ISSN:0975-9646

A Symmetric-Key Latin Square Image Cipher with Probabilistic Encryption for Grayscale and Color Images

Tanvi Nema
Student, M.Tech
CSE, VNSIT Bhopal

Prof. Amit nandanwar
Computer *Science and*
Engineering, VNSIT Bhopal

ABSTRACT
In this paper Considerations of privacy and confidentiality in a computer environment have given recognition to the need for protecting certain communications and stored data from theft and misuse. A suitable methodology for protecting communicated or stored data involves the use of cryptographic techniques. Cryptography is the study of mathematical techniques related to aspects of information security such as confidentiality, data integrity, and entity authentication and data origin authentication. A message is plaintext. The process of disguising a message in such a way to hide its substance is encryption. An encrypted message is cipher text. The process of turning cipher text back into plain text is decryption. Basic operations that can be carried out in encryption/decryption are: substitution and transposition. Due to advent of computers, these operations are carried out on binary bits.
The field of encryption is becoming very important in the present era in which information security is of utmost concern. Security is an important issue in communication and storage of images, and encryption is one of the ways to ensure security. Image encryption has applications in internet communication, multimedia systems, medical imaging, telemedicine, military communication, etc.
Keywords Cryptography, Authentication, Information, Security, Decryption

1 INTRODUCTION

In this paper Information that can be read and implicit without any special procedures or method is termed as plaintext or clear text. The technique of concealing plaintext in order to hide its particular material is called encryption. The impression of encryption is to make a message incomprehensible, except to the receiver.
Data encryption technology is used to benefit protection against loss, exploitation or alteration of private information. Encrypting plaintext results in indecipherable rubbish called cipher text. Encryption is used to guarantee the hidden information from anyone of concern not intended to, even those who can comprehend the encrypted data. The procedure of backsliding cipher text to its original plaintext is considered as decryption.

2 IMAGE ENCRYPTION TECHNIQUES

2.1 Classic Image Encryption

an improved AES based algorithm by including a key stream generator (A5/1, W7) to AES to guarantee enhancing the encryption execution for image encryption process. An alternate algorithm proposed by Subramanyan et al. [6] focused around AES Key Expansion in which the encryption methodology is a bit astute XOR operation of a set of image pixels besides a 128 bit key that varies for each set of pixels. The keys to be utilized are produced freely at the sender and recipient side focused around AES Key extension transform thus the preliminary key is distant from everyone else imparted instead of offering the entire set of keys. DES, a prevalent block cipher algorithm utilizes 64 bit key, which is an alternate printed cryptosystem that utilized for image encryption by Qian Gong-canister et al. In [7] another image encryption plan focused around DES consolidated with a chaotic map introduced to enhance the security and develop the key space. The results demonstrate that blend of word-based cryptosystems with different strategies or rolling out a few improvements, enhance the security and against anti attack capacity of those algorithms adequately.

2.2 Public Key Image Encryption

Most of application does not provide a facility of a secure channel to transfer the private key or desire to keep the decryption key in secret, so we need to utilize public key cryptography. In the first place public key was circulated by Diffie and Hellman [8]. It was a key trade down to earth strategy for making an imparted secret key over a verified correspondence channel without utilizing a former imparted secret. The greater part of conventional public key cryptosystems intended to encode printed information. A few works have been distributed on public key image encryption, one is proposed by Shuihua et al. [9].
In this plan, the plain image isolated into blocks utilizing a certain network change and all pixels in each one block exchanged to DCT field. Public key, private key, encryption methodology and unscrambling procedure are characterized focused around change network of DCT coefficients. The results show that this system is vigorous in contradiction of JPEG lossy clamping and other general assaults. An alternate public key system focused around Chebyshev chaos map portrayed for colour images encryption and features progressively applications. In the first place they attempted to cryptanalysis the encryption focused around Chebyshev polynomial map and results demonstrate that it is not powerful on a few attacks, so they attempted to improve the security by utilizing a non-Xoring hash function to secure it against attack of picked plaintext. They do proficiency check and some testing for cryptanalysis, for example, key affectability, connection, mono bit, long run test and time examination for both image and video and determined from the result that their

recommended cryptosystem is more secure and strong to any invader attack and the time investigation exhibits the effectiveness of encryption for 64x64 and 128x128 video encryption.

2.3 Compression and Encryption

Compression procedures help us to lessen the transmission data transfer capacity or storage space. These procedures can be actualized in both spatial and frequency domain. Also frequency domain procedures are further effectual and consuming collective and widespread transforms such as DCT, DFT and DWT. Data compression lessons can be classified into two types:

• LOSSY: Lossy methods compromise a definite loss for information in return with the high compression proportion. Usually lossy methods decline the superiority of the object so they are sought out for images, videos and audios for the reason of human observation. There Lossy coding method also moreover categorised into the following types:

a. Predictive coding
b. Transform coding

• LOSSLESS: On the other hand, some kinds of data could not accept any loss (e.g. Database records, executable files and word processing files and medical images), otherwise the data will be degraded, and here the lossless techniques play role. The Lossless coding technique also further classified into following categories:

a. Run length encoding
b. Huffman encoding
c. Arithmetic encoding
d. Entropy coding
e. Area coding

Ordinary cryptosystems identifies with the compressed multimedia. Encryption and compressed multimedia are typically extremely contradictory and an exchange off depends between them. Encrypting the interactive media content before pressure uproots a ton of repetition and this result in an exceptionally poor compression proportion. Then again, encrypting the information after compression demolishes the codec designs, which are the bases for the decoders to crash. As a final point, encryption is taken lightly for many applications to reserve approximate perceptual data [14]. B. Mohammed et al. [15] in their projected encryption-compression method initially enforced a FMT technique to compress the particular image and at that time AES-Based algorithm functionalized to encrypt the image. L. Vorwerk et al. [16] strained to syndicate encryption and wavelet compression. The methodology of encryption utilizes a symmetric key for encrypting image and wavelet filter, a public key cryptosystem is recommended to encrypt the symmetric key for secure key interchange.

A recent mixing arrangement of encryption and compression for images suggested by I. Masanori et al. [17] centred on Independent Component Analysis (ICA) and Discrete Cosine Transform (DCT). To attain a quick and secure image transmission they utilized DCT and a low pass filter for image compression and by rotating and making a mixture of the DCT blocks with an arbitrary image, the source image is encrypted. When this journey's end, the encrypted expected image is decrypted by taking out the protected images from the mixtures by spreading over ICA and lastly by utilizing rotation keys and IDCT the novel image is recreated.

Additional methodology to assimilate compression and encryption is deliberated in the system of C. Wu and J. Kuo [18]. They debated about benefits and drawbacks of discerning encryption and anticipated an encryption schemes which can transforms the entropy of simple message as an outcome to be a cipher message by spreading on Huffman coder and QM coder. Finally, concluded that this high security scheme can be applied to compression techniques such as MPEG and JPEG with acceptable computational speed.

2.4 Selective Encryption

A methodology that offered to abstain from encrypting the complete image is called selective encryption also acknowledged as partial encryption, soft encryption or perceptual encryption. The primary inspiration is to lessen the computation time for real-time applications that runtime performance is frequently serious deprived of compromising the security of the broadcast moreover. The primary objective is to divide the image content into two shares, public share and protected share. One significant feature in selective encryption is to decrease the protected part to least as it can be. Selective encryption generally accompanies compression. In frequency domain, low frequency coefficients convey most of the data of the image and high frequency coefficients convey the fine points [19].

In lossy compression techniques for example JPEG standard, an image changes to a frequency domain by DCT and at that point roughly high frequency coefficients are reproduced by zeros and new compressed image is recreated. Therefore only few low frequency coefficients can be encrypt relatively than all in frequency domain that also has many benefits [20]:

• It is less demanding to recognize the critical parts to be encrypted
• It is less demanding to recognize parts of the information are not compressible.

2.5 Chaos Theory and Cryptography

Chaos hypothesis is the investigation of nonlinear dynamical frameworks that are display compelling affectability to starting conditions and have arbitrary like practices, founded an impact which is prominently alluded to as the butterfly impact that has a definition: Does the fold of a butterfly's wings in Brazil set off a tornado in Texas? The fluttering wings speak to a little change in the starting state of the framework, which causes a bind of occasions prompting extensive scale phenomena. Had the butterfly not fluttered its wings, the trajectory of the framework may have been immensely distinctive [31].

For the most part it implies that little contrasts in beginning conditions, (for example, those because of adjusting errors in numerical calculation) yield generally separating results for chaotic systems, interpreting long-

term prediction usually intolerable. This happens despite the fact that these frameworks or systems are deterministic, implying that their future conduct is completely dictated by their starting conditions, with no arbitrary components included. At the end of the day, the deterministic nature of these frameworks or systems makes them volatile [32].

According to [33], there are two general ways to apply a chaos map in a cipher system:

• Using chaotic systems to generate pseudo-random key stream which corresponds to stream ciphers.

• Using the plaintext or the secret key(s) as the preliminary conditions and control parameters then apply some iterations on chaotic systems to obtain cipher-text corresponding to the block ciphers.

This conduct is known as deterministic chaos, or basically chaos. Irregular like conduct, non-anticipating and affectability to preliminary value are three features that make it an adequate choice to relate it with cryptography. The main distinction is that encryption operations are characterized on limited sets of numbers while chaos maps are characterized on true numbers. Chaotic behaviors are displays by chaotic maps. These maps are grouped by non-stop maps and discrete maps.

Discrete maps typically take the manifestation of iterated functions. Iterates are like rounds in cryptosystems, so discrete chaotic dynamic systems are utilized as a part of cryptography. Every map consist of parameters which are correspondent to the encryption key in cryptography.

As per [32], there are two general approaches to apply a chaos map in a cipher system:

• Chaotic systems utilization for production of pseudo-arbitrary key stream which compares to stream ciphers.

• Utilization of the plaintext or the mystery key(s) as the preliminary conditions and control parameters then apply a few cycles on chaotic systems to acquire cipher content relating to the block ciphers

2.5 Digital Signature for Image Authentication

The demonstration of digital signature is like manually written on paper signature which assuming the principle part in validating reports and confirms the individuality. Subsequently, digital signature has numerous applications in data security. It is a system that offers verification, information reliability and on-revocation. Several years ago the first idea of digital signature was a plan focused on RSA [33] and today it is a standout amongst the most functional procedures.

The greater part of the digital signatures is focused around asymmetric cryptography. In these frameworks, the private key is utilized to make a digital signature that particularly proofs the underwriter who is holder of the private key and can be verified just with the relating public key.

Digital signature and watermarking are connected and both utilized for verification and confirmation however there is somewhat contrasts in their structure [35]. The preference of this consolidation is to spare the obliged data transfer capacity for signature which is encoded to a different record. To attain this point, they install the signature record as a watermark, so their proposed plan can be able to verify the image, as well as can perform a copyright security. An alternate image verification plan [36] utilize the substance of an image wavelet change space to develop a structural digital signature. They demonstrated that this plan is forceful to content stabilizing controls and delicate to content evolving alterations.

plan focused around an arrangement of comprehensive synchronization Henon discrete-time chaotic system which utilizes as a pseudo-arbitrary number generator to build encryption and digital signature. [37]. The enormous key space as 10158, affectability to misperception of parameters and preliminary condition on account of applying chaos in encryption make this plan certain to be utilized as a part of secure correspondence.

2.2 SECURITY ANALYSIS OF ENCRYPTED IMAGE

Security investigation is the specialty of discover the shortcoming of a cryptosystem and recovery of entire or a piece of a ciphered message (here we consider an image) or discovering the mystery key without knowing the decryption key or the algorithm. There are numerous methods to investigate, contingent upon what access the expert has to the plaintext, cipher content, or different parts of the cryptosystem. The following are probably the most widely recognized sorts of assaults on encrypted images:

2.2.1 Key Space Analysis

Attempt to discover the decryption key by checking all conceivable keys. The quantity of attempt to discover specifically denotes to key space of the cryptosystem become exponentially with aggregate key size. It implies that multiplying the key size for an algorithm does not just twice over the obliged number of operations, but instead squares them. An encryption algorithm with a 128 bit in key size characterizes a key space of 2128, which takes around 1021 years to check all the conceivable keys, with superior computers of these days. So a cryptosystem with key size of 128 bit computationally looks powerful against a beast energy assault.

2.2.2 Statistical Analysis

Original and encrypted image relationship can be determined by analysing data statistically. In this manner, image after encryption must be totally differentiate from the original. Because of Shannon hypothesis.

It is conceivable to illuminate numerous sorts of images by statistical investigation. For a image there are a few approaches to figure out if the ciphered image releases any data about the first one or not.

2.2.3 Correlation Analysis

Two contiguous pixels in a plain image are intensively corresponded vertically and on a level plane. The most extreme estimation of relationship coefficient is 1 and the base is 0 considered as the property of an image, where a strong image that has been encrypted to measurable assault ought to have a connection coefficient estimation of 0.

2.2.4 Differential Analysis

The point of this examination is to focus the affectability of encryption algorithm to minor changes. On the off chance that an challenger can make a little change (e.g. one pixel) in the plain image to watch the results, this control ought to cause a noteworthy change in the image that has been encrypted and the challenger ought not to have the capacity to discover a compelling relationship between the original and the image that has been encrypted as for distribution and misperception, the distinct assault loses its productivity and get to be inadequate..

2.2.5 Key Sensitivity Analysis

Moreover of vast enough key space to oppose a cryptosystem at brute force attack, additionally a protected algorithm ought to be totally delicate to mystery key which implies that the encrypted image can't be decrypted by somewhat changes in mystery key.

3 Proposed Work

A Latin square of order N is an N *N array filled with a symbol set of N distinctive elements, with each symbol appears exactly once in each row and each column. The name Latin Square is motivated by the mathematician Leonhard Euler, who used Latin characters as symbols.

Mathematically, we can define a Latin square L of order N via a tri-tuple function f

L of (r; c; i) as follows fL(r; c; i) =1 ; L(r; c) = Si 0 ; Otherwise (1)

where r denotes the row index of an element in L with r 2 N = f0,1….N1g; c denote the column index of an element in L with c 2 N; i denotes the symbol index of an element in L with I 2 N; and Si is the ith symbol in the symbol set S = fS0; S1; ; SN1g.

Therefore, if L is a Latin square of order N , then

for arbitrary c; i 2 N, we have

N 1 X r=0 fL(r; c; i) = 1 (2)

for arbitrary r; i 2 N, we have

N 1 Xc=0fL(r; c; i) = 1 (3)

which implies that each symbol appears exactly once in each row and each column in L.

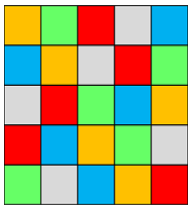
Fig. 1 shows examples of Latin squares at different orders with various symbol sets. It is worthwhile to note that the popular Sudoku puzzle [88] is also a special case of Latin square with additional block constraint as shown in Fig. 1(d).

A	B	C
B	C	A
C	A	B

(a) 3*3

■	Δ	o	◇
o	◇	■	Δ
Δ	■	◇	o
◇	o	Δ	■

(b) 4 *4



3	4	5	2	1	6	7	9	8
8	6	1	9	3	7	4	2	5
9	7	2	5	8	4	1	3	6
2	1	9	6	4	3	5	8	7
5	8	6	7	2	9	3	4	1
4	3	7	1	5	8	2	6	9
6	2	4	8	7	1	9	5	3
1	9	3	4	6	5	8	7	2
7	5	8	3	9	2	6	1	4

(c) 5 *5 (d) 9 *9

Figure 3.1 Latin square examples

Throughout of the paper, we are interested in N N Latin squares with the symbol set of integers from 0 to N-1, i.e. S = {0,1, ……. , N-1}

3.1.1 Latin Square Generator

Although Latin squares can be generated via a variety of means, for the sake of simplicity we use Algorithm 1described below for Latin square generation in the paper.

Algorithm 1. A Latin Square Generator L = LSG(Q1;Q2)

Require: Q1 and Q2 are two length-N sequences

Ensure: L is a Latin square of order N

Qseed = SortMap(Q1)

Qshift = SortMap(Q2)

for r = 0 : 1 : N 1 do

L(r; :) = RowShift(Qseed;Qshift(r))

end for

In Algorithm 1, both Q1 and Q2 are length-N sequences from a pseudo-random number generator (PRNG), e.g. Linear Congruential Generators (LCG); SortMap(Q) is a function which finds the index mapping between a sequence Q and its sorted version Q*in the ascending order; and RowShift(Q; v) ring shifts the sequence Qwith v elements towards left.

For example, if we want to generate a 4 4 Latin square L with

Q1= [.1,.6,.7,.9] and Q2= [.3, .9, .4, .2]

Then function SortMap(:) first calculates the sorted version of the input sequence and obtains

Q*1= [.1,.6,.7,.9] and Q*2= [.2, .3, .4, .9]

it then compares Q1with Q*1and Q2 with Q*2 and obtains the element mapping sequences as

Qseed = SortMap(Q1) = [0; 1; 3; 2] and Qshift = SortMap(Q2) = [3; 0; 2; 1], where the permutation sequences Qseed and Qshift indicate for i 2 {0; 1; 2; 3} Q*1 (Qseed

(i)) = Q1 (i) and Q*2(Qshift(i)) = Q2(i)

Finally, function RowShift(Q; v) left shifts Qseed with the amount v = Qshift (r) indicated by the rth element in Qshift, and assign this row to be the rth row in L. We therefore have the 4 4 Latin square L:

whose 1st row is obtained by left shifting Qseed for 3 units; 2nd row is obtained by left shifting Qseed for 0 unit; 3rd row is obtained by left shifting Qseed for 3 units; and 4th row is obtained by left shifting Qseed for 1 unit. It is noticeable that this shifting amount sequence {3; 0; 2; 1} is indeed Qshift.

3.2 SUBSTITUTION-PERMUTATION NETWORK

In cryptography, an input message and its corresponding output message of a cryptsystem are referred to as plaintext and ciphertext, respectively. A substitution-permutation network is a cipher structure composed of a number of substitution and permutation ciphers with multiple iterations. This structure is widely used in many well-known block ciphers, e.g. Rijndael i.e. AES, and ensures good confusion and diffusion properties [89].

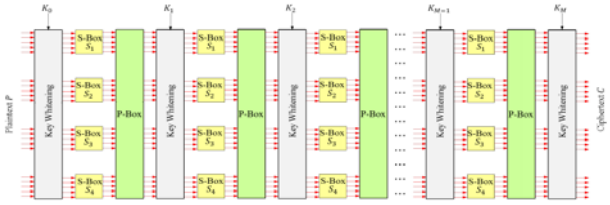


Figure 3.2 A M-round substitution-permutation network for block ciphers

A typical M -round SPN for block ciphers has a structure shown in Fig. 3.2.Conventionally in a SPN, plaintext, commonly in the form of a bit stream and denoted as P , is the original message to be encrypted; Key Whitening denotes an operation to mix the plaintext P with a round key; S-Box denotes a substitution-b ox, which maps one input byte to another in a deterministic way; P-Box denotes a permutation-box, which shuffles bit positions within the input bit stream in a deterministic way; and ciphertext denotes the output bit stream C, which is an encrypted message by the SPN. The decryption process of a SPN cipher is simply to reverse the arrow directions of all processing and to use inverse S-Box, and inverse P-Box instead.

The classic SPN ciphers are able to obtain good Shannon's confusion and diffusion properties [89]. For the diffusion property: if one changes one bit in plaintext P the corresponding ciphertext C changes in many bits. This one-bit change results in a different byte after passing through a S-Box, then leads more byte changes after passing through a P-Box, so on and so forth in each cipher round. Finally, one-bit change leads to significant changes in ciphertext C. The confusion property is the similar to the diffusion property. One bit change in encryption key K; will spread over all bits and result significant changes in ciphertext C.

3.3 PSYCHOVISUAL REDUNDANCY IN IMAGE

As a typical type of two-dimensional data, digital images contain many kinds of redundancies:

1) Coding redundancy, which requires expressing a pixel with the number of bits than the optimal number;

2) Temporal or spatial redundancy, which implies strong correlations between pixels within a neighborhood; and

3) Psycho visual redundancy, which is visually negligible details in the human vision system.

The first two redundancies about real visual information are commonly removed by lossless compression techniques, such as Huffman coding and lossless predictive coding. In contrast, psychovisual redundancy within an image is not essential for normal visual processing, and thus it is commonly removed by lossy compression techniques, such as quantization techniques used in JPEG [90]. Besides image compression, psychovisual redundancy is also commonly used for data hiding,

especially for the least significant bit-plane (LSB) data hiding techniques [90], which encode secret information within these unintelligible redundancies for human visual perception

3.4 LATIN SQUARE IMAGE CIPHER

To standardize the encryption/decryption processing, the cipher processing block is set to a 256* 256 grayscale block, i.e. its pixel intensity is denoted as a 8-bit byte. In the rest of the paper, we use P to denote a 256*256 plaintext image block, C to denote a corresponding ciphertext image block of P , L to denote a keyed Latin square of order 256, and K to denote a 256-bit encryption key. The new proposed Latin square image cipher is of a SPN structure with eight rounds as shown in Fig. 4.1. It is composed of the probabilistic encryption stage noise embedding in LSB and the SPN stage containing three encryption primitives Latin Square Whitening, Latin Square Substitution and Latin Square Permutation. It is worthwhile to note that this SPN is of a loom-like structure designed for image data, which encrypts plaintext image along rows and columns iteratively. In the rest of this section, we will discuss these stages and the detail encryption/decryption algorithms for LSIC.

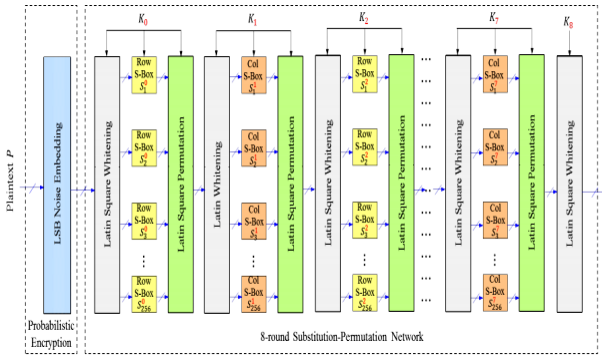


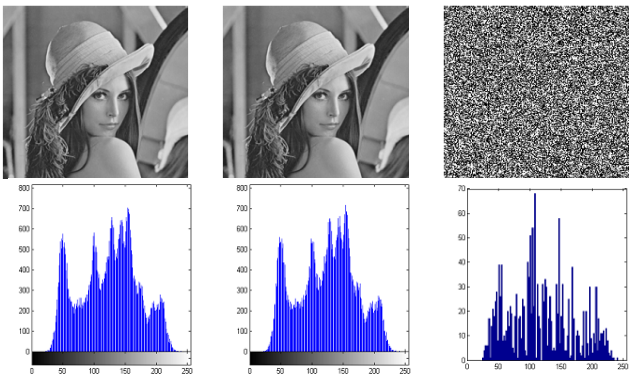
Figure 4.3 Overview of the Latin square image cipher

3.4.1 LSB Noise Embedding

Probabilistic encryption [91] means to use randomness in a cipher, so that this cipher is able to encrypt one plaintext with the exact same encryption key to distinctive ciphertexts. It is well known that such randomness is crucial to achieve semantic security. In this paper, we introduce such randomness by embedding noise in the least significant bit-plane of an image. More specifically, we XOR a randomly generated 256 256 bit-plane with the least significant bit-plane of the plaintext image, where the generation of this random bit-plane is completely independent of the encryption key. Fig. 4.4 shows an example of LSB noise embedding. Once again, this introduced noise in LSB does not affect any image visual quality from the point view of human visual perceptibility. However, any slight change in plaintext here will lead to significant changes in ciphertext after it is encrypted by by the SPN.

3.4.2 Key Translation

In conventional block ciphers, keys are directly used without translation, e.g. in key whitening process, but the proposed LSIC uses a 256-bit encryption key K with key translation to eight key-dependent Latin squares of order 256 before using in LSIC. Specifically speaking, for a given 256-bit encryption key K, we



(a) (b) (c)

Figure 3.4 Noise embedding in LSB – (a) plaintext Lenna P with histogram, (b) noise embedded plaintext P 0 with histogram, and (c) jP P 0 j with the difference of histograms WU et al. 7

1) Divide the encryption key K using function SubKeyDiv into eight 32-bit subkeys, i.e.
K = [k0; k1; ; k7]
2) Generate pairs of pseudo-random sequences
(Q01;Q02); (Q1;Q12); ; (Q81;Q82),
each pair with 2 256 ele-ments by using PRNGs by feeding these subkeys as seeds.
3) Generate key-dependent Latin squares i.e. L0; L1; ; L8 with the order of 256 by feeding these pseudo-random sequences in Algorithm 1. Namely, 8n 2 f0; 1; 8g, we have
Ln = LSG(Qn1;Qn2)
The first two steps can be realized via Algorithm 2 with encryption key K and M = 8.
Algorithm 2. Key Dependent Sequence Generator (Q1; Q2) = KDSG(Key;M)
Require: K is a 256-bit key
Require: n is a nonnegative integer
Ensure: and are n-element set of random sequences, each of a length 256.
K0 = K
for n = 0 : 1 : M do
$[k_0, k_1, \cdots, k_7] = SubKeyDiv(K_n)$
for i = 0 : 1 : 8 do
$q^i(0) = PRNG(k_i)^1$
for j = 1 : 1 : 63 do
$q^i(j) = PRNG(q^i(j-1))$
end for
end for
$Q_1^n = \left[q^0(0:31), q^1(0:31), \cdots, q^7(0:31)\right]$ %^{2}
$Q_2^n = \left[q^0(32:63), q^1(32:63), \cdots, q^7(32:63)\right]$
$K_{n+1} = \left[q^0(63), q^1(63), \cdots, q^7(63)\right]$
end fo

All PRNGs can be used in Algorithm 2, but not all of them are secure for cryptography, for example, the linear congruential generator [92]. Using cryptographic secure PRNGs in Algorithm 2 can further enhance the cipher security, for example, PRNGs from eSTREAM project 3 with nonce [93].

3.4.3 Latin Square Whitening

In the conventional SPN for block ciphers, the Whitening stage normally mixes a plaintext message P with a round key, e.g. XOR operation in [94], such that The statistics of the plaintext message P is redistributed after mixing. The relationship between ciphertext and encryption key is very complicated and involved.
In image encryption, a plaintext message is an image block, P , composed of a number of pixels. Each pixel is represented by several binary bits (a byte). Therefore, XOR whitening scheme become inefficient for image data, in the sense that it requires to extend an encryption key to be a equal size to a plaintext image, and to impose bitwise XOR to byte pixels. And this type of image encryption is called a naive algorithm [95]. Since the objective of key whitening is to mix plaintext data with encryption keys, we therefore define whitening as a transposition cipher [95] over the finite field GF (28) for image data, as shown in Eq. (4)
y = [x + l]28 (4)1 Any PRNG can be used here by taking the key as its seed.2qi(j1 : j2) denotes a vector of pseudo-random numbers with elements
[qi(j1); qi(j1 + 1); ; qi(j2 1); qi(j2)].3
where x is a byte in plaintext, l is a corresponding byte in the keyed Latin square, y is the whitening result and[:]28 denotes the computations over GF (28). Above whitening process can be easily reversed by applying Eq. (5).

$$x = [y + l]_{2^8} \quad (5)$$

In image encryption, plaintext byte x is a pixel, say it is located at the intersection of rth row and cth column i.e. x = P (r; c). Now let l = L(r; c) be an element located at the corresponding position in the keyed Latin square L, and y be the ciphertext byte with y = C(r; c), then we have the pixel-level equation

$$\begin{cases} C(r,c) = [\mathbf{SR}\,(P(r,c), [D]_3) + L(r,c)]_{2^8} \\ P(r,c) = \mathbf{SR}\,([C(r,c) + L(r,c)]_{2^8}, [D]_3) \end{cases} \quad (6)$$

where symbol n denotes the current round number (n 2 [0; 7]), D = L(0; 0) is the rotating parameter, and SR denotes the spatial rotating function (X; d) rotates an image X according to different values of the direction d as defined in Eq. (7)

$$Y = \mathrm{SR}(X,d) = \begin{cases} X, & \text{if } d = 0 \\ \text{Flip } X \text{ up} \rightarrow \text{down}, & \text{if } d = 1 \\ \text{Flip } X \text{ left} \rightarrow \text{right}, & \text{if } d = 2 \end{cases} \quad (7)$$

Notice that if Y =SR(X; d), then the following identity always holds X = SR(Y; d) (8)
Apply key whitening for all pixels using the pixel-level Eq. (6), the Latin Square Whitening(LSW) in the image-level then can be denoted as

$$LSW : \begin{cases} C = Ecr_w(L,P,D) \\ P = Dcr_w(L,C,D) \end{cases} \quad (9)$$

Therefore, we can restore the plaintext image block P from the ciphertext image block C using Eq. (9).

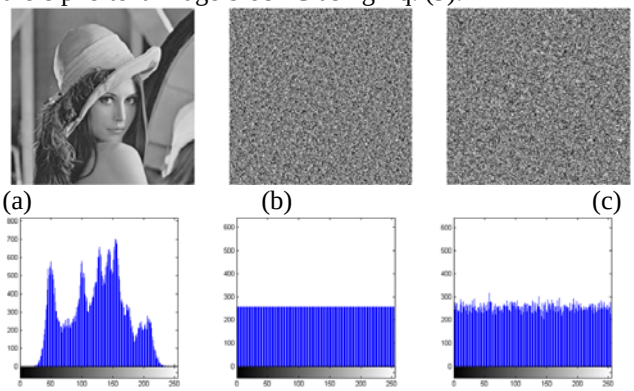


Figure 3.5 A Latin Square Whitening example - (a) plaintext Lenna P , (b) reference Latin square L, and (c) ciphertext C = Ecrw(L; P; 0)

Fig. 4.5 shows an example of Latin Square Whitening, where the first row shows images and the second row shows corresponding histograms of these images. From this example, it is easy to verify that the ciphertext image after the Latin Square Whitening is unrecognizable and its pixels are redistributed to uniform-like.

3.4.4 Latin Square Row and Column Bijections

Since Eqs. (2) and (3) hold, each row and each column in a Latin square L of order N is a permutation of the integer number sequence [0; 1; ;N 1], we can define bijections (one-to-one and onto mapping) by mapping this integer number sequence to either a row or a column in a Latin square, which is a permutated sequence of the integer number sequence.

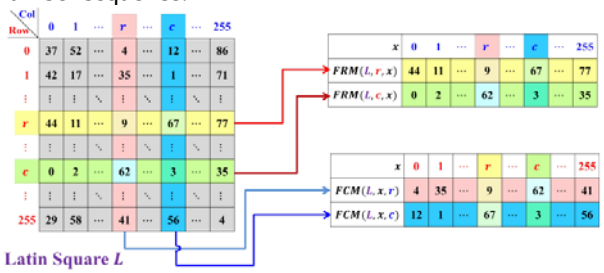


Figure 3.6 Examples of building forward row mappings and forward column mappings using a Latin square

In other words, we are able to construct forward and inverse row mapping functions (FRM and IRM) with respect to the rth row in L as shown in Eq. (10), and also forward and inverse column mapping functions (FCM and ICM) with respect to the cth column in L as shown in Eq. (11), where x and y denote the input and output of the mapping functions, respectively.

$$\begin{cases} y = FRM(L,r,x) = L(r,x) \\ x = IRM(L,r,y) = \arg\max\limits_{z\in\mathbb{N}} \left(f_L(r,z,y)\right) \end{cases} \quad (10)$$

$$\begin{cases} y = FCM(L,x,c) = L(x,c) \\ x = ICM(L,y,c) = \arg\max\limits_{z\in\mathbb{N}} \left(f_L(z,c,y)\right) \end{cases} \quad (11)$$

where fL is the tri-tuple function defined in Eq. (1). Its maximum is equal to 1, i.e. fL(r; x; y) = 1, only for the column number x satisfying the constraint, L(r; x) = y. Further, we have row mapping identities hold for arbitrary x and y within a Latin square

$$\begin{cases} IRM(L,r,FRM(L,r,x)) = x \\ FRM(L,r,IRM(L,r,y)) = y \end{cases} \quad (12)$$

Similarly, we also have column mapping identities as follows:

$$\begin{cases} ICM(L,FCM(L,x,c),c) = x \\ FCM(L,ICM(L,y,c),c) = y \end{cases} \quad (13)$$

Fig. 3.6 shows FRM and FCM functions defined by a Latin square. As can be seen, given a row number r, the effect of forward row mapping is to use this Latin square as a look-up table and to find the corresponding element in row r. Similarly, given a column number c, the effect of forward column mapping to use this Latin square as a look-up table and to find the corresponding element in column c. Such nice property is directly from the constructional constraint in a Latin square.

Since a N th order Latin square has N rows and N columns, there are N bijections from N rows and another N bijections from N columns in the Latin square. It is well-known that a bijection can be directly used as a P-Box [96] and can also serve as a S-Box [28]. We therefore are able to construct S-Boxes and P-Boxes from a Latin square.

3.4.5 Latin Square Substitution

An S-Box in cryptography is a basic component performing byte substitution. Each S-Box can be defined as a bijection, also known as a one-to-one and onto mapping. In image encryption, an image pixel is commonly represented as a byte, i.e. a sequence of bits. For example, 8-bit grayscale image has 256 gray intensity scales with each intensity scale represented in an 8-bit sequence. Because of the existence of FRM/IRM and FCM/ICM bijections in a Latin square, we are able to perform byte substitution in an image cipher using bijections from rows and columns in a Latin square. The substitution with respect to a row in a Latin square is called Latin Square Row S-box(LSRS) in this paper:

$$LSRS: \begin{cases} C = Ecr_s^{row}(L,P) \\ P = Dcr_s^{row}(L,C) \end{cases} \quad (14)$$

In regard to pixel-level function of LSRS, each ciphertext byte is determined by the FRM function (see Eq. (10)) using the keyed Latin square L with function parameters given by plaintext bytes and ciphertext bytes as follows

$$Ecr_s^{row}: C(r,c) = \begin{cases} FRM\left(L, C(r-1,c), P(r,c)\right), \text{ if } r \neq 0 \\ FRM\left(L, 0, P(r,c)\right), \text{ if } r = 0 \end{cases} \quad (15)$$

Clearly, plaintext bytes then can be perfectly restored from ciphertext bytes, if we use IRM instead of FRM as follows

$$Dcr_s^{row}: P(r,c) = \begin{cases} IRM\left(L, C(r-1,c), C(r,c)\right), \text{ if } r \neq 0 \\ IRM\left(L, 0, C(r,c)\right), \text{ if } r = 0 \end{cases} \quad (16)$$

Similarly, we use bijections from columns in a Latin square to perform byte substitutions. And this is called Latin Square Column S-box(LSCS) i.e.

$$LSCS: \begin{cases} C = Ecr_s^{col}(L,P) \\ P = Dcr_s^{col}(L,C) \end{cases} \quad (17)$$

and the corresponding LSCS encryption and decryption process then can be defined as:

$$Ecr_s^{col} : C(r,c) = \begin{cases} FCM(L, P(r,c), C(r,c-1)), & \text{if } c \neq 0 \\ FCM(L, P(r,c), 0), & \text{if } c = 0 \end{cases} \quad (18)$$

$$Dcr_s^{col} P(r,c) = \begin{cases} ICM(L, C(r,c), C(r,c-1)), & \text{if } c \neq 0 \\ ICM(L, C(r,c), 0), & \text{if } c = 0 \end{cases} \quad (19)$$

Fig. 3.7 shows encryption results of Latin Square Row S-box and Latin Square Column S-box. As can be seen, the plaintext image block P becomes unrecognizable after applying either LSRS or LSCS. Histogram analysis also shows that the statistics of the pixel intensity changes dramatically after substitution.

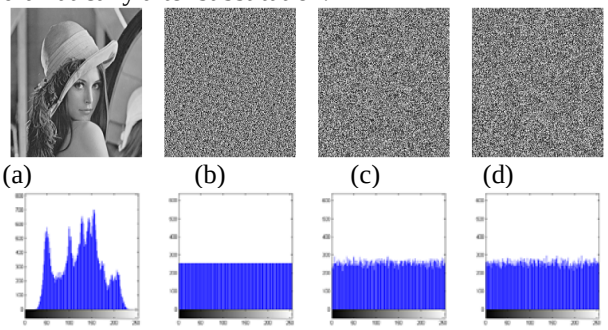


Figure 3.7 A Latin Square Substitution example - (a) plaintext P : Lenna Image, (b) Latin square L, (c) ciphertext , and(d) Ciphertext

The Latin square row/column substitution defined above has excellent diffusion properties. One pixel change in the plaintext P will diffuse to a column/row of pixels after a round of LSRS or LSCS. This diffusion quickly spreads to the entire ciphertext image in several cipher rounds.

3.4.6 Encryption/Decryption Algorithms

After constructing Latin square based encryption primitives for image data, we now are able to construct the SPN cipher as shown in Fig. 4. Algorithm 3 and 4 describe the encryption and decryption process of the Latin square image cipher, respectively.

4 LSBNoiseEmbedding could be any function to embed random binary noise in the least significant bit-plane of a plaintext image, e.g. a function randomly change the parity of a plaintext pixel on its LSB.

Algorithm 3. Latin Square Image Cipher- Encryption C = E(P;K)
Require: K is a 256-bit key
Require: P is a 256 256 8-bit grayscale image block
Ensure: C is a 256 256 8-bit grayscale image block
(Q1; Q2) = KDSG(K; 8)
for n = 0 : 1 : 7
do
if n == 0 then
end if
if mod(n; 2) 6 = 0 then
else
end if
end for

Algorithm 4. Latin Square Image Cipher- Decryption P = D(C;K)
Require: K is a 256-bit key
Require: C is a 256*256 8-bit grayscale image block
Ensure: P is a 256*256 8-bit grayscale image block
(Q1; Q2) = KDSG(K; 8)for n = 7 : 1 : 0 do
if n == 7 then
end if
if mod(n; 2) 6 = 0 then
else
end if
end for
P = PLSW

4. Result Analysis

Our design criteria behind the Latin Square Substitution are to achieve the following objectives,

1) A ciphertext image is very sensitive to any slight change in a plaintext image.

2) A deciphered image is insensitive to slight change in a ciphertext image.

In practice, Latin Square Substitution is of an asymmetric structure as shown in Fig. 10, in the sense that encrypting one plaintext byte requires one plaintext byte and one ciphertext byte, while decrypting one ciphertext byte requires two ciphertext byte but no plaintext byte. This process is similar to weave a thread of ciphertext bytes on the ciphertext image, which has to be done by intersecting the longitudinal threads of plaintext bytes.

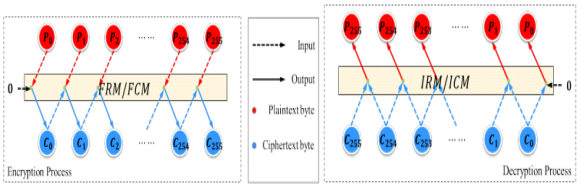


Figure 4.1 Latin Square Substitution has an asymmetric structure for encryption and decryption

As a result, changing one pixel in a plaintext image influence all pixels after it in its row in the first encryption round, further all pixels after these influenced pixels in corresponding columns in the second encryption round, and then more pixels in the third encryption round, so on and so forth until the last round. As a result, this small change in plaintext image will lead to a completely different ciphertext image. In contrast, changing one pixel in a ciphertext image only leads a two-pixel-change in the first decryption round, and at most 28 = 256 pixel changes in the deciphered image. Hence the Latin Square Substitution indeed achieves both goals above and gains encryption diffusion properties against plaintext change and decryption robustness against ciphertext noise simultaneously.

The proposed Latin square image cipher is carefully designed with respect to image data. We use the psychovisual redundancy within an image and realize the probabilistic encryption by introducing random noise in the LSB of a plaintext image block. We also notice the information redundancy within adjacent image pixels, which might lead to information leakage by estimating a pixel's intensity from its neighbors. We break these high

correlated neighbour pixels by applying Latin Square Whitening, which shifts pixels within a homogenous region to random-like in the sense that shifting amounts of any two highly correlated pixels within a row or a column in the homogenous region are always different. This way prevents the prediction of a pixel's intensity from its neighbors and thus decorrelates these pixels. In addition, Latin Square Whitening also allows to spatial transform a plaintext image in a random-like manner. Further our proposed Latin Square Substitution substitutes pixel bytes along rows and columns iteratively like weaving in a loom, if we consider a row of pixels is a longitudinal thread and a column of pixels is a latitudinal thread. Finally Latin Square Permutation shuffles plaintext image pixels all over the domain. In summary the SPN used in LSIC differs from the conventional SPN block cipher.

Table 4.1 Proposed Method Comparison with Information Entropy

P	Channeln Method	Wang etal. 2011[107]	Liu etal. 2011[108]	Patidar et l.2011 [109]	Proposed Method
4.2.04 (Color Lenna)	Red	7.999324	7.9871	7.9957	7.999351
	Green	7.999371	7.9802	7.9963	7.999416
	Blue	7.999292	7.9878	7.9951	7.999314
4.2.07 (Color Pepper)	Red	N/a	7.9877	7.9952	7.999309
	Green	N/a	7.9881	7.9959	7.999285
	Blue	N/a	7.9877	7.9954	7.999258

5. CONCLUSION AND FUTURE WORK

At existing eras where the most important communication is through wireless techniques using internet network to transfer data, so main concerns are on the subject of the security of such personal or countries defence data. Encryption is unique way to guarantee worthy security from unofficial access at many grounds. Image encryption is striking extent for research in this case because communication with the support of multimedia objects is growing promptly. Various important encryption techniques have been presented in demand to make it acquainted with a number of encryption algorithms used in encrypting the image which has been transmitted over network. The outcome of every algorithm has advantages and disadvantages based on their techniques which are being practised on images.

In this paper, we introduce a symmetric-key Latin square image cipher with probabilistic encryption for grayscale and color images.

The image encryption and decryption algorithm is designed and implemented to provide confidentiality and security in transmission of the image based data as well as in storage. Future work will be focused on the development of this algorithm to get exactly errors equal to zero.

Proposed system provides diffusion and confusion concepts which presented good security. Therefore the working and developing on it is very visible. We can suggest some future works: Random key generation: which make the system able to• generate efficient random key (the key has good randomness) without selected it. Hashing: The proposed system ought to be proficient in• being changed to a one-way hash function.

REFERENCES

[1] J. Daemen and V. Rijmen, AES Proposal: Rijndael, AES Algorithm Submission, September 3, 1999.

[2] M. Zeghid, M. Machhout, L. Khriji, A. Baganne and R. Tourki, A Modified AES Based Algorithm for ImageEncryption, World Academy of Science, Engineering & Technology, 2007.

[3] B. Subramanyan, V. M. Chhabria, T. G. S. babu, Image Encryption Based On AES Key Expansion, Second International IEEE Conference on Emerging Applications of Information Technology, 2011.

[4] J. Daemen and V. Rijmen, AES Proposal: Rijndael, AES Algorithm Submission, September 3, 1999.

[5] M. Zeghid, M. Machhout, L. Khriji, A. Baganne and R. Tourki, A Modified AES Based Algorithm for ImageEncryption, World Academy of Science, Engineering & Technology, 2007.

[6] B. Subramanyan, V. M. Chhabria, T. G. S. babu, Image Encryption Based On AES Key Expansion, Second International IEEE Conference on Emerging Applications of Information Technology, 2011.

[7] Q. Gong-bin, J. Qing-feng and Q. Shui-sheng, A New Image Encryption Scheme Based on DES Algorithm and Chuas Circuit, Int. Journal of Computer Science and Network Security, VOL.8, No.4, April 2008.

[8] W. Diffie and M. E. Hellman, New Directions in Cryptography, IEEE Transactions on Information Theory, Vol. 22, Issue 6, Nov 1976.

[9] H. Shuihua and Y. Shuangyuan, An Asymmetric Image Encryption Based on Matrix Transformation, Transactions on Computer and Infor- mation Technology, Vol. 1, No. 2, 2005.

[10] K.Ganesan, I. Singh and M. Narain, Public Key Encryption of Images and Videos in Real Time Using Chebyshev Maps, Fifth International Conference on Computer Graphics, Imaging and Visualization, 2008