**DNA Computing Based Image Encryption Algorithm for PHM**

By

Sulabh Kumar Jain,19BCE1384

Vaibhavi Lohani,19BCE1513

***A project report submitted to***

**Dr. VIJAYAKUMAR PEROUMAL**

**Associate Professor, School of Electronics Engineering**

in partial fulfilment of the requirements for the course of

**CSE3501-Informtion Security Analysis and Audit**

in

**B.TECH SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**



**Vellore Institute of Technology, Chennai**

**Vandalur – Kelambakkam Road**

**Chennai – 600127**

**JUNE 2022**



**Chennai**

**BONAFIDE CERTIFICATE**

Certified that this project report entitled “**DNA Computing Based Image Encryption Algorithm for PHM”** is a bonafide work of Vaibhavi Lohani,19BCE1513 and Sulabh Kumar Jain,19BCE1384 who carried out the project work under my supervision and guidance.

***Guide Signature***

**Abstract** :

There has been a surge in the number of users surfing internet, and thereby generating more and more data. These data can be a plain text, an image file, or a sensitive video, audio file. These data can hold highly sensitive information, and should be protected at all cost. It has been seen that images are encrypted using the traditional hashing algorithm, which have been cracked over the last few years. In order to make image encryption more secure, we have come up with a solution to use a modified chaotic map to generate a random key, rather than the conventional hashing algorithms such as the SHA-256. The resultant will then be converted to DNA strands using the DNA computing technology, after passing through Linear Feedback Right shift algorithm. This will make the image more secure, and the image will take-up relatively lesser space to be stored in the database. We will use this methodology to store the patient health data in the database.

Keyword: DNA computing; Image Encryption; Patient Health; Linear Feedback Right shift algorithm; chaotic map; random key

1. **Introduction**

1.1 Preamble

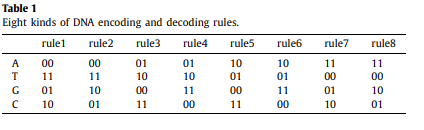
Through our project, we propose to provide a system which will use chaotic nature of our function, along with Linear Feedback right shift algorithm, and DNA computation to make the highly sensitive Healthcare Images of patients more secure. DNA computation will create a strand that can store massive information just like the DNA in human body, which has the capacity to store massive amount of information.

Year after year, data breaches are becoming an increasingly critical issue for the healthcare industry. Nearly 32 million records have been exposed through June of 2019 alone, more than double the number for 2018. As reported by many practitioners, from 2005 to 2019, the total number of individuals affected by healthcare data breaches was 249.09 million. Out of these, 157.40 million individuals were affected in the last five years alone. In the year 2018, the number of data breaches reported was 2216 from 65 countries. Out of these, the healthcare industry faced 536 breaches. This implies that the healthcare industry has faced the highest number of breaches among all industries.

The aim of our project is to propose a system which will not only make the images more secure, but will also reduce the space needed to store the images in the database. Generation of key to encrypt the image using hashing algorithm is slowly becoming obsolete, as researches have come up with solutions to crack these encrypted hashed-images in a limited time. Such hashing techniques include SHA-256, MD5, etc. These hashing algorithms works on the principle that an image is passed onto a hashing function, which yield an encrypted image. In our proposed system, we have used a chaotic map, that outputs a random number for a small change in the input provided. A key sequence will be generated using the provided function which is then converted to the Binary sequence. Meanwhile, we have taken an initial random Binary seed, and have applied Linear Feedback right shift algorithm on the binary seed taken. This yield a sequence of binary numbers (2^n - 1) which will be XOR with the binary sequence obtained from chaotic map, to generate a new key. Finally, we convert all the pixels of the original image into binary form, and perform XOR operation between each pixel and the last key generated from the above step. We will get an encrypted cipher image as a result. Now, we will convert the pixels from the encrypted image into a DNA strand consisting A,T,C,G characters only. A novel DNA computing-based Image Encryption algorithm is proposed to store the patient health data.

1.2 DNA computing

Deoxyribonucleic Acid , which is a short form of DNA is a biological matter that is found in almost all living beings. DNA molecules are essential for the growth and development of species. DNA molecules have the ability to store, process and transmit data. They have the capacity to store very he amounts of information, low power consumption and parallelism. Each DNA sequence contains 4 nucleic acid bases, namely, A (adenine), C (cytosine), G (guanine) and T(thymine). There are some DNA complementary rules defined on these 4 nucleic acid bases. According to these rules, DNA bases in each pair must be complementary to each other, like A and T are complementary, C and G are complementary. Like in a binary system, 0 and 1 are complementary, so 00 and 11 are complementary, 01 and 10 are also complementary. There are total of 4! = 24 encoding rules, but only 8 encoding rules comply with the complementary rules, as shown in Table 1. For 8-bit grayscale image, the pixel values ranging from 0 to 255, each pixel can be represented in 8-bits,so each pixel value can be encoded into a DNA sequence whose length is 4. For example, if a pixel value is 189, it is converted to a binary sequence [10111101], so if we encode it using the DNA encoding rule 1 in Table 1, we can obtain the DNA sequence [CTTG]. And by using the same using encoding rule 1 to decode this DNA sequence [CTTG], we can get a binary sequence [10111101]. But if we use another decoding rule (for example, rule 3) to decode the same DNA sequence [CTTG], we will get incorrect binary sequence (for example, [11101000]).In other words, the same rule should be used for encoding and decoding to get the correct pixel value of the image.



1.3 Image Encryption Algorithm

Image encryption is a technique which converts the original image to another form that is difficult to understand and completely hides the visual of the original image. Nobody can know the content of the original image without knowing what is the decryption key of the cipher image. Image encryption has many applications in the corporate world, health care, military operations, and multimedia systems. Encryption is the process of encoding plain text message into cipher text message whereas the reverse process of transforming cipher text back to plain text is known as decryption. Cryptography consists of encryption and decryption techniques. In this paper, the image encryption technique that we use is DNA encoding with Logistic Chaotic Map.

Contribution of the author

* To implement the DNA computing based Image encryption algorithm
* Involving Logistic Chaotic Map
* Using DNA Encoding rules
* Measure the results using Histogram Analysis, Root Mean Square Value, PSNR Value, Correlation Coefficient value Analysis, etc. to prove the efficiency of the algorithm.

Paper organization

This paper is organized as follows to achieve the objection of the research work : section I gives the introduction about the DNA computing, need / importance of the project, image encryption. Section II shows the related works carried out by the researchers. Section III give the detailed information about the proposed DNA computing based image encryption algorithm,

**II. Related Work**

T.Y. Wu et al. [1] proposed an image encryption algorithm which is based on DNA coding and hyperchaotic map which can be applied to cloud CCTV system. It uses the chaotic property of the hyperchaotic map in addition to DNA computing. Here we will record the images using the IoT camera and then encrypt using our image encryption algorithm and then send it to a cloud server for storage. This algorithm can handle 2 problems very effectively:1) The encryption algorithm has a low computation cost to continuously encrypt digital content at lightweighted IoT camera and 2)It has a procedure which can securely store those encrypted content in a low storage cost by utilizing DNA computing technique. DNA computing is very useful in storing large amounts of data and achieving parallelization by duplicating the large volume of DNA strands. The IoT camera captures and encrypts the image and this encrypted content will be stored in server which is present in cloud. Here a hyperchaotic map is used, that generates key stream which scrambles the plain image. The scrambled image is then encoded by using DNA techniques and then executed by a diffusion process. After getting processed by another DNA coding scheme it outputs an encrypted cipher image. In the security and performance analysis, this proposed algorithm has been compared against recent proposed image encryption algorithms using histogram analysis, correlation analysis, entropy analysis, differential attacks, and encryption time analysis . This proposed algorithm is found to have both security and efficiency.

H. Dongming et al. [2] proposed an image encryption scheme that uses DNA and DPRE(double random phase encoding) technique. In this algorithm, the DNA rules are randomly determined by the piecewise linear chaotic map, long for PWLCM. Different plaintexts will use different start values of PWLCM, and these start values are generated by Message Digest Algorithm 5 (MD5). Firstly PWLCM(piecewise linear chaotic map) is used to make 2 key images and 2 RPMs(Random Phase masks) that will be used in encryption process. Secondly, the plain image and the key image 1 are encoded by rows respectively with DNA rules that are picked from the eight DNA rules. Next the algorithm converts two DNA sequences into binary sequences according to the decoding rules, after that, these two sequences are used as the input to the optical XOR gate to obtain intermediate image 1. Then key image 2 is encoded by columns according to DNA rules selected from the 8 DNA rules. The DNA sequences of intermediate image1 and encoded key image 2 are transformed into binary form and then the XOR function is applied to them to get intermediate image 2. Finally, the intermediate image 2 is re-encrypted by DRPE technique to form an ultimate cipher image. Numerical simulation and attack analysis on the encrypted image are implemented to show the security and efficiency of the proposed approach.

T. Wangetal. et al. [3] proposed a 6-dimensional super-chaos image encryption algorithm based on bit-level permutation and DNA encoding. First, the proposed algorithm counteracts the traditional method of decoding low-dimensional chaos maps using a 6-dimensional super-chaos system. At the same time, the chaos sequence generated by the chaos system is associated with the properties of the original image. Therefore, different original images can result in completely different chaos sequences. This disables the selected plaintext and ciphertext attacks. Then a bit-level sequence is used. Then use DNA coding and manipulation to encode the image. The combination of bit-level ordering and DNA encoding can improve the security of cryptographic systems. Finally, DNA decoding with various rules is used to get the encrypted image. Compared to existing hyperchaos-based image encryption algorithms, the proposed algorithm is more secure because it uses bit-level ordering, DNA coding, and a more complex hyperchaos system. We have conducted many experiments including key space analysis, key sensitivity analysis, histogram analysis, correlation analysis, difference analysis, robustness analysis, etc. and show that the proposed algorithm is safe and reliable for image encryption.

E. Satiretal. et al. [4] proposed a new cryptographic method by combining DNA carrier media, DNA coding and DNAXOR algorithms with the Feistel network structure. The proposed DNA encryption process is digitally and biologically integrated into the designed bioengineering hardware. To confuse the data, DNA coding and DNA XOR have been integrated into the Feistel network. Because DNA can contain very large amounts of data (powers of 10 to 8 terabytes), this approach uses carrier media instead of using traditional multimedia carriers such as images and text. We have succeeded in increasing the number of hidden bits using DNA as. , Video in encryption, etc. Experimental results have shown that the proposed studies provide efficient results in terms of volume, brute force attack, keyspace, and entropy analysis. However, the points that can be improved are as follows. It may be based on an asymmetric encryption process. Also, additional techniques are needed to speed up the entire process, taking into account processing time and the speed of synthetic DNA, rather than silicon media. Dynamic encoding of each character is another requirement to increase the robustness of the algorithm. However, we should design without occupying memory or slowing down this process.

M.A. Malik et al. [5] proposed an image encryption algorithm based on ultra-chaotic dynamic systems and DNA computing. It's a four-dimensional chaotic dynamic system. In addition, CDSVSP is used to generate multiple keystreams from the same chaotic data obtained through iterations of a chaotic dynamical system. As a result, the size of the chaotic data used in the proposed cryptographic schemes is much smaller, even less than one-tenth that used by existing cryptographic schemes. To maximize the potential of DNA computing, various sequences of pseudo-random numbers are generated based on the keystream. These sequences are used as rules for DNA manipulation and conversion of decimal pixel values ​​to DNA and vice versa. This greatly improves the unpredictability and randomness of the resulting DNA sequence or cryptographic image. In order to properly propagate a simple image, it is first diffused using a masked image (keystream) at the decimal level, and then at the DNA level, a variety of DNA manipulations through the three channels of red, green, and blue. Mix randomly with any combination. To properly confuse a simple image, the three channels are combined as a one-dimensional array and then sorted so that the pixel values ​​are randomly distributed across all three channels. To ensure that the high security standards of the proposed new image encryption scheme are maintained, simple image pixel values ​​are included in every step of the encryption algorithm. H. Key stream (pseudo-random) generation, DNA conversion, DNA manipulation, and decimal conversion. Simple images and small changes to the keyset, coupled with large keyspaces and ultra-chaotic data that are very sensitive to private keys, produce completely different cryptographic images, making cryptographic schemes more resistant to known attacks. It will be safe.

H. Nematzadeh et al. [6] proposed a new hybrid method that combines the power of DNA with the randomness of a binary search tree (BST) to achieve a more accurate encryption method. To do this, first convert a simple image into a DNA image using the binary coding rules for DNA. Next, a BST is created from the DNA key. Each time, the BST route is randomly assigned to one of the pixels of the DNA image, and the remaining BST nodes are assigned the route corresponding to the DNA image. This process continues until all DNA image pixels are covered at least once by the BST node due to the encryption process. DNA image pixels and their associated BST nodes form a cryptographic image. Here, the color image is first converted to a gray tones and then into a DNA sequence to form the DNA image. Similarly, BST keys generated using the logistic mapping feature are also converted to the relevant DNA sequence. Finally, the DNA image and DNABST build the cryptographic image. This is done by repeatedly overlaying the DNA BST on the DNA image until every pixel in the DNA image is covered with the BST node at least once to apply the XOR operation.

X. Zhu et al. [7] proposed a system that uses the Kronecker product (indicated by ⨂) operation on two matrices of any size. The result is a selectable size and a better block matrix for encoding images of different sizes. In this article, we used the Chenhyperchaotic system to generate the keys. Two new Kronecker product matrices on a finite field have been created to scramble and scatter images of any size at the same time. The matrix was not only symmetric, but also reversible. Finally, DNA encryption is performed on the image.

Q. Cun et al. [8] has changed the commonly used chaotic map to something a little more chaotic. In this paper, we used a 1D chaos map to analyze the chaos of the equations using Lyapunov exponential analysis, bifurcation diagrams, and the NIST test. The image threshold is set. The final binary sequence is calculated and above the threshold, the center pixel of the original image is retained to form the selected area of ​​the original image. If the pixel value is less than the threshold, it will be converted to 0. Finally, image DNA coding is performed using an essentially dynamic pseudo-random number generator.

X. Wanga et al. [9] took a pure text image as input. This image is passed to SHA256, which hashes the image and produces 64-bit values ​​in the form of an array. Some initial values ​​are taken along with the sum of all the pixels in the image. Hexadecimal is converted to decimal, so you can perform XOR and modulo operations. Finally, the image pixels are encrypted with DNA encoding using the Hamming distance algorithm.

Y. Zhanga et al. [10] suggested generating the key in two parts. The first part of the key is generated by SHA256. The second half of the image is created with chaos mapping. The image pixels are then scrambled using the Arnold transform and the phase-reduced fractional Fourier transform. Then we performed a DNA XOR operation to spread the image. Finally, DNA was added to increase the security of the image.

**III Proposed DNA Computing based Image Encryption Algorithm:**

Using logistic map function, a sequence is generated(K1,i),this (K1,i) is multiplied with a factor of 255.Then using Linear Feedback Right Shift Register(LFRSR) another sequence called K2,i is generated using a starting seed value. These 2 sequences are then XORed to get the key. After this the sequence is XORed with our 8-bit image pixel to get the cipher. This is then transformed into DNA sequence using DNA encoding.

***Encryption:***

Image Conversion:

Our algorithm will first covert the M\*N size image into binary sequences of 8 bit per pixel image matrix, where M, N = 1,2,3,…..n. This image matrix can be called as Imat. The values of the image matrix range from 0-255, represented in binary.

Generating Keys:

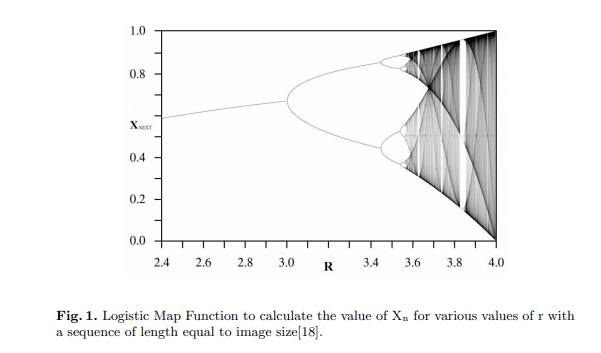
*Chaotic Logistic map:*

Chaotic Logistic Map generates a pseudo random series using some starting parameters to give values within the domain 0-1.The function for Logistic Chaotic map is given as following equation:

Xn+1 = rXn(1-Xn) (1)

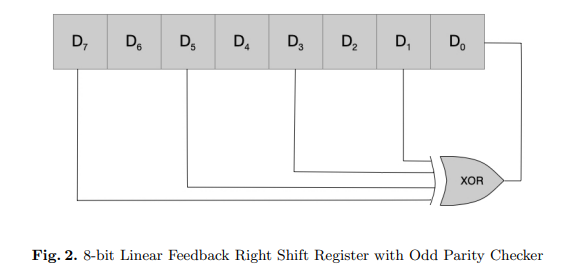
Where r is the bifurcating parameter, whose values ranges between 0-4 and it determines the randomness of the key. The starting value of X0 is calculated using the random function of the python library by setting a range between 0-1. The equation keeps iterating based on the image size. On estimation, it was found that the value of r between 3.5 to 3.99(which depend on the image size) gave high randomness. So we generate random values of r between 35 to 3.99. The figure below shows the random key generation space. The chaotic sequence that is generated is converted into positive unsigned integers by multiplying with 255 and then rounding off to the nearest integer. This integer sequence is then converted into 8-bit binary form matrix. So our unsigned key sequence as explain above can be calculated as:

K1,i = *Round(*Xi \* 255) (2)

 **Fig.1.**Logistic Map Function for calculating the value of Xn for various values of r with a sequence of length equal to image size.

*8-bit LFRSR:*

Initially an 8-bit register is taken, which we will initialize with any random 8-digit binary number. All the odd bits of the LFRSR is connected to an XOR. We will XOR all the numbers present in the odd position, which will yield us an output for this instance. We will right-shift the binary number present in the LSFR. As a result of the above step, the most significant bit of the LFRSR will become free. We will store the value produced from the above step in the most significant bit of the LFRSR after performing right shift on it. When we perform the similar operation recursively on the LFRSR, we find that the LFRSR is producing random number in every step. The original number that we had taken in the LFRSR will be repeated after 2^n-1 times, thereby making it very difficult to guess. From the above steps, we will choose a random LFRSR number, which we will use to XOR with each pixels of the image.



The 2 generated key sequences K1,i(generated from logistic chaotic map) and K2,i(generated from LFRSR) are converted into binary form . Then they are XORed to generate a key Ki for encrypting our original image.

Ki= K1,i XOR K2,i

Generating Cipher Matrix:

The generated key Ki  is XORed with Imat (Image Matrix) which gives a sequence of binary fashion. After that, each 8-bit binary sequence is converted into positive integer values. All these values will be the pixel values of the cipher image Ci.

Creation of Cipher Image:

To see the cipher image, the pixel values of the cipher image are organised as a cipher matrix and then this matrix is converted to cipher image. As we see this cipher image, we realize that how the pixels from the original image were confused and randomized so we get a cipher image that gives no clue of what the original image could be.

DNA Encoding:

The sequences of binary fashion that we received from “Generating Cipher Matrix” section are converted into DNA sequences using DNA encoding mechanism. We apply the rules shown in Table 1. We apply all the rules to our binary sequences in a circular fashion. Ex: to encode first 2 bits, we use Rule 1, then to encode the next 2 bits, we use Rule 2 and so on. This DNA sequence obtained is stored into a text file.

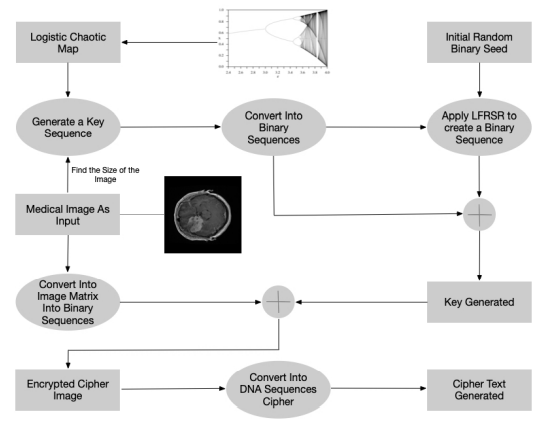
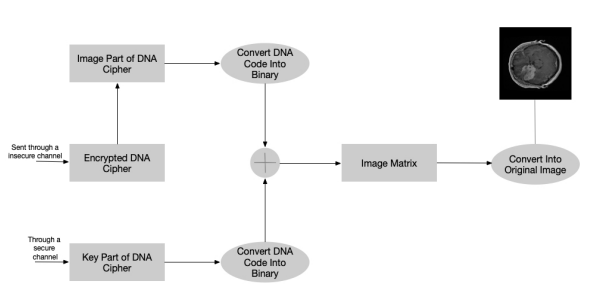


Fig: Block Diagram for Encryption Model

***Decryption:***

We take the encrypted DNA cipher and extract the image part of the DNA cipher. We first convert the DNA code to binary numbers. Simultaneously, we will convert the key part of the DNA cipher into binary digits. We will take the two binary numbers and pass them through an XOR operation which will result the image matrix. Now, all we need to do is convert this image matrix into an image, which will be in a greyscale form.



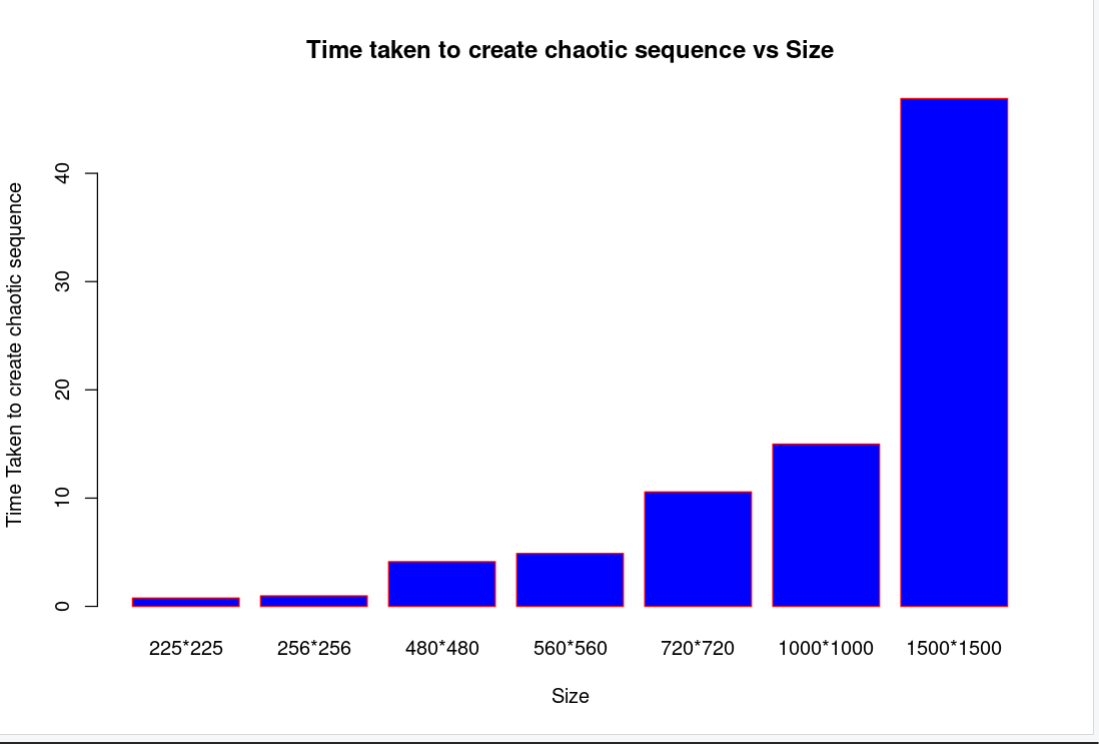
**Results and Discussion:**

We carried out the image encryption experiment using the proposed algorithm, using Python 3.10.5 ,with the system configuration of Windows 11 with 4 GB RAM and Intel Core i5 processor.

The various parameters used while analysing the results are:

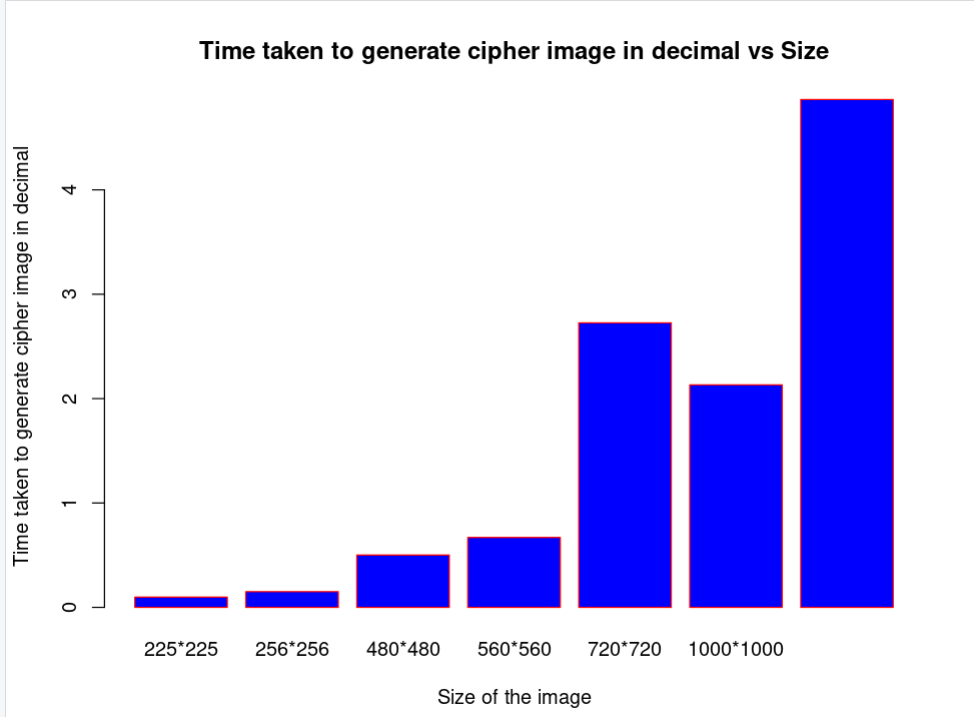
* Size: refers to the size of the image used.
* Time taken to create chaotic sequence: Time required to create the chaotic sequence Xi using Chaotic Logistic Map function.
* Time taken to generate cipher image in decimal: To ensure that the cipher image doesn’t carry any visual cue of the original image, we convert the cipher image (binary)matrix to decimal matrix and then convert that matrix to cipher image.
* Time taken for logistic map encryption of the image: Time taken to encrypt the image by using the key generated from Chaotic Logistic map and XORing it with our original image matrix.
* Time taken to generate cipher image in binary: To decrypt the image, we first need to generate a binary matrix of the cipher image and then XOR that binary cipher image matrix with the key that was passed through the secure channel.
* Time taken for DNA encoding: It is the time taken to convert the resultant binary sequences into DNA strands by using all the 8 rules (fig. 1),each rule applied for two bits at a time in a circular fashion.
* Time taken to perform LFSR of the image: Time taken to encrypt the image by using the key generated from LFSR (K2).
* Time taken to create chaotic sequence of the image Vs. Size

We understand that chaotic sequence values lie in the domain 0-1 and are generated by Chaotic Logistic Map using some initial parameters. As the image size increases, the time required to create its chaotic sequence also increases.



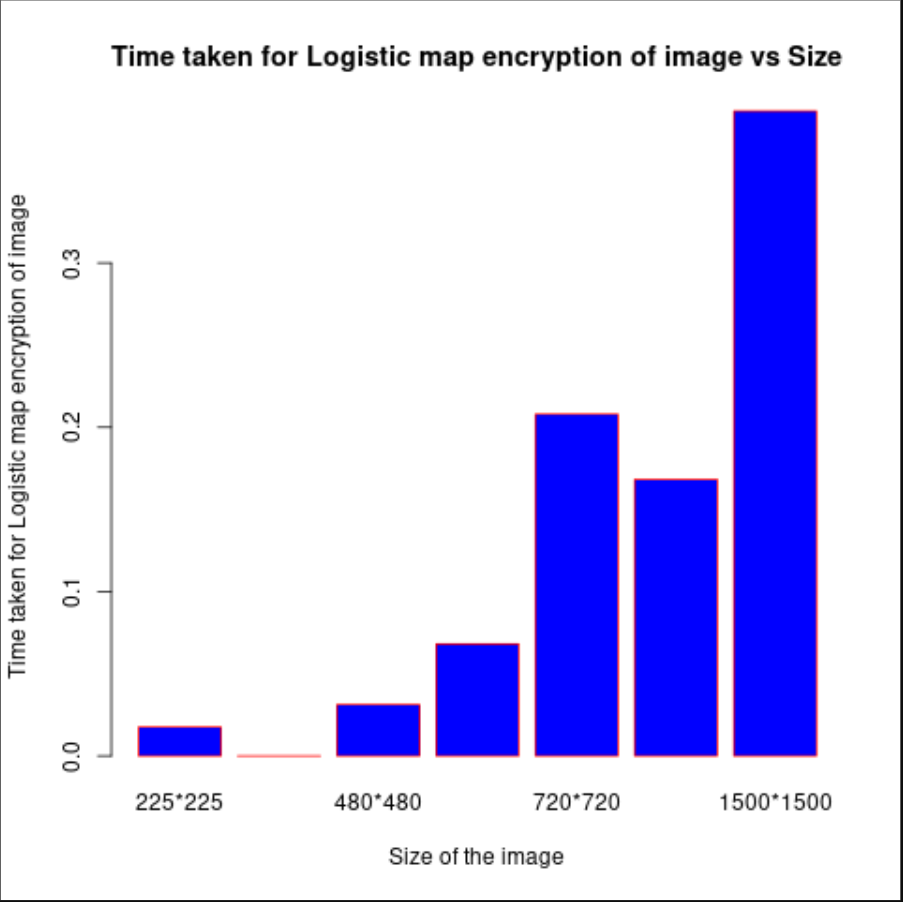
* Time taken to generate cipher image in decimal Vs Size

As the size of the image increases, the time taken to convert the cipher image in decimal also increases. The Cipher Image Matrix and the key that is received through a secure channel is then XORed to produce the Image Matrix (that is in Binary Form),in the Decryption process.



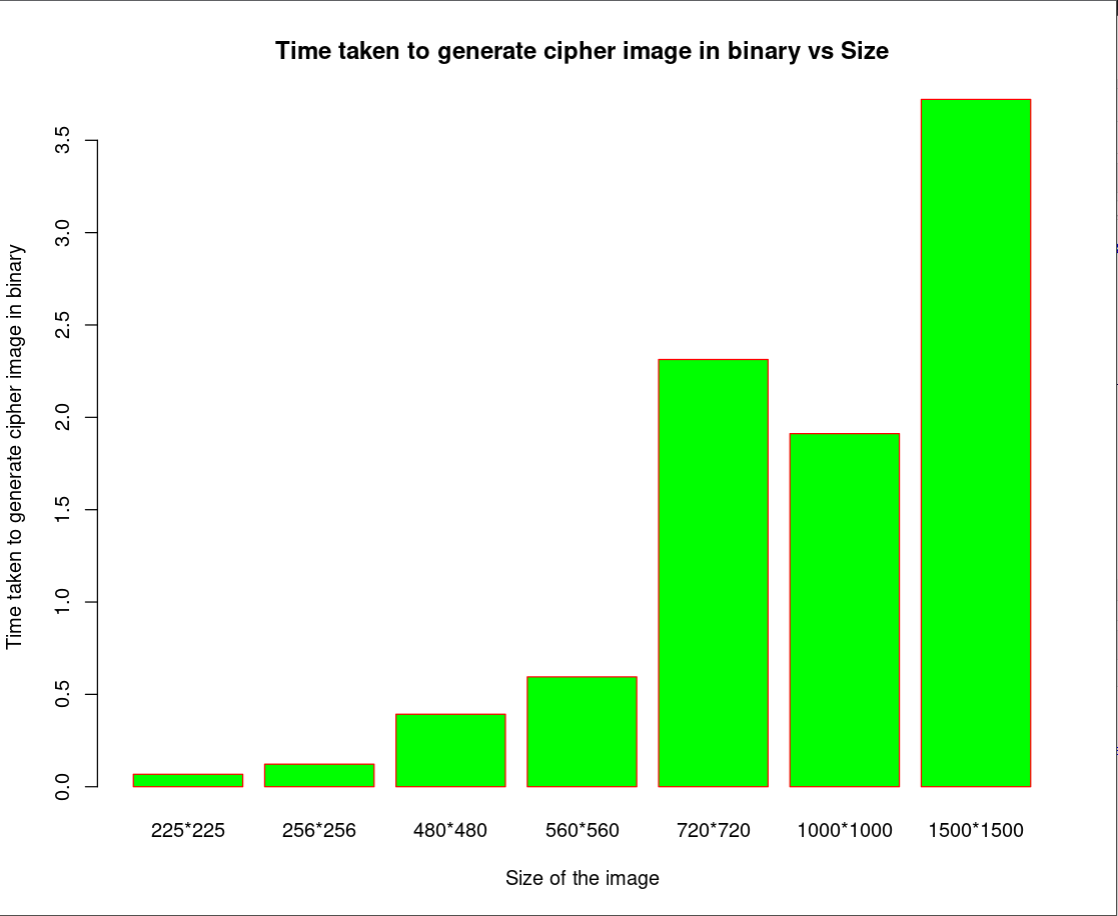
* Time taken for Logistic map encryption of image Vs Size

The following graph shows time taken to encrypt the image by using the key generated from Chaotic Logistic map and XORing it with our original image matrix, with respect to the image size. The time for logistic map encryption increases with increase of image size.



* Time taken to generate cipher image in binary Vs Size

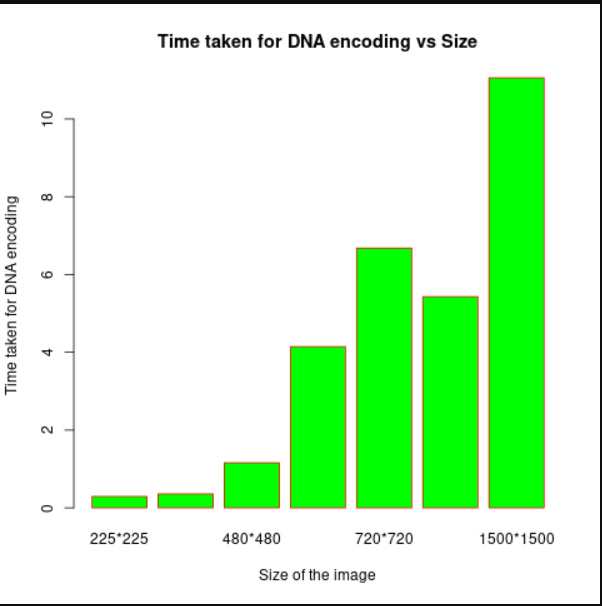
As the size of the image increases, the time taken to convert the cipher image in binary also increases. This Cipher Image Matrix and the key that is received through a secure channel is then XORed to produce the Image Matrix (that is in Binary Form),in the Decryption process.



* Time taken for DNA encoding Vs Size

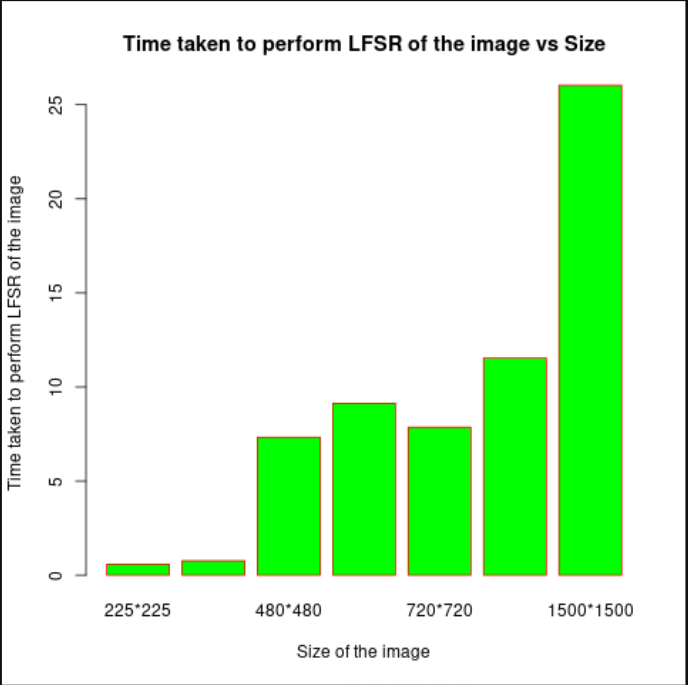
The binary sequences generated after XORing the key Ki with the image matrix Imat, are converted into a series of DNA strands by using each rule for every 2 bits in a circular fashion as discussed before. So the following graph shows that with relatively smaller image sizes value, the amount of time taken for DNA encoding is usually less

and the increases with increasing image size.



* Time taken to perform LFSR of the image Vs Size

On encrypting our given image using LFSR, the image required for very small sized images is very less whereas for images of large size (eg. 1500\*1500), the time required is relatively much more.



**References:**

[1] T.Y. Wu, X. Fan, K.H. Wang, C.F Lai, N. Xiong, and J.M.T. Wu, “A DNA Computation-Based Image Encryption Scheme for Cloud CCTV Systems”, *IEEE Access*, *7*, pp.181434-181443,October 2019.

[2] H. Dongming, Z. Ding-fu, Y. Sheng, Y. Shaoliang, Z. Luozhi and Z. Xin, “Image Encryption using Exclusive-OR with DNA Complementary Rules and Double Random Phase Encoding”, *Physics Letters A*, *383*(9), pp.915-922, February 2019.

[3] T. Wang and M.H. Wang, “Hyperchaotic Image Encryption Algorithm based on Bit-level Permutation and DNA Encoding”, *Optics & Laser Technology*, *132*, p.106355, December 2020.

[4] E. Şatir and O. Kendirli, “A Symmetric DNA Encryption Process with a Biotechnical Hardware”, *Journal of King Saud University-Science*, p.101838, January 2022.

[5] M.A. Malik, Z. Bashir, N. Iqbal, and M.A. Imtiaz, “Color Image Encryption Algorithm Based on Hyper-Chaos and DNA Computing”, *IEEE Access*, *8*, pp.88093-88107, April 2020.

[6] H. Nematzadeh, R. Enayatifar, M. Yadollahi, M. Lee, and G. Jeong, “Binary Search Tree Image encryption with DNA”, *Optik,* 163505, October 2019.

[7] X. Zhu, H. Liu, Y. Liang, and J. Wu, ”Image encryption based on Kronecker product over finite fields and DNA operation”, *Optik*, 164721, April 2020.

[8] Q. Cun, X. Tong, Z. Wang, and M. Zhang, “Selective image encryption method based on dynamic DNA coding and new chaotic map”, *Optik*, 167286, May 2021.

[9] X. Wanga, Y. Wang, X. Zhub , and C. Luoc, “A novel chaotic algorithm for image encryption utilizing one-time pad based on pixel level and DNA level”, *Optics and Lasers in Engineering*, 105851, 2020.

[10] Y. Zhanga , L. Zhanga , Z. Zhonga,b , L. Yua, M. Shana, and Y. Zhaoc, “Hyperchaotic image encryption using phase-truncated fractional Fourier transform and DNA-level operation”, *Optics and Lasers in Engineering,* 106626, March 2021.