 and C8.
- 4.4) Repeat 4.3 for all the blocks.
- 4.5) Recombine all the blocks to get cipher image

Step 5: Decryption is the reverse of encryption.

3. Simulation results and discussions

The proposed key-based encryption algorithm was simulated using Matlab R 2012 a. The original Lena image of size 256×256 is given in Fig. 2a and its histogram in Fig. 2b. The resultant encrypted image and its histogram after the first round is given in Fig. 3a and b, respectively. Fig. 4a and b provides the encrypted image after second round and its corresponding histogram, respectively. Fig. 5a and b shows the final encrypted image and its corresponding histogram after round 16, respectively, and Table 1 provides the metrics of various images.The original West concord orthophoto image of size 256×256 is given in Fig. 6a and its histogram in Fig. 7a. The resultant encrypted image and its histogram after the first round is given in Figs. 6b and 7b, respectively. Figs. 6c and 7c provide the encrypted image after second round and its corresponding histogram, respectively. Figs. 6d and 7d show the final encrypted image and its corresponding histogram after round 16, respectively.

The original Cameraman image of size 256×256 is given in Fig. 8a and its histogram in Fig. 9a. The resultant encrypted image and its histogram after the first round is given in Figs. 8b and 9b, respectively. Figs. 8c and 9c provide the encrypted image after second round and its corresponding histogram, respectively. Figs. 8d and 9d show the final encrypted image and its corresponding histogram after round 16, respectively.

The original Peppers image of size 256×256 is given in Fig. 10a and its histogram in Fig. 11a. The resultant encrypted image and its histogram after the first round is given in Figs. 10b and 11b, respectively. Figs. 10c and 11c provide the encrypted image after second round and its corresponding histogram, respectively. Figs. 10d and 11d show the final encrypted image and its corresponding histogram after round 16, respectively.