3D Chaotic Functions for Image Encryption

Pawan N. Khade and Prof. Manish Narnaware

Department of Computer Science and Engineering
G.H. Raisoni College of Engineering, Nagpur

Summary

This paper proposes the chaotic encryption algorithm based on 3D logistic map, 3D Chebyshev map, and 3D, 2D Arnolds cat map for color image encryption. Here the 2D Arnolds cat map is used for image pixel scrambling and 3D Arnold's cat map is used for R, G, and B component substitution. 3D Chebyshev map is used for key generation and 3D logistic map is used for image scrambling. The use of 3D chaotic functions in the encryption algorithm provide more security by using the, shuffling and substitution to the encrypted image. The Chebyshev map is used for public key encryption and distribution of generated private keys.

Keywords: *logistic map, cat map, chebyshev map, chaos.*

1. Introduction

Large numbers of chaotic algorithms are being proposed for the image encryption now days. Chaotic functions are blessed with properties like sensitivity to the initial condition, and ergodicity which make them very desirable for encryption [1]. Many authors have proposed the image encryption algorithms based on low dimension chaotic functions. Security provided by these function is limited since these functions provide the limited key space and possesses some weaknesses [5]. 3 dimension functions are far more secure from cryptanalytic attacks [5]. In this paper we propose the enhanced 3D logistic map that is useful in encrypting the Red, Green, Blue component of the image separately. In section II we are going to discuss the methodology of the colour image encryption using 2D Arnolds cat map for image scrambling, 3D Chebyshev map for key generation, 3D logistic map for image encryption and 3D Arnolds cat map for R, G, and B substitution. The encryption algorithms make use of both diffusion and confusion mechanism which make it very secure. We provide the experimental results of our encryption algorithm.

2. Multidimensional Chaotic Functions

2D Arnolds Cat Map: The proposed image encryption algorithm makes use of the 2D Arnolds Cat Map which is used for scrambling the pixels of the colour image. The 2D Arnolds Cat Map is given by $\begin{bmatrix} x' \\ y' \end{bmatrix} \rightarrow \begin{bmatrix} 1 & p \\ q & pq+1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$ mod n[5]. Here n is dimension of image. Here $\begin{bmatrix} x \\ y \end{bmatrix}$ represents the original location of the image pixel and $\begin{bmatrix} x' \\ y' \end{bmatrix}$ represents the location of the pixel after the Arnolds Cat Map transformation. Here taking different values of p and q we will get variation of the Arnolds Cat Map[5]. Following is the algorithm of Arnolds Cat Map.

```
Start
        While there exist pixels
                Change location of pixel at (x,y)
                according to transformation.
        End loop
end
```

Fig1: Arnolds Cat Map algorithm.

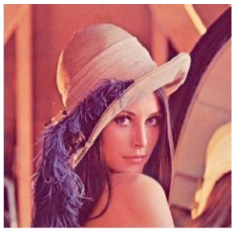


Fig2: Original image

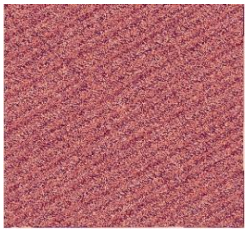


Fig3: Scrambled image

The use of Arnolds Cat Map provides extra security in the process of image encryption. Arnolds Cat Map does not change the intensity/colour composition of the image; it only shuffles the image data.

Chebyshev map: In the proposed algorithm Chebyshev polynomial is used to generate the private keys. Chebyshev polynomial $Tn(x)$ of the first kind is a polynomial in x of degree n, defined by the relation $\mathrm{Tn}(\mathrm{x}) = \cos \mathrm{n}\theta$ where $\mathrm{x} = \cos\theta$ [8]. Putting n=0, 1, 2, 3, 4 we get $\cos 0\theta = 1$, $\cos 1\theta = \cos\theta$, $\cos 2\theta = 2\cos^2\theta - 1$, $\cos 3\theta = 4\cos^3\theta - 3\cos\theta$, $\cos 4\theta = 8\cos^4\theta - 8\cos^2\theta + 1$.

Put cosθ = x we get $T0(x) = 1, T1(x) = x$, $T2(x) = 2x^2 - 1, T3(x) = 4x^3 - 3x$, $T4(x) = 8x^4 - 8x^2 + 1$. Here $T2(x), T3(x), T4(x)$, and so on polynomials exhibit the chaotic behavior hence we can use them to generate the random values for generating random keys [8]. The proposed algorithm uses the $T2(x), T3(y), T4(z)$ for generating private key separately for Red, Green, and Blue components of the image seperately.

3D Logistic Map: The logistic map is simplest chaos function and given by an equation $x_{n+1} = \lambda x_n(1 - x_n)$. For $0<x_n<1$ and λ=4 the equation exhibit the chaotic behavior.

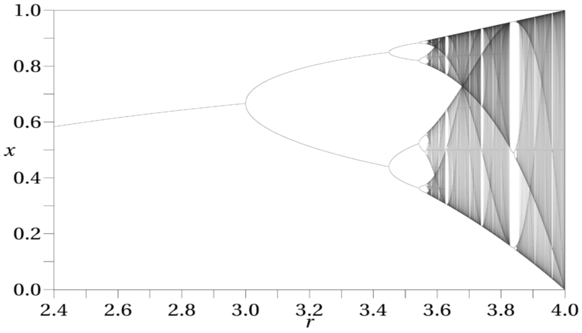


Fig4: Bifurcation diagram of logistic diagram.

Hongjuan Liu. et al proposed the 2D logistic map given by the following formula:

$x_{i+1} = \mu_1 x_i(1 - x_i) + \gamma_1 y_i^2$

$y_{i+1} = \mu_2 y_i(1 - y_i) + \gamma_2(x_i^2 + x_i y_i)$ [2].

The above formulas increase the quadratic coupling of the items $y_i^2, x_i^2, x_i y_i$ and provide the more security to the system. When 2.75 < μ_1<3.4, 2.7<μ_2 3.45, 0.15<γ_1< 0.21, and 0.13<γ_2<0.15, the system comes into chaotic state and can generate a chaotic sequence in the region (0,1][2]. In this paper we are extending the idea of the 2D Logistic map to 3 dimensions by using the following formula:

$x_{i+1} = \lambda x_i(1 - x_i) + \beta y_i^2 x_i + \alpha z_i^3$,

$y_{i+1} = \lambda y_i(1 - y_i) + \beta z_i^2 y_i + \alpha x_i^3$,

$z_{i+1} = \lambda z_i(1 - z_i) + \beta x_i^2 z_i + \alpha y_i^2$.

Here the above equations exhibit the chaotic behavior for 3.53<λ<3.81, 0<β<0.022, 0<α<0.015 and can take the value between [0, 1].

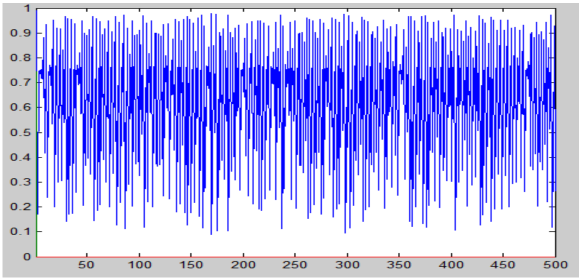


Fig5: Plot of X component of 3D logistic map.

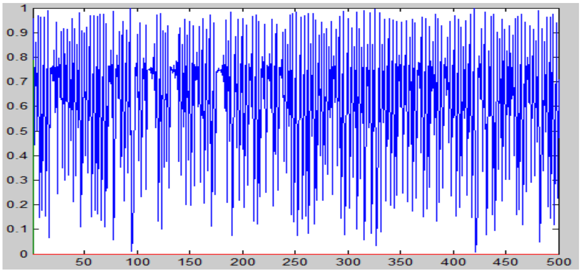


Fig6: Plot of Y component of 3D logistic map.

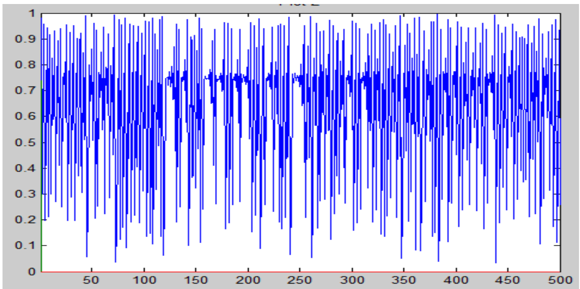


Fig7: Plot of Z component of 3D logistic map.

Presence of cubic, quadratic coupling and 3 constant terms make the 3D logistic map even more complicated and secure. In this algorithm we are going to use the 3D logistic map for encrypting the R, G, B component separately.

3D Arnolds Cat Map:

Many authors have suggested the improved 3D Arnolds Cat Map for image scrambling. **A)** One of it is suggested by Hongjuan Liu et al. which is improved by introducing two new control parameters c &d. Following is the enhanced ACM

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{pmatrix} 1 & a & 0 \\ b & ab+1 & 0 \\ c & d & 1 \end{pmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \bmod N$$ [2]. This 3D

ACM perform the dual encryption, firstly it performs the shuffling and secondly it performs the substitution. Using ACM the correlation among the adjacent pixels can be disturbed completely [2]. On the other hand, the 3D cat map can substitute grey/ colour values. We implemented 3D ACM on both colour and grayscale image and following are the results.

Fig8: Original image

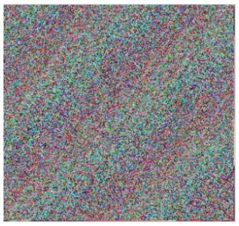


Fig9: Scrambled image

Figure 8 represents the colour image scrambled for 65 iterations of the 3D ACM.

Fig 10: Original image

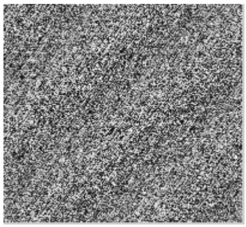


Fig 11: Scrambled image

The 3D ACM was implemented in two steps as follows:

$\begin{bmatrix} x' \\ y' \end{bmatrix}$ are the location of pixel after mapping and $\begin{bmatrix} x \\ y \end{bmatrix}$ are location of pixels before mapping. The third parameter inserted is z' which is given by z'= (c*x+d*y+z) mod M. Here z is the intensity/colour code of pixel before mapping and z' is intensity/colour code of image after mapping. M is the maximum value of intensity of pixel that is M=256. We first find x' and y' using ACM and then calculate z' using above equation. The 3D ACM is more secure than that of ACM because of two factors. First is presence of additional constants c and d that can take any random value. Secondly ACM can only shuffle the pixel location but 3D ACM can perform the additional substitution and make distribution of colour/gray value uniform.

3. Methodology

In this section we are going to discuss the proposed methodology. The proposed algorithm make use of the 2D Arnolds Cat Map for image scrambling, 3D Chebyshev map is used for private key generation, The 3D logistic map is used for encryption purpose and 3D Arnolds Cat map is used for additional scrambling and substitution. Following diagram indicate the proposed algorithm.

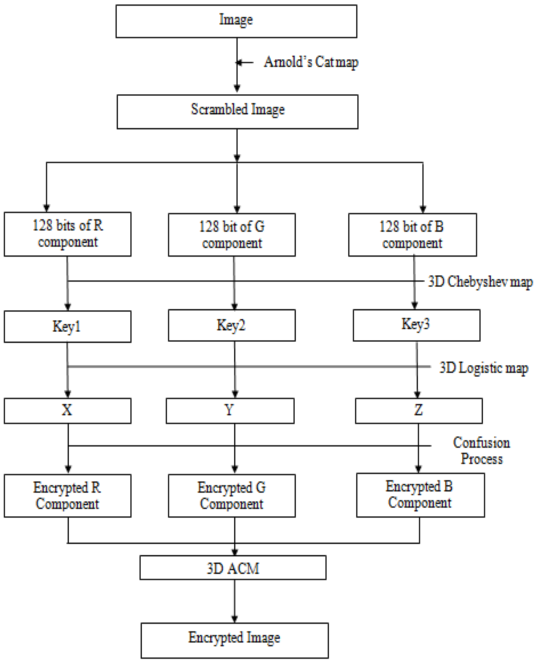


Fig12: Proposed methodology (Algorithm).

According to proposed algorithm we first apply the Arnolds Cat Map so that the image pixels will be scrambled. Scrambling provide additional security before encrypting the image. This scrambled image is used for key generation as follows; the 128 decimal bits of R, G, and B component are iterated in the 3D chebyshev map for 80 times to provide the 3 keys separately for Red, Green, and Blue component. Following figure clearly indicate the key generation mechanism.

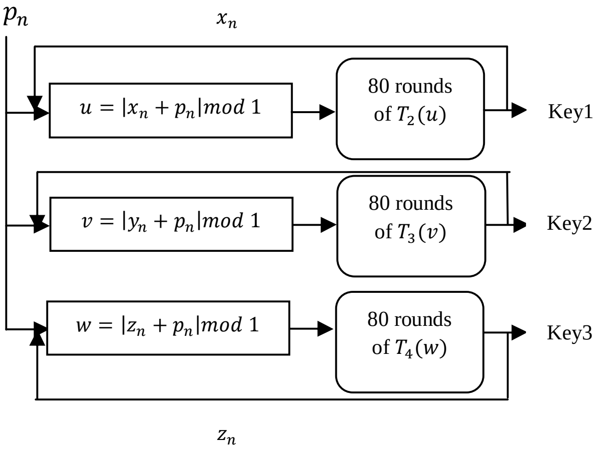
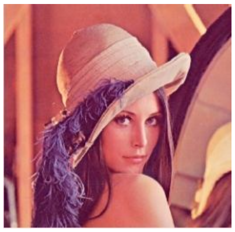


Fig13: Key generation using 3D Chebyshev map.

These keys will be fed to the 3D logistic maps for generating the X, Y, and Z component that will be used for XOR operation with the scrambled image. In this case the key1, key2, and key3 are provided as the initial input x0,y0, and z0 to the 3D logistic map and X, Y, Z components are generated seperately. In the confusion process the X component is XORed with Red component of the scrambled image, Y component is XORed with Green component of the scrambled image and same trnasformation is used for Blue component and at last all the components are combined into single image. The last step of the encryption process is the additional scrambling and substitution, which is provided using the 3D Arnolds Cat Map to give the encrypted image. The double shuffling and additional substitution provide the huge security to the encryption. In next section we are providing the experimental results those we got after performing proposed algorithm on colour image.

4. Process & Experimental Results

Following are the steps:
1) Accept the image
2) Apply Arnolds Cat Map transformation. Following are the results:



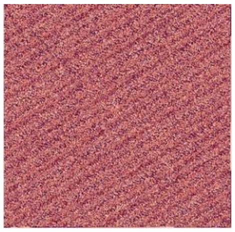
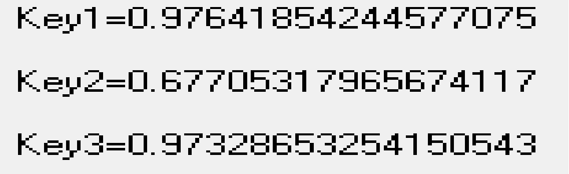


Fig14: Original image

Fig15: Scrambled image

3) Generate 3, 17 decimal bit random numbers.
4) Divide each random number with 10^{17} to get the fractional numbers R_{k1}, R_{k2}, R_{k3} keys.
5) Initialize Red(1)= R_{k1} , Green(1)= R_{k2} , and Blue(1)= R_{k3} and iterate in the 3D Chebyshev map along with 128 decimal bits of scrambled image to generate remaining values to get key_1, key_2, key_3.



```
Key1=0.97641854244577075

Key2=0.67705317965674117

Key3=0.97328653254150543
```

Fig16: Generated keys.

6) Initialize X (1) =key1, Y (1) =key2, Z (1) =key3 and iterate using the 3D logistic map to generate complete values of X, Y, and Z component.
7) R=Red XOR X, G=Green XOR Y, and B=Blue XOR Z and combine R, G, B to form an image pn. Finally Perform Dn = pn XOR Xn XOR Dn-1. Here X_0=0 and pn is plaintext (in this case XORed image).
8) Perform final scrambling with additional substitution using 3D Arnolds Cat Map.

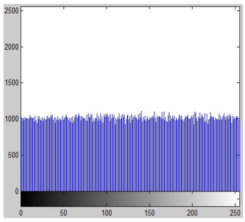
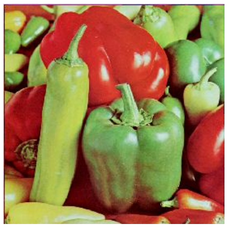


Fig17: Image after encryption.

Fig18: histogram of R, G, B

Figure 17 represent the images after final encryption and figure 18 indicate the histogram of Red, green, and blue colour after image encryption. Here we can see that Red, Green and Blue component of the image are distributed uniformly. The above result indicates that it will be very difficult for cryptanalyst to figure out the statistical relationship between original and encrypted image. In order to get the Original image we need to follow reverse procedure.



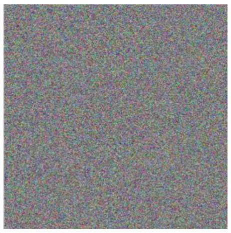
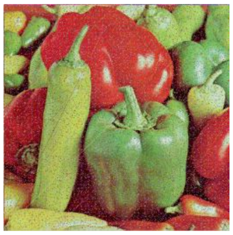


Fig19: Original image

Fig20: Encrypted image



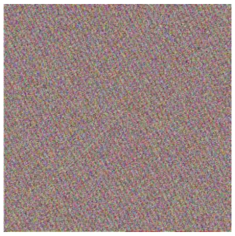


Fig21:Decrypted Image

Fig22: Image decrypted with wrong key

If the cryptanalyst will try to recover the image using slightly wrong key then he/she will get the result as given in figure 22. Moreover the Red, Green, Blue component of the image are uniformly distributed hence it is impossible for cryptanalyst to find any statistical relation between sample image that he/she might have.

IV A. Key Space:

If the key space of the system is high then only we can resist the cipher text attacks and cryptanalytic attacks. In case of proposed method Original 51 bit random decimal number is divided into 3 parts each of 17 bit number, which indicate that the possible key space of the proposed algorithm is 10^{51}which is practically huge key space to protect the system from brute force attack. This system uses the addition of three keys to determine the c and d constant for the 3D Arnolds Cat Map hence slight change in the value of the key will give different output. This make the system more secure.

5. Public Key Encryption

The encryption process creates multiple text files which contain the key and other security parameters like number of iterations and so on, and their secure communication with the other party is necessary. This can be provided by using the public key encryption of the generated text files. This paper discusses public key encryption using the Chebyshev map using El Gammel public key encryption for floating point numbers. The Chebyshev map possesses the semi group property which lead to TsTr= TrTs. Here A Chebyshev polynomial map Tp : R→R of degree p is defined using the following recurrent relation: Tp+1(x) = 2xTp(x)−Tp−1(x)[8]. Which can be implemented using iterative method; following is the algorithm to implement this Chebyshev map.

```
Start
        loop p>0
                num3=2*x*num2-num1
                p=p-1
                temp=num2
                num2=num3
                num1=temp
        end loop
end function
```

Though the value of TsTr is not exactly similar to TrTs but it is same up to the 7 digits after decimal place, which might impose certain level of restriction in security of the system.

Algorithm:

Alice, in order to generate the keys, does the following:

1. Generates a large integer s.
2. Selects a random number x ∈[−1,1] and computes Ts(x).
3. Alice sets her public key to (x,Ts(x)) and her private key to s.

Encryption Algorithm: Encryption requires five steps:

Bob, in order to encrypt a message, does the following:

1. Obtain Alice's authentic public key (x,Ts(x)).
2. Represents the message as a number M ∈[−1,1].
3. Generates a large integer r.
4. Computes Tr(x),Tr·s(x) = Tr(Ts(x)) and X = M· Tr·s(x).
5. Sends the ciphertext C = (Tr(x),X) to Alice.

Decryption Algorithm: Decryption requires two steps:

Alice, to recover the plaintext M from the ciphertext C, does the following:

1. Uses her private key s to compute Ts·r = Ts(Tr(x)).
2. Recovers M by computing M = X/Ts·r(x) [8][11][10].

6. Conclusion

This paper discusses the method for image encryption using 3D Chaotic maps. The method consists of shuffling, substitution and confusion and diffusion mechanism which make it secure. Key size and 3D chaotic maps provide the additional security to the proposed system. The use of 3D Logistics map provide the higher key sensitivity so that making slight change in the initial condition will make the large change in the encrypted result and consecutively the decrypted image.

References

[1] Bose, R. and Banerjee, A., "Implementing Symmetric Cryptography Using Chaos Functions", Advanced Computing & Communication Conference, 1999.

[2] Hongjuan Liu, Zhiliang Zhu, Huiyan Jiang, Beilei Wang. "A Novel Image Encryption Algorithm Based on Improved 3D Chaotic Cat

Map", The 9th International Conference for Young Computer Scientists, 2008.

[3] Mao-Yu Huang, Yueh-Min Huang, Ming-Shi Wang. **"**Image encryption algorithm based on chaotic maps**",** Computer Symposium (ICS), 2010.

[4] Zhao Mingming, Tong Xiaojun. "A Multiple Chaotic Encryption Scheme for Image",6th International Conference on Wireless Communications Networking and Mobile,Computing (WiCOM), 2010.

[5] Zhou Zhe, Yang Haibing, Zhu Yu, Pan Wenjie, Zhang Yunpeng. "A Block Encryption Scheme Based on 3D Chaotic Arnold Maps", International Asia Symposium on Intelligent Interaction and Affective Computing, 2009.

[6] A .Senthil Arumuga1, D. Kiruba Jothi. "Image Encryption Algorithm Based On Improved 3d Chaotic Cat Map", 2010 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC).

[7] Min Long; Li Tan; "A Chaos-Based Data Encryption Algorithm for Image/Video ", Second International Conference on Multimedia and Information Technology (MMIT), 2010.

[8] Ljupco Kocarav , Shingao Lian; "Chaos-based Cryptography theory algorithms and applications" , Studies in Computational Intelligence,Volume 354,2011.

[9] Prasadh, K.; Ramar, K.; Araman, R.G.; "Public Key Cryptosystems Based on Chaotic-Chebyshev Polynomials", International Conference on Advances in Recent Technologies in Communication and Computing, 2009. ARTCom '09.

[10] Ganesan, K.; Singh, I.; Narain, M.; "Public Key Encryption of Images and Videos in Real Time Using Chebyshev Maps ",Fifth International Conference on Computer Graphics, Imaging and Visualisation, 2008. CGIV '08.

[11] Kocarev, L.; Tasev, Z.; "Public-key encryption based on Chebyshev maps "Proceedings of the 2003 International Symposium on Circuits and Systems, 2003. ISCAS '03.

[12] Bergamo, P.; D'Arco, P.; De Santis, A.; Kocarev, L.; "security of public-key cryptosystems based on chebyshev polynomials", IEEE Transactions on Circuits and Systems I: Regular Papers 2005.