Efficient Hybrid Image Cryptosystem using ECC and Chaotic Map

Kamlesh Gupta
Jaypee University of Engg. &
Technology, Guna, M. P. India

Sanjay Silakari
University Institute of Technology,
RGPV,Bhopal, M. P. India

ABSTRACT

Encryption is used to securely transmit data in open networks. Each type of data has its own features; therefore different techniques should be used to protect confidential image data from unauthorized access. The asymmetric cryptography doesn't fit for transmission of Images, since it takes more time for encryption at the source and likewise for decryption at the destination. We are therefore using hybrid cryptosystem, In this paper, we designed an algorithm to generate diffusion template using 3D standard map and rotated image by using vertically and horizontally red and green plane of the input color image. We then shuffled the red, green, and blue plane by using 3D Cat map and Standard map. Finally the Image is encrypted by performing XOR operation on the shuffled image and diffusion template. The objectives of this new design includes: (1) to efficiently extract good pseudorandom sequences from a cascading of 3D cat and Standard map for color image and (2) to simultaneously perform permutation and diffusion operations for fast encryption. The symmetric key cryptography totally depends on the session key, we are using ECC algorithm for session key exchange in secure manner.

Keywords

Image encryption, Chaos function, Cat map, Standard map, Elliptic Curve Cryptography.

1. INTRODUCTION

In today's Hi-Tech world of innovations in the field of internet, medical imaging systems, military message communication, it is needed to transmit the images through network confidentially. For reliable and secure image transmission, we heavily rely on image cryptography which is the base of security in digital communication. Many image encryption methods are available to protect the content of digital images but, few of them are at par with the expected goals i.e. speed, reliability and security. For the above mentioned application the transmission of image needs to be fast and highly reliable. Beside this, internet applications have to deal with security issue also. They want to be protected and to ensure their privacy; network security and image encryption has become important. The traditional encryption schemes do not fit for modern image transmission requirement. Many researchers have tried to innovate better solutions for image encryptions. In particular, application of chaos theory in multimedia encryption is one of the important research directions. The field of chaotic cryptography has undergone tremendous growth over the past few decades. The primary motivation of employing chaotic systems is its simplicity in form and complexity in dynamics.

In their seminal paper [1], Diffie and Hellman introduced the notion of public key cryptography. They described how two entities can agree on a common secret key by communicating over an insecure channel. This is known as the Diffie-Hellman key agreement protocol. The security of the protocol is related to the apparent difficulty of computing discrete logarithms modulo of a large prime number p, i.e. given two numbers (g mod p) and (g^X mod p), it seems to be infeasible to compute x for general large enough p.

A few years later, Rivest, Shamir and Adleman [2] proposed a public key encryption scheme and a digital signature scheme, the security of which is related to factoring a large integer. The papers [1] and [2] laid the foundations of public key cryptography. Since their appearance, many other schemes have been proposed that are based on the Integer Factorisation Problem and the Discrete Logarithm Problem (DLP), such as the El-Gamal encryption and signature scheme [3] and the Digital Signature Algorithm (DSA) [4].

Instead of using the DLP modulo of a large prime p as the basis of cryptographic protocols, one can consider the DLP in an arbitrary group that admits an efficient element representation and group law. Let G be such a group, then the DLP in G is defined as follows: given two elements g and h = g^X € G, determine the exponent x. The reason for considering other groups is that the most efficient methods for solving the DLP in a general, i.e. black-box, group take O (order (g) $)^{1/2}$ steps [5]. The DLP for the integers modulo of a prime (or more generally, in the multiplicative group of any finite field F^*q) can be solved in a far more efficient way, requiring only $O(\exp(\log(q)^{1/2} \log(\log(q))^{1/2}))$ steps. So if one uses a group in which the DLP is as hard as for a general group, one can use much smaller parameters and key sizes than when using the multiplicative group of finite fields and still obtain the same level of security.

If the curves are chosen carefully then, as far as one knows, the DLP in these groups is as hard as for general groups. The ECC has the highest strength-per-bit compared to other public key cryptosystems. Small key sizes translate into savings in bandwidth, memory and processing power. This makes ECC the obvious choice in this situation. We used ECC for session key exchange in secured manner,

This research work attains the following objectives:

- **To transfer the images efficiently and securely over the internet.**

Asymmetric cryptography doesn't fit for transmission of images because of the bulk data capacity, strong pixel correlation and high redundancy. Moreover encryption at the source and decryption at the destination lowers the encryption performance.

a). To design and analyze chaos–based image encryption schemes and elliptic curve cryptography.

As per our study and analysis many existing schemes under these categories are found to merely achieve moderate or even low security. Only a few of them promise to achieve sufficient security, while maintaining a satisfactory speed performance.

2. PROPOSED HYBRID CRYPTOSYSTEM

We propose an innovative solution for the above problems by using hybrid cryptosystem in fig. 6,7, in which the image is encrypted by the chaos based encryption using an efficient permutation and diffusion technique. The chaos based encryption comes under the symmetric cryptography which depends on session key, so we are using Elliptic Curve Cryptography method for session key exchange in secure manner. The proposed cryptosystem are divided into several stages and explained below.

2.1 Diffusion Template:

The diffusion template must have the same size as main image. Let the main image have **m** number of rows **n** number of columns then the diffusion template is created as follows

$$(i,j,k) = round\left(\frac{255}{n} * j\right) \dots\dots\dots\dots\dots\dots.. (1)$$

$$where\ 1 \le i \ \le m, 1 \le j \le n\ and\ 1 \le k \le 3$$

Equation 1 form the matrix with all rows filled with linearly spaced number in between 0 to 255. The sequence is randomized by 3D standard map in discrete form as given below:

The 3D standard map randomizes the pixels by reallocating it in new position by utilizing its property of one to one mapping. Diffusion template is generated by the following equation:

$$i' = (i + j) modm \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots .... (2)$$

$$j' = \left[j + k + K1 * \sin\left(i * \frac{c}{2 * pi}\right) \right] \boldsymbol{modn} \dots (3)$$

$$k' = \left[k + K1 * \sin\left(i * \frac{p}{2 * pi}\right) + K2 * sin\left(j * \frac{p}{2 * pi}\right)\right] \boldsymbol{mod}\, p \dots\dots (4)$$

Where the K1, K2 are the integers, p=3 for the case of color image and i', j' k' shows the transformed location of i, j, k

$$\mathrm{I}'_{\mathrm{diff}}(\mathrm{i}', \mathrm{j}', \mathrm{k}') = \mathrm{I}_{\mathrm{diff}}(\mathrm{i}, \mathrm{j}, \mathrm{k})$$

2.2 Image Encryption:

Step1. The main image is divided into three separate images I_R, I_G and I_B as follows

$$I_R\,(x, y) = I(x, y, 1)$$

$$I_G\,(x, y) = I(x, y, 2)$$

$$I_B\,(x, y) = I(x, y, 3)$$

$$where\ 1 \le x \ \le m\ and\ 1 \le y \ \le n$$

Step2. The Red and Green image are transformed vertically and horizontally respectively. The blue image remains same and then the new image is reconstructed as per the following equations..

$$I'_R\,(x, y) = I_R\left(\left(x + \frac{m}{2}\right) \boldsymbol{mod}\, m, y\right)$$

$$I'_G(x, y) = I_R\left(x, \left(y + \frac{n}{2}\right) \boldsymbol{mod}\, n\right)$$

$$I'_B(x, y) = I_B\,(x, y)$$

$$I_{new}(x, y, 1) = I'_R\,(x, y)$$

$$I_{new}(x, y, 2) = I'_G\,(x, y)$$

$$I_{new}(x, y, 3) = I'_B\,(x, y)$$

Step3. The first level confusion by using 2D cat map is performed. Plane normal to R, G, B Planes is sliced by the following equations

$i'_{new}\,(i, j, k) = I'_{SRGB}(j', k') = I_{new}\,(i, j, k)$

$$where\ 1 \le i \ \le m, 1 \le j \le n\ and\ 1 \le k \le 3.$$

$$j' = \left(j + r_x + r_y + \ p * k\right) \boldsymbol{mod}\, m$$

$$k' = \left(q * j + r_y + (\ p * q + 1) * k\right) \boldsymbol{mod}\, n$$

Where,

j' and k' are obtained by 2D cat map,

p and q are integers > 0 and

r_x, r_y are offset integers such that $0 \le r_x \le m$, $0 \le r_y \le n$

Step4. Final confusion stage is generated by two cascade 3D maps; first by cat map then by standard map. So the transformation of location (i, j, k) into (i'', j'', k'') is performed by following equations.

$$i = \big[\big(1 + a_x a_z b_y\big) * i + a_z * j + \big(a_y + a_x * a_z + a_x * a_y * a_z * b_y\big) * k\big]\ \boldsymbol{mod}\ m$$

$$j = \big[\big(b_z + a_x * b_y + a_x * a_z * b_y * b_z\big) * i + (a_z * b_z + 1) * j + \big(a_y * a_z + a_x * a_y * a_z * b_y * b_z + a_x * a_z * b_y + a_x * a_y * b_y + a_x) * k\big]\ \boldsymbol{mod}\ n$$

$$k = \big[\big(a_x * b_x * b_y + b_y\big) * i + b_x * j + \big(a_x * a_y * b_x * b_y + a_x * b_x + a_y * b_y + 1) * k\big]\ \boldsymbol{mod}\ p$$

$$i = [(i + k)]\ \boldsymbol{mod}\ m$$

$$j = \left[i + j + K1 * \sin\left(i * \frac{n}{2*pi}\right)\right] \boldsymbol{mod}\ n$$

$$k = \left[k + K1 * \sin\left(i * \frac{p}{2*pi}\right) + K2 * \sin\left(j * \frac{p}{2*pi}\right)\right] \boldsymbol{mod}\ p$$

$$I_{conf}(i, j, k) = I_{new}\ (i, j, k)$$

$where\ a_x, a_y, a_z, b_x, b_y, b_z\ and\ K1, K2\ are\ integers\ > 0$

Each confusion step is followed by diffusion obtained by XOR operations performed between each pixel of I_{conf} and diffusion I_{diff}. i.e. $I_{encp} = I_{conf} \oplus I_{diff}$.

2.3 Key Generation Process

The proposed method has a large number of variables which can be used as key parameters but to avoid the exceptionally large key and decreased key sensitivity, the parameter which does not have considerable effect on encryption is avoided or scaled. The selected key parameters and their respective length are given below

Step1. Diffusion template shuffling D_s = 8 bits.

Step2. Diffusion template offset value

$D_xD_yD_z$ = 8 + 8 + 2 = 18 bits.

Step3. Diffusion template variables

$D_{k1}D_{k2}$ = 8 + 8 = 16 bits.

Step4. Sliced RGB plane Shuffling = S_s = 8 bits.

Step5. Sliced RGB plane offset values S_xS_y = 8 bits.

Step6. Sliced RGB Plane Variables

S_pS_q = 8 + 8 = 16 bits.

Step7. Final Confusion shuffling C_s = 8 bits.

Step8. Confusion offset of cat map

$C_xC_yC_z$ = 8 + 8 + 2 = 18 bits.

Step9. Confusion cat map variables

C_aC_b = 8 + 8 = 16 bits.

Step10. Confusion offset of standard map

$C_x{'}C_y{'}C_z{'}$ = 8 + 8 + 2 = 18 bits.

Step11. Confusion standard map variables

$C_{k1}C_{k2}$ = 8 + 8 = 16 bits.

Final key structures
$D_sD_xD_yD_zD_{k1}D_{k2}S_sS_xS_yS_pS_qC_sC_xC_yC_zC_aC_bC_x{'}C_y{'}C_z{'}$

$C_{k1}C_{k2}$

Total bits = 8 + 18 + 16 + 8 + 8 + 16 + 8 + 16 + 16 + 18 + 16 = 148 bits.

2.4 Session key exchange using ECC

Step1. The 148-bit session key generated by procedure 2.3, this is assume plain message $P_{M.}$

Step2. We consider an elliptic curve over a finite field associated with a prime number *p* > 3 whose equation is y^2 (mod p) = (x^3+ax+b) *mod p*……. (5)

Where a, b are two integers which satisfy $4a^3 + 27b^2 \neq 0$ (mod p).Then the elliptic group, E**p** (a, b), is the set of pairs (x, y), where 0 ≤ x, y < p,

satisfying the equation (5). The smallest value of n

for which n*G = O is a very large prime number. (*Here nG is special multiplication called Multiplication over an elliptic curve group).

Step 3: User A select a private key, $n_A < n$ and compute the public key P_A as: $P_A = n_A\, G$.

Step 4: P_A, p, *a* and the generator point *G* are made public. $2^n - 1 < N$

Where, N is defined as

$$NG = O$$

Step 6: Now for each combination of n bits a point

Step 5: User B divides the massage into group of n bits, the number of bits in group is defined by

P_M from G to (N – 1)G is used as replacement for that group of bits.

Step 7: The cipher text is generated using following formula

$$P_c = [(kG), (P_M + n_B P)]$$

$where\ n_B\ is\ user\ B\ private\ key$

Step 8: On receiving of cipher text user A finds the original points using the equation

$$P_M = (P_M + n_B P) - n_A(kG)$$

$where\ (P_M + n_B P)\ and\ (kG)\ are\ directly\ taken\ from\ P_c$

Step 9: The original message is generated by performing inverse of Step 5.

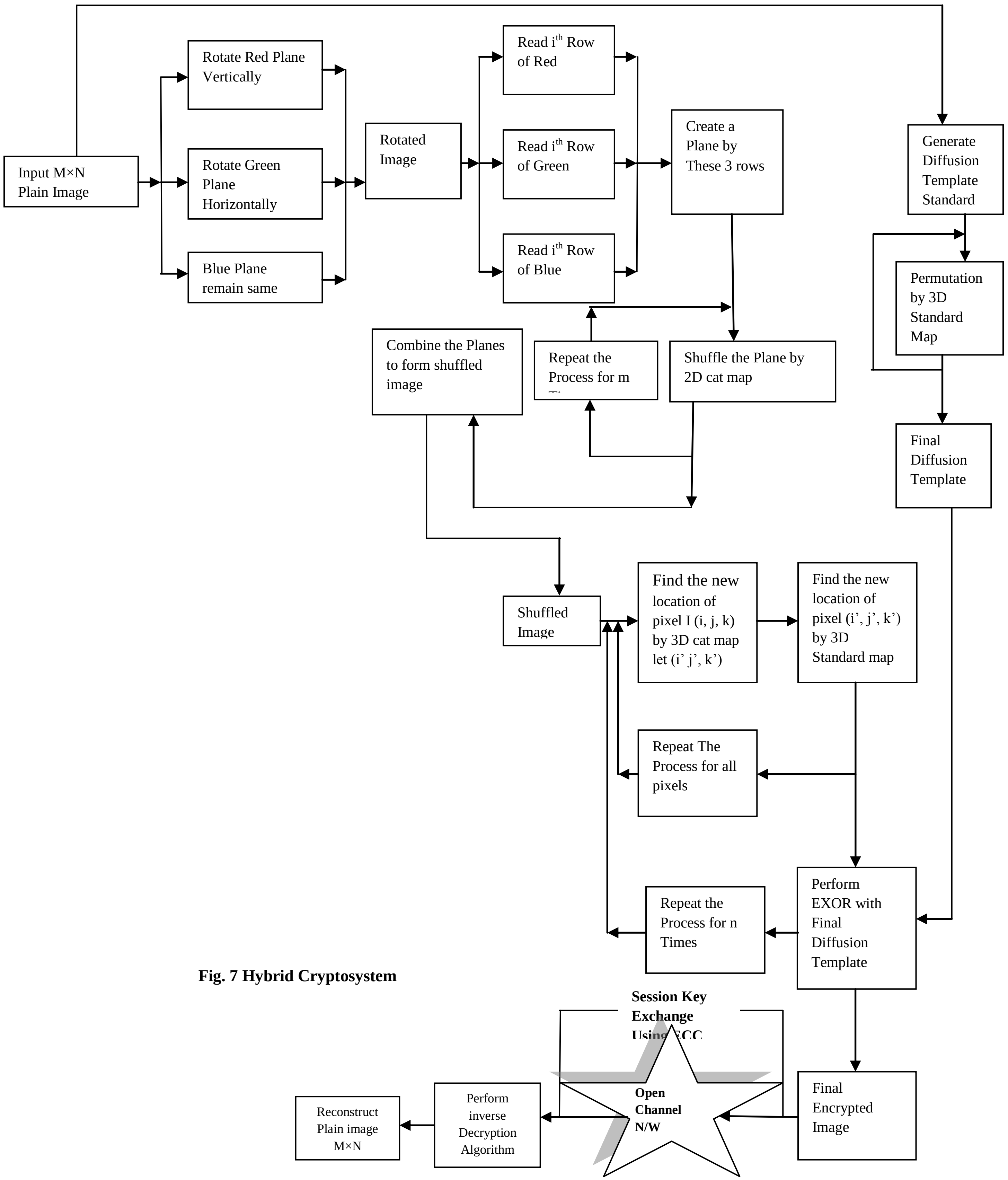
Input M×N Plain Image
Rotate Red Plane Vertically
Rotate Green Plane Horizontally
Blue Plane remain same
Rotated Image
Read ith Row of Red
Read ith Row of Green
Read ith Row of Blue
Create a Plane by These 3 rows
Generate Diffusion Template Standard
Permutation by 3D Standard Map
Final Diffusion Template
Combine the Planes to form shuffled image
Repeat the Process for m
Shuffle the Plane by 2D cat map
Shuffled Image
Find the new location of pixel I (i, j, k) by 3D cat map let (i' j', k')
Find the new location of pixel (i', j', k') by 3D Standard map
Repeat The Process for all pixels
Repeat the Process for n Times
Perform EXOR with Final Diffusion Template
Session Key Exchange Using ECC
Open Channel N/W
Final Encrypted Image
Perform inverse Decryption Algorithm
Reconstruct Plain image M×N

Fig. 7 Hybrid Cryptosystem

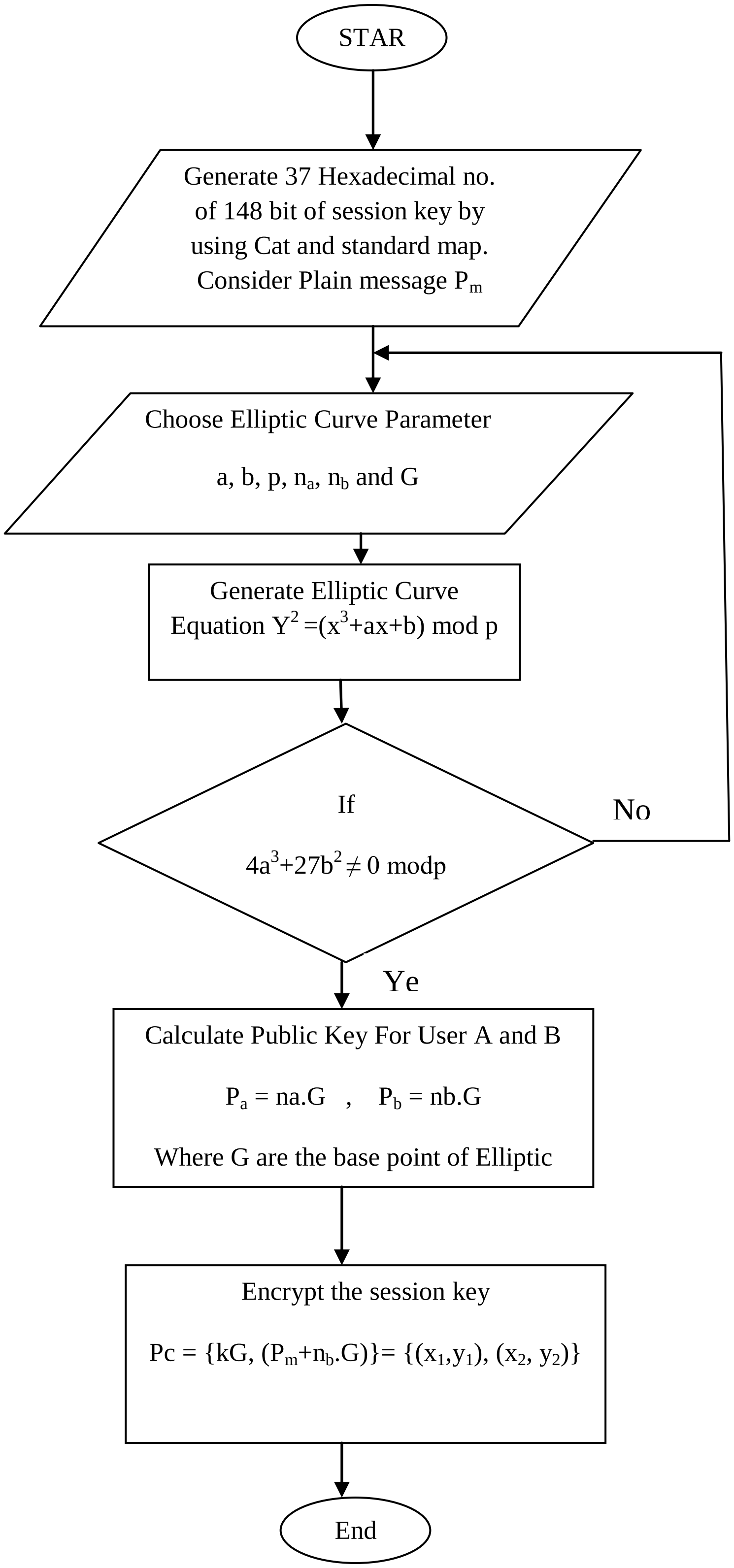
STAR
Generate 37 Hexadecimal no. of 148 bit of session key by using Cat and standard map. Consider Plain message P_m
Choose Elliptic Curve Parameter
a, b, p, n_a, n_b and G
Generate Elliptic Curve Equation $Y^2 = (x^3+ax+b) \bmod p$
If
$4a^3+27b^2 \neq 0$ modp
No
Ye
Calculate Public Key For User A and B
$P_a = na.G$, $P_b = nb.G$
Where G are the base point of Elliptic
Encrypt the session key
$Pc = \{kG, (P_m+n_b.G)\} = \{(x_1,y_1), (x_2, y_2)\}$
End

Fig. 6 Secure session key exchange using ECC

3. PERFORMANCE ANALYSIS

3.1 Key space Analysis

The strong point of the proposed algorithm is the generation of the permutation sequence with the chaos sequence. The key space should also be suitably large to make brute-force attack not feasible. In the proposed algorithm, we use 148 bit key (37 Hex number). It is observed in Figure 1 (a-b) that with slight variation in the initial condition of the chaotic sequence, the resultant image after decryption differs a lot from the original image. So it is very difficult to breach the security by brute force attack.

3.2 Statistical analysis

An ideal cipher should be strong against any statistical attack, so statistical analysis on cipher-text is of crucial importance for a cryptosystem. In order to prove the security of the proposed image encryption scheme, the following statistical tests are performed.

3.2.1 Histogram Analysis

To prevent the access of information to attackers, it is important to ensure that encrypted and original images do not have any statistical similarities. The histogram analysis clarifies that pixel values of image are distributed. A number of images are encrypted by the encryption schemes under study and visual test is performed.

As shown in Fig. 2(a-f). The histogram of original image contains great sharp rises followed by sharp declines and the histograms of the encrypted images for different round have uniform distribution which is significantly different from original image and has no statistical similarity in appearance. So, the proposed algorithms do not provide any clue for statistical attack. The encrypted image histogram, approximated by a uniform distribution, is quite different from plain-image histogram.

3.2.2 Correlation Analysis

The correlation coefficient between original and cipher image of horizontal, vertically and diagonally is calculated in Table 3(a) and it is shown in Fig. 3

3.3 Key Space Analysis

Key space size is the total number of different keys that can be used in the cryptography. Cryptosystem is totally sensitive to all secret keys. A good encryption algorithm should not only be sensitive to the cipher key, but also the key space should be large enough to make brute-force attack infeasible. The key space size for initial conditions and control parameters is over than 2^{148}. Apparently, the key space is sufficient for reliable practical use.

3.4 Differential Analysis

In general, a desirable characteristic of an encrypted image is being sensitive to the little changes in a Plain - image (e.g. modifying only one pixel). Adversary can create a small change in the input image to observe changes in the result. By this method, the meaningful relationship between original image and cipher image can be found. If one little change in the plain-image can cause a significant change in the cipher-image, with respect to diffusion and confusion, then the differential attack actually loses its efficiency and becomes almost useless.

Tests have been performed on the encryption schemes on a 256-level color image of size 256×256 shown in Fig. 2(a-f). The MAE, NPCR and UACI experiment result is shown in Table 2(b) and Table 1. The Table 2(a) and table 3 compare the result of Yong's [6] work based on chaotic map and results obtained from NPCR shows that the sensitivity of encryption scheme for little changes in the input image is under 0.01%. According to the UACI estimation result, the rate influence due to one pixel change is very low. The results demonstrate that a swift change in the original image will result in a negligible change in the ciphered image.

3.5 Information entropy analysis

The proposed algorithm is secure against the entropy attack. The test result on different image for different round is defined in Table4.

3.6 Speed analysis

Apart from the security consideration, some other issues on image encryption are also important. This includes the encryption speed for real-time processes. In general, encryption speed is highly dependent on the CPU/MPU structure, RAM size, Operating System platform, the programming language and also on the compiler options. So, it is senseless to compare the encryption speeds of two ciphers image without using the same developing atmosphere and optimization techniques. Inspite of the mentioned difficulty, in order to show the effectiveness of the proposed image encryption algorithms, we evaluated the performance of encryption schemes with an un-optimized MATLAB7.0 code. Performance was measured on a machine with Intel core 2 Duo 2.00 GHz CPU with 2 GB of RAM running on Windows XP. The time for encryption and decryption is measured for different round is shown in Table (5-6).

3.7 FIPS 140 testing

We also show that our proposed algorithm pass the FIPS 140-2 randomness tests. There are four tests: Mono-bit, Poker, Runs tests and Long run tests. Each of the tests was designed to test the randomness of a sample sequence with the length of 20,000 bits as follows: Table7 shows the FIPS-140 test pass in different round for encryption.

3.8 Secured session key exchange using ECC

Session key length: 148 bits

Session key in Hex:
12EE9145C013ABA784B3A369C1EC777588CCA

Elliptic Curve Parameters: P = 751, a = -1, b = 188, Generator Point G = (297,569), $n_A = 13$, $n_B = 12$.

Followings are the encrypted coordinates of the ciphered session key : [(45, 97), (64, 738), (333, 435), (333, 435), (324, 7), (45, 97), (84, 613), (529, 254), (653, 422), (414, 88), (45, 97), (492, 167), (627, 59), (472, 336),(627, 59), (297, 569), (160, 140), (84, 613), (472, 336), (492, 167), (627, 59), (492, 167), (732, 180), (324, 7), (653, 422), (45, 97), (333, 435), (653, 422), (297, 569), (297, 569),(297, 569),(529, 254), (160, 140), (160, 140), (653, 422), (653, 422), (627, 59)] point on elliptic curve is depicted in fig. 4. The encryption/decryption time for the length of session key is depicted in fig. 5.

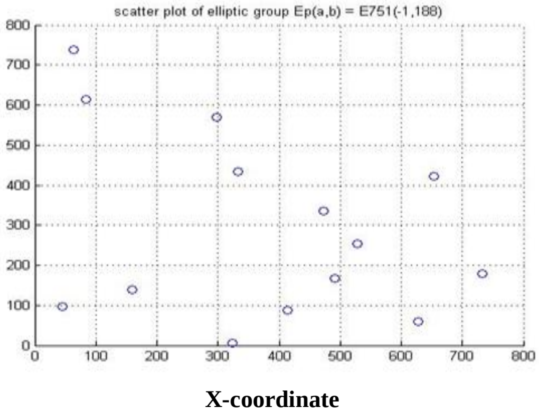

Fig 4. Point on Elliptic Curve

4. CONCLUSION AND FUTURE WORK

In this work we propose a technique which replaces the traditional preprocessing complex system and utilizes the basic operations like confusion, diffusion which provide same or better encryption using cascading of 3D standard and 3D cat map. We generate diffusion template using 3D standard map and rotate image by using vertically and horizontally red and green plane of the input image. We then shuffle the red, green, and blue plane by using 3D Cat map and standard map. Finally the Image is encrypted by performing XOR operation on the shuffled image and diffusion template.

The Elliptic Curve discrete logarithm problems are used for session key exchanged in secured manner. The Elliptic Curve discrete logarithm problem is considered harder than either the integer factorization problem or the discrete logarithm problem modulo p. The ECC has the highest strength-per-bit compared to other public key cryptosystems. ECC is useful in those applications where storage, bandwidth and processing capacity are limited. Completion of the design, both theoretical analyses and experimental tests have been carried out, both confirming that the new cipher possesses high security and fast encryption speed. In conclusion, therefore, the new cipher indeed has excellent potential for practical image encryption applications. In future this method are used in secured and fast transmission of video and paid TV channels.

5. REFERENCE

[1]. W. Diffie and M. E. Hellman, “New directions in cryptography”, IEEE Trans. Information Theory, IT-22(6), pp. 644–654, 1976.

[2]. R. L. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems”, Comm. ACM, 21(2), pp. 120–126, 1978.

[3] T. ElGamal, “A public key cryptosystem and a signature scheme based on discrete logarithms”. IEEE Trans. Inform. Theory, 31(4), pp. 469–472, 1985.

[4] FIPS “ Digital Signature Standard. Federal Information Processing Standards Publication”, 186-2, February 2000.

[5]. V. Shoup “Lower bounds for discrete logarithms and related problems”, In Advances in cryptology—EUROCRYPT ’97, volume 1233 of LNCS, pp. 256–266, 1997.

[6] Yong Wanga, Kwok-Wo Wong, Xiaofeng Liao, Guanrong Chen, “A new chaos-based fast image encryption algorithm”, Applied Soft Computing pp. 514–522, 2011 www.elsevier.com/locate/asoc.

[7]. Z. Guan, F. Huang, W. Guan, “Chaos-based image encryption algorithm”, Physics Letters A 346, pp. 153–157, 2005

[8]. S. Lian, J. Sun, Z. Wang, “A block cipher based on a suitable use of the chaotic standard map”, Chaos Solitons & Fractals 26, pp. 117–129, 2005

[9]. K.W. Wong, B.S. Kwok, W.S. Law, “A fast image encryption scheme based on chaotic standard map”, Physics Letters A 372, pp. 2645–2652, 2008

[10]. Y. Mao, G. Chen, S. Lian, “A novel fast image encryption scheme based on 3D chaotic baker maps”, International Journal of Bifurcation and Chaos 14, pp. 3613–3624, 2004.

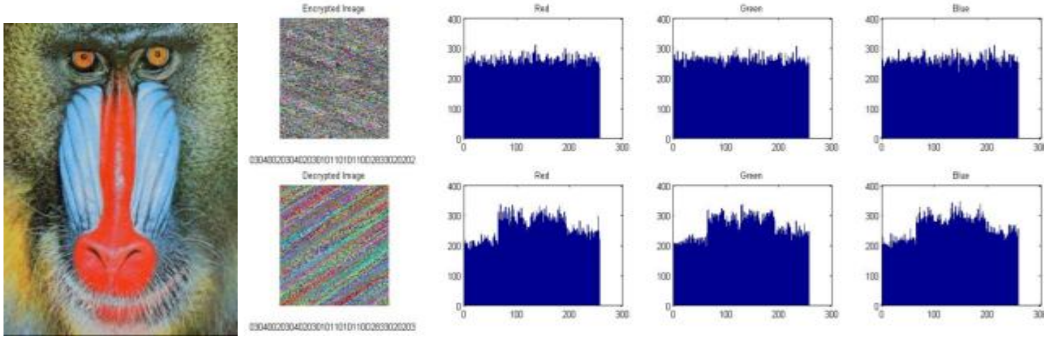

Fig.1(a) Input image Encrypted with 0304002030402 0301011010110D2833020202 and Decrypted by 03040020304020301011010110D2833020203

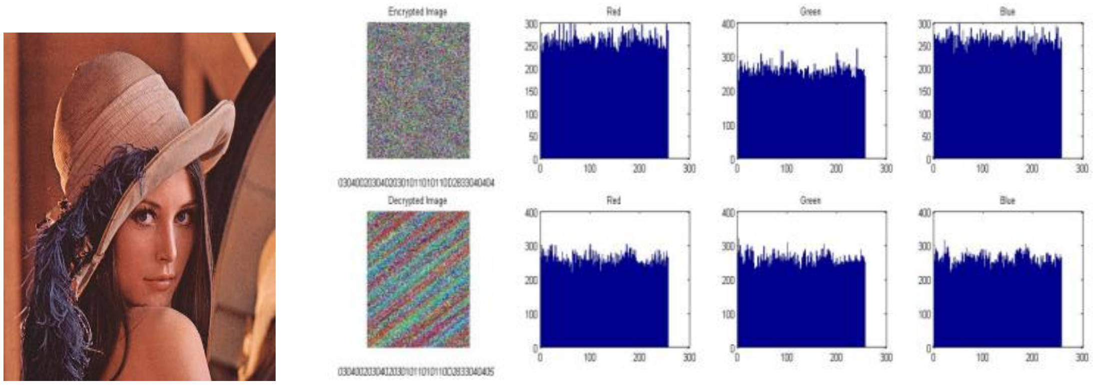

Fig.1 (b) Input Lenna Image Encrypted with 03040020304020301011010 110D2833040404 and Decrypted by 030400203040203010110101110D2833040405

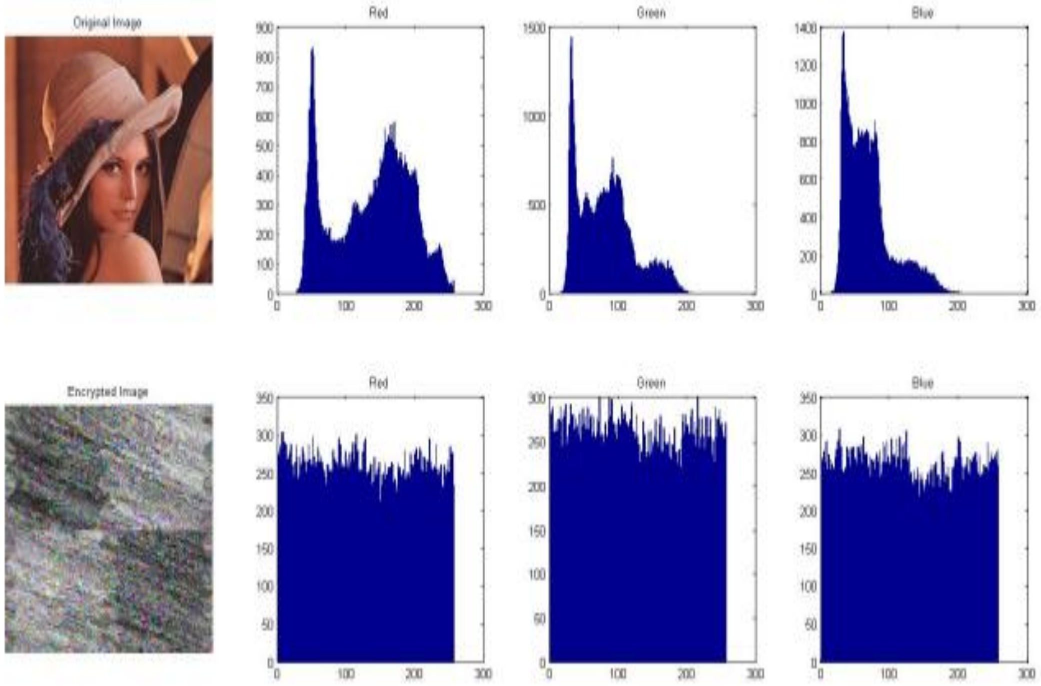

Fig.2 (a) Histogram for Red, green and Blue plane of original and encrypted image for R=1

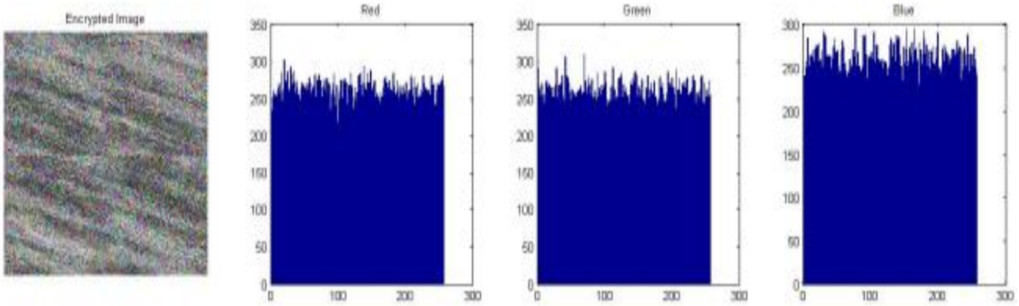

Fig.2 (b) Histogram for Red, green and Blue plane of encrypted image for R=2

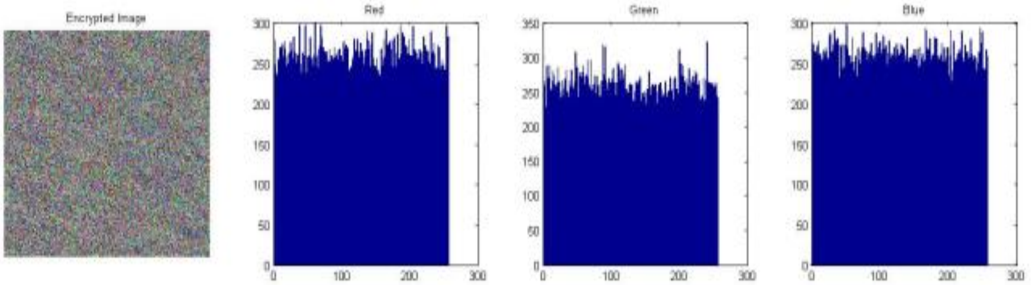

Fig.2 (c) Histogram for Red, green and Blue plane of encrypted image for R=4

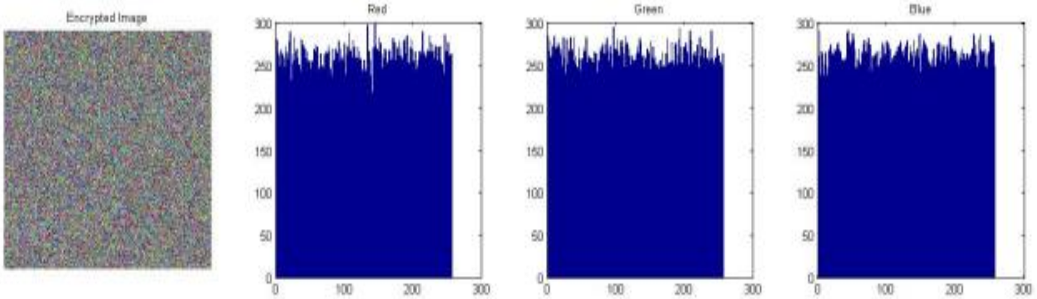

Fig.2 (d) Histogram for Red, green and Blue plane of encrypted image for R=8

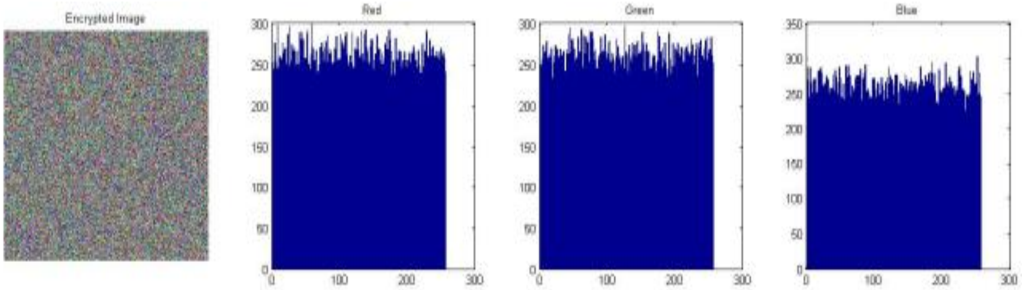

Fig.2 (e) Histogram for Red, green and Blue plane of encrypted image for R=16

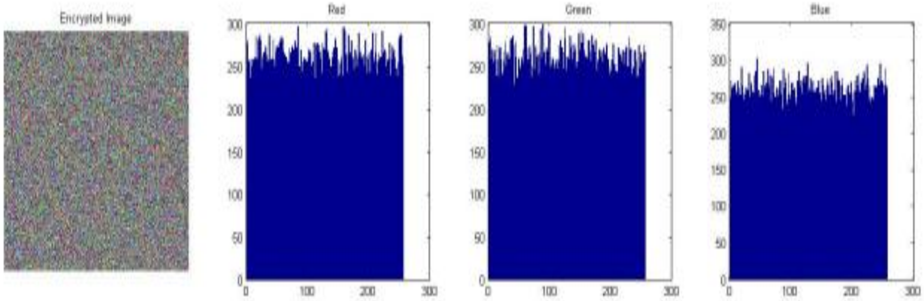

Fig.2 (f) Histogram for Red, green and Blue plane of encrypted image for R=32

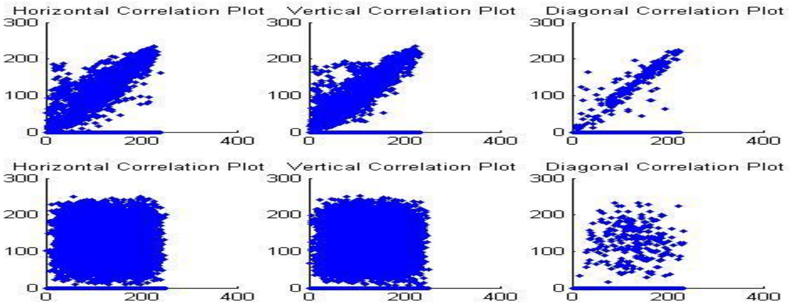

Fig.3 Correlation for horizontal, vertical and diagonal

Table1. NPCR, UACI and Entropy for key sensitivity test.

Lenna Error Image	**R=2**	**R=3**	**R=4**
NPCR	99.5966593424	99.6098836263	99.651082356
UACI	52.5394813687	51.6816741344	50.603535970
Entropy	7.99913068980	7.99912231127	7.9991865161
Baboon Error Image			
NPCR	99.6103922526	99.5905558268	99.599711100
UACI	46.9998348460	47.8581327550	48.719709807
Entropy	7.99901078968	7.99905756543	7.9991734549

Table2(a). Comparison of NPCR and UACI with Yong Wong et. al.[6]

Name of image		**R = 1**		**R = 2**		**R = 3**	
Airplane 256×256		**Our**	**Yong Wang et. al**	**Our**	**Yong Wang et. al**	**Our**	**Yong Wang et. al**
	NPCR	**99.62**	97.621	99.60	99.637	99.62	99.634
	UACI	**33.19**	32.909	33.10	33.575	33.35	33.580

Table2(b). NPCR and UACI for different round on different color image

Image 256×256		**R = 1**	**R = 2**	**R = 3**	**R = 4**	**R = 8**	**R = 10**	**R = 16**	**R = 32**
Baboon	NPCR	99.55	99.57	99.59	99.59	99.61	99.60	99.60	99.61
	UACI	37.17	38.68	39.00	38.78	38.69	38.88	39.03	38.89
	MAE	71.65	74.52	75.29	75.18	75.23	75.37	75.34	75.50
Lenna	NPCR	99.63	99.64	99.59	99.62	99.62	99.61	99.62	99.59
	UACI	28.87	27.51	27.33	27.42	27.43	27.64	27.51	27.37
	MAE	80.84	77.24	77.46	77.58	77.76	77.84	77.67	77.54

Pepper	NPCR	99.62	99.62	99.58	99.58	99.63	99.62	99.62	99.62
	UACI	38.05	38.26	37.99	38.03	38.34	38.20	38.33	38.11
	MAE	75.20	74.68	74.27	74.48	74.89	74.62	74.62	74.61
Airplane	NPCR	99.62	99.60	99.59	99.62	99.63	99.61	99.59	99.60
	UACI	33.19	33.10	33.35	33.27	33.329	33.28	33.32	33.31

Table3. The round number of scanning-image, permutation and diffusion to achieve NPCR and UACI

	NPCR	**UACI**	**No. of Round for Confusion and Diffusion**
Our	**>0.996**	**>0.287**	**1**
Ref.[6]	>0.996	>0.333	2
Ref.[7]	>0.996	>0.333	18
Ref.[8]	>0.996		5
Ref.[9]	>0.996	>0.333	6
Ref.[10]	>0.996	>0.333	6

Table 3(a). Correlation Coefficient for plain and cipher image

	Correlation coefficient of plain image			**Correlation coefficient of Cipher image**		
Images 256×256	**Horizontal**	**Vertical**	**Diagonal**	**Horizontal**	**Vertical**	**Diagonal**
Baboon	0.8646	0.8293	0.8114	0.004	0.007	0.037
Lena	0.9156	0.8808	0.8603	0.001	0.006	0.091
Pepper	0.9376	0.9364	0.8935	0.005	0.006	0.023

Table4. Entropy test for different color image

Image 256×256	**R = 1**	**R = 2**	**R = 3**	**R = 4**	**R = 8**	**R = 10**	**R = 16**	**R = 32**	**Yong Wang et. al [7]**
Our Scheme									
Baboon	7.9988	7.9990	7.9991	7.9992	7.9990	7.9990	7.9990	7.9991	-
Lenna	7.9981	7.9991	7.9991	7.9992	7.9991	7.9990	7.9990	7.9990	7.9990
Pepper	7.9987	7.9991	7.9991	7.9989	7.9989	7.9992	7.9992	7.9992	7.9990
Airplan	7.9989	7.9989	7.9991	7.9991	7.9991	7.9991	7.9991	7.9992	-
Boat	7.9986	7.9992	7.9991	7.9990	7.9990	7.9990	7.9991	7.9991	-

Table5. Encryption time in second for different round

Image 256×256	R = 1	R = 2	R = 3	R = 4	R = 8	R = 10	R = 16	R = 32
Our Scheme								
Baboon	0.510	0.87	1.20	1.56	2.94	3.65	5.73	11.24
Lenna	0.521	0.87	1.197	1.561	2.933	3.662	5.697	11.225
Pepper	0.521	0.87	1.197	1.561	2.933	3.662	5.697	11.225
Airplan	0.521	0.87	1.197	1.561	2.933	3.662	5.697	11.225
Boat	0.521	0.87	1.197	1.561	2.933	3.662	5.697	11.225

Table 5(a) Encryption time in second for different size and different round

Image	R = 1	R = 2	R = 3	R = 4	R = 8	R = 10	R = 16	R = 32
Our Scheme								
Lenna256×256	0.521	0.87	1.197	1.561	2.933	3.662	5.697	11.225
Lenna512×512	1.044	1.214	1.323	1.704	3.125	4.395	7.109	12.918
Lenna1024×1024	1.265	1.300	1.354	2.096	3.657	5.586	7.809	14.550

Table6. Decryption time in second for different round

Image 256×256	R = 1	R = 2	R = 3	R = 4	R = 8	R = 10	R = 16	R = 32
Our Scheme								
Baboon	0.43	0.77	1.12	1.470	2.85	3.55	5.62	11.17
Lenna	0.429	0.77	1.137	1.471	2.869	3.548	5.618	11.20
Pepper	0.430	0.78	1.139	1.472	2.869	3.549	5.619	11.22
Airplan	0.429	0.77	1.137	1.471	2.869	3.548	5.618	11.20
Boat	0.434	0.722	1.065	1.414	2.807	3.523	5.594	11.15

Table 7 FIPS-140 test P=pass, F= fail.

Name of image		R = 1	R = 2	R = 3	R = 4	R = 8	R = 10	R = 16	R = 32
Baboon 256×256	runs	10P,10P	11P,12P	13P,15P	12P,14P	15P,11P	12P,12P	16P,12P	17P,14P
	pocker	374.7F	9.8944P	15.443P	12.985P	13.644P	8.678P	12.556P	12.556P
	mono	10082P	9975P	9969P	9952P	9990P	10031P	9913P	10032P
Lenna 256×256	runs	10P,11P	13P,15P	13P,15P	17P,13P	12P,12P	13P,13P	15P,20P	12P,12P
	pocker	317.4F	24.30P	14.022P	8.9920P	18.227P	21.856P	11.558P	18.752P
	mono	9956P	10025P	10103P	9967P	9938P	10054P	9900P	10112P
Pepper 256×256	runs	12P,16P	13P,15P	12P,13P	12P,14P	12P,13P	13P,13P	13P,11P	14P,14P
	pocker	138.41F	13.196P	12.057P	13.337P	18.227P	20.454P	10.227P	27.929P
	mono	10085P	9982P	10001P	9967P	10048P	9994P	9913P	10036P

Airplan 256×256	runs	12P,10P	14P,13P	17P,23P	12P,14P	14P,13P	15P,12P	12P,13P	12P,11P
	pocker	880.25F	30.016P	13.504P	12.134P	16.761P	20.108P	12.800P	9.568P
	mono	10030P	10075P	9975P	9996P	9940P	10071P	9946P	9973P

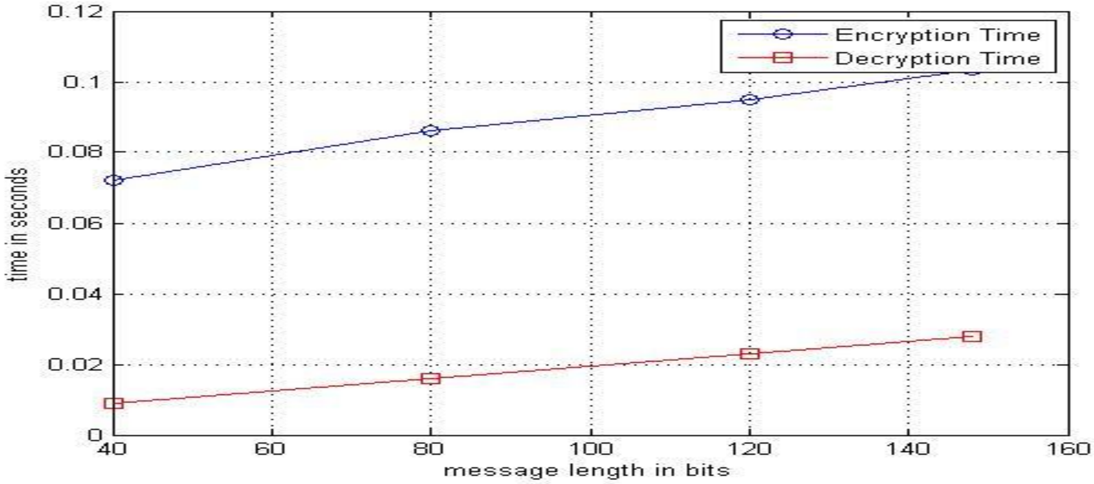

Fig. 5. Session key encryption/decryption time for different length.