A NEW PROCESSING OF CHAOS-BASED FAST IMAGE ENCRYPTION ALGORITHMS

[1]WALEED KHALID ABDULJABBAR, [2]SYARIZA ABDUL-RAHMAN, [3]RAZAMIN RAMLI

[1, 2]Senior Lecturer, University Utara Malaysia, School of Quantitative Science, Malaysia

[3]Associate Professor, University Utara Malaysia, School of Quantitative Science, Malaysia

E-mail: [1]waleed@uum.edu.my , [2]syariza@uum.edy.my,[3]razamin@uum.edu.my

ABSTRACT

Recently, many studies has been shown that the image encryption could be done using some techniques like RES, DES, IDEA but the new and effective technique for speedy and secure encryption using chaos-based cryptography is the most preferred encryption technique. Chaos-based encryption algorithms are a hybrid technique of multiple chaotic maps and can be repeated the same process for multiple cycles to increase the security. But, the increasing the number of steps to process an image will increase the processing time too. Another reason, if the number of pixels being encrypted increased will increase the processing time. Since, chaos-based algorithms are selected as a good choice for encrypting images during the real time applications, lesser response time and even the higher security are important. This paper will focus on evaluating encryption techniques by using two dimensional chaotic maps and comparing the strength of the encryption algorithms security and the time responses for many images have different sizes. Finally, quantitative results are compared evaluated and implemented of the serial implementation, parts of encryption and decryption process that can be parallelized (using GPU and CUDA programming) with the serial implementation.

Keywords: *Image Encryption; Cryptography; Information Security; Chaos; Network*

1. INTRODUCTION

Nowadays, security has become an essential part of digital media. Images, videos and speech are being shared and distributed in various fields like public use it for bank transactions or business communications, government use it to share secret confidential data, and in the medical field, it is used to account patients reports. All these require user authentication, reliability and accuracy of data and encryption techniques are useful tools to provide that required security. Consumer electronics like mobile phones use wireless network to share and receive images and videos which have limited bandwidth that definitely needs multimedia security.

In real time applications, time to compress or decompress and encrypt or decrypt the images is major impediments and hence it becomes difficult to handle a large amount of data. Hence, it is important to decide the right encryption algorithm depending upon the requirements and resources. It is important to understand that traditional text encryption algorithms cannot always be used for multimedia encryption, because images and videos have larger, redundant data and pixel values are highly correlated with each other, hence using text encryption algorithms like AES, IDEA, RES will take large computational time, power and will require more space to process. Unlike, text encryption decrypted images are acceptable even if we have minor inconsistencies as compared to the input image until those differences are minimal and not noticeable. In the past two decades, there have been several image encryption algorithms proposed, which can be broadly classified into three major group's position permutation [1], value transformation [2] and visual transformation based algorithms.

This paper will discuss chaotic maps and their key properties. Also, the close relationship between chaos theory and cryptography which will explain why chaos-based image encryption are often preferred over traditional encryption techniques.

Also, evaluates the security properties of the different arrangements using the chaotic maps. The security analysis is done using commonly used quantities like the number of changing pixel rate (NPCR) and the unified averaged changed intensity (UACI). Then the response time for these arrangements is calculated for both encryption and decryption process in a serial implementation.

2. CHAOS BASED IMAGE ENCRYPTION

All systems can be broadly classified as deterministic, stochastic (probabilistic) or chaotic systems of which chaotic systems are most unpredictable. Chaotic maps are often used in the study of dynamical systems which exhibit behaviour that is highly sensitive to initial conditions and even small perturbations can yield widely diverging outcomes.

There is a close relationship between chaotic systems and cryptography which makes chaos based algorithms a natural candidate for image encryption. The two basic properties of a good cipher are confusion and diffusion and both these are important features of chaotic systems too. For real-time applications encryption schemes which take lesser computational time but wouldn't compromise with the desired security are suitable. And chaos-based encryption technique is a good amalgamation of high speed, security, complexity and less power consumption. [3] provides further details about the relation between chaotic systems and cryptographic algorithms.

For given parameters [4] two initial conditions can deviate exponentially into two different ajectories. These parameters can be used for encryption and decryption and keys can be chosen from these conditions. Due, to these chaotic parameters and initial condition we could generate a large key space which further enhances the security. Because of the random behaviour, the output seems random to the attacker whereas only the sender and receiver know that the system is well defined.

2.1. Chaotic Maps

Chaotic maps can be represented using continuous and discrete time parameters. The maps are usually iterative functions with the general representation of $f: X \rightarrow X$. The process of recurrently calling the same function, where the result generated from the initial condition is fed to the same function again and the process is continued. The output from these chaotic maps exhibits

fractal-like property. Fractals are expanding symmetry which is repeating patterns; hence, it won't me incorrect to say that chaotic maps have periodicity.

The number of bits used to represent pixel tells the range of colors an image can represent. The color scale can be divided into two parts 8 bits per pixel and 24 bits per pixel. An 8 bits pixel is able to represent 2^8 or 256 different colors of grays, rather the pixel values represent the intensity of black and white colors.

2.2. Arnold Cat Map

Arnold Cat maps which was named after Vladimir Arnold. He used an image of a cat to display the effect of this chaotic map. In this mapping technique, images go through a transformation that randomizes the original image pixels. Equations 1 and 2 shows the matrix notation of the mapping transformation [5].

$$\Gamma\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \; mod \; n \qquad (1)$$

$$\Gamma \begin{bmatrix} \mathrm{x} \\ \mathrm{y} \end{bmatrix} \rightarrow \begin{bmatrix} x + y \\ x + 2y \end{bmatrix} mod \; n \qquad (2)$$

Some key features of this mapping technique are it is area preserving that is the transformed image requires the same area as the actual image, it can be even deduced as the determinant of the matrix is 1. [3] gives an overview of the properties of Cat map and basic principles of chaos-based systems. figure 1 shows how the linear map changes the unit square and how the pieces are rearranged after modulo operation.

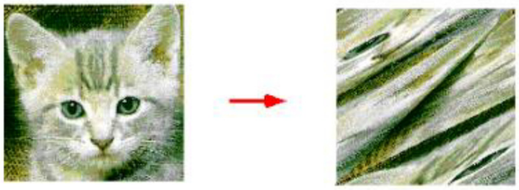


Figure.1: Arnold Cat Map transformation

2.3. Henon Map

Henon Map is a discrete dynamic system, which was developed by Michel Henon as a simplified version of Lorenz model. In 1963, Edward Lorenz examined three first-order differential equation which was attracted to a strange attractor. The equations 3, 4, and 5 are non-linear, deterministic and three dimensional. Because, of its simplicity they were widely used in areas like electric circuits, motors, chemical reactions [6].

$$\frac{dx}{dt} = \sigma(y - x) \quad (3)$$

$$\frac{dy}{dt} = x(\rho - z) - y \quad (4)$$

$$\frac{dz}{dt} = xy - \beta z \quad (5)$$

Henon carried out the experimentation using the formula defined in equation 6 and 7 for initial conditions a=1.4 and b=0.3 and based on an initial (X_0, Y_0) condition, the sequence generated a Henon attractor, which was diverging to infinity or was converging to a strange attractor. Figure 2 shows the Henon attractor after several successive iterations starting from (X_0, Y_0). [7] explains in detail the properties of Henon map and how it is derived using Lorenz systems.

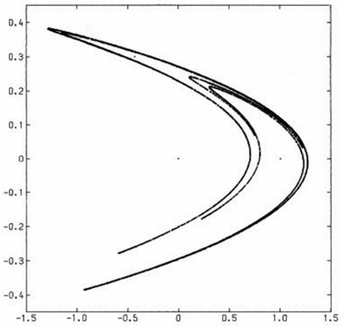


Figure 2: Henon Map after several iterations

For a given 'a' and 'b' values two unstable initial points are deduced to be x_0 =0.631354477 and

y_0=0.189406343 which is derived using the calculations shown in equations 6 and 7. Points close to this point either converge or diverge towards a fixed point or strong attractor.

$$x = \frac{\sqrt{609}-7}{28} \approx 0.631354477 \quad (6)$$

$$y = \frac{3\sqrt{609}-7}{280} \approx 0.189406343 \quad (7)$$

Duffing Map

Another famous discrete-time dynamical system which exhibits chaotic behaviour is Duffing map, equation 8 and 9 shows the x and y mapping equation. Similar, to Henon map, it takes input (X_n, Y_n) to generate (X_{n+1}, Y_{n+1}) and hence it is critical to decide the right value of 'a' and 'b' so that the behavior is chaotic. For a= 2.75 and b=0.2 [8], Duffing map produces the plot shown in figure 3.

$$x_{n+1} = y_n \quad (8)$$

$$y_{n+1} = -bx_n + ay_n - y_n^3 \quad (9)$$

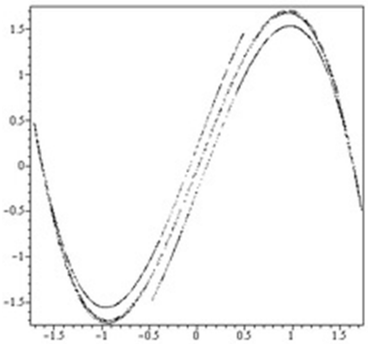


Figure 3: Duffing Map after several iterations

2.4. Cross Chaotic Map

Cross chaotic map is an amalgamation of two chaotic maps, Logistic, and Chebyshev, the equation is being referenced from [9] Both, these algorithms are one dimensional and non-linear dynamic systems and in order to reduce doing the multipart calculations, it is more efficient to combine these chaotic maps as shown in equations 10 and 11 and achieve better security level by using the resultant map in two dimensions. As per the evaluation in [10], for values, of µ=2 and k=6 the system produces great dynamic behaviour. Like Duffing map the points x_0 =0.1933 and y_0=0.8087 generated using a random generator is used as initial points for the cross chaotic map.

$$x_{i+1} = 1 - \mu . y_i . y_i \quad (10)$$

$$y_{i+1} = \cos(k . cos^{-1} x_i) \quad (11)$$

3. GRAPHICS PROCESSING UNITS

Graphics Processing Units (GPU) has become an important part of current computational systems. This progress has been possible because of the stagnation in traditional CPU clock speed and more people have started shifting focus on using GPUs for general purpose computing. [11] has information about how GPU has evolved over the years and about the different fields it is being used. GPU-accelerated computing uses GPUs along with Central Processing Units (CPU) to accelerate applications that have ample parallelism. These performance gains are best realized when the computation intensity is high and elements are largely independent. GPU can provide the speed-up by offloading the compute-intensive portions of application from CPUs as shown in figure 4.

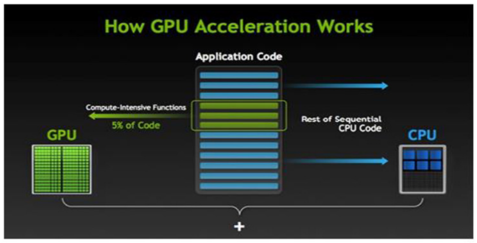

How GPU Acceleration Works
Application Code
Compute-Intensive Functions
5% of Code
GPU
Rest of Sequential CPU Code
CPU
+

Figure 4: GPU acceleration framework

4. CUDA – PROGRAMMING MODEL

CUDA is the most commonly used parallel programming model and computing platform invented by NVIDIA. CUDA makes it easy to use high-level languages like C, C++ and Fortran on GPU, hence a single code can have some part that runs sequentially on CPU and some part that runs on GPU. Earlier Graphic Processing Units were exclusively used to render graphics, but over the course of time, GPU programming has improved by introducing several new extensions and functions.

All GPU processes are termed as Kernel functions, which are executed by an array of threads. To manage thousands of threads run easily, they can be grouped into blocks which are further grouped into grids. Hence, kernel functions are executed as a grid of blocks of threads as shown in figure 5.

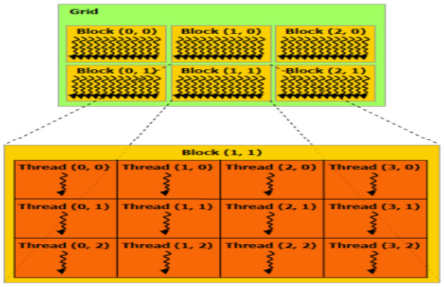


Figure 5 CUDA Thread Organizations

All threads are organized in a block uses threadIdx indexes and blocks use blockIdx to be organized inside a grid, which is predefined variables in CUDA.

5. PARALLEL PROCESSING OF PROPOSED ALGORITHM

The serial implementation of two-dimensional chaotic maps is using a C program. For Arnold Cat map, it takes the pixel values and shuffles them using simple matrix multiplication as shown in figure 6 where the red, green and blue pixel values are transformed individually and to maintain the newly mapped index within the image size it is being modulated with the image width.

The limitation of using Arnold Cat map is that image width and height must be same for the transformation to work and decrypted and the original image to be same.

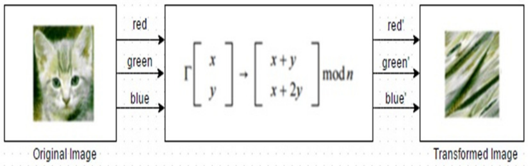

red
green
blue
$\Gamma\begin{bmatrix} x \\ y \end{bmatrix} \to \begin{bmatrix} x+y \\ x+2y \end{bmatrix} \bmod n$
red'
green'
blue'
Original Image
Transformed Image

Figure 6: Arnold Cat Map Serial implementation

For the other three 2d chaotic maps the steps for the algorithm are different and the chaotic model formula changes, figure 6 shows the two step process of the algorithm's implementation.

The first step is to use the chaotic maps to run a recursive process where X_{n+1} and Y_{n+1} pseudorandom value is calculated using X_n and Y_n as shown in figure 7. This process can run for any value of n, but these values are going to be used as a key for encrypting the pixel values, hence they need to be restricted by some constraint. The size of the key stream is a very important parameter when choosing an image encryption algorithm for real-time application. In this approach, the above process is repeated for N*N times, where N is the width and height of the image.

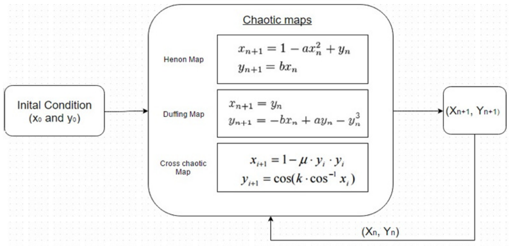


Figure 7: Key generation algorithm

This key generation logic is based on the similar approach used in [12]. Next, the N*N image is broken into individual red, green and blue pixel values into a 2-d matrix. Every individual pixel value in then XORed using the key generated in the previous step as shown in figure 8.

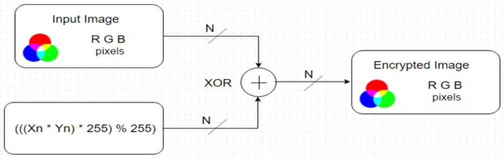


Figure 8: Encryption process

The decryption process is fairly simple because it is just repeating the same steps with the encrypted image to get the original image. Arnold cat map can be decrypted by reversing the assignment. This is the reason to use an equal width and height image so that the area is preserved and no pixel value is lost in the decryption process. In the case of Henon, Duffing and Cross chaotic maps the key generation algorithm will remain similar, but instead of XORing, the input pixels and key, the encrypted image pixels and key will be XORed to get the decrypted image pixels as shown in figure 9. Because, XOR cipher is like an additive cipher, which implies an input can be encrypted by applying the bitwise XOR function with key and when reapplying the same XOR operation will even remove the cipher key.

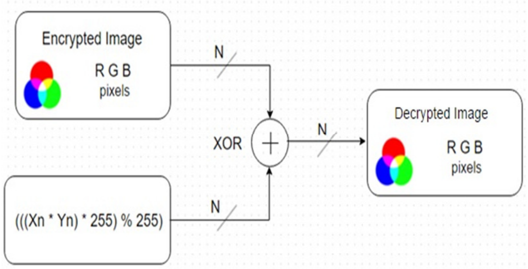


Figure 9: Decryption process

6. SECURITY ANALYSIS TECHNIQUE: RANDOMNESS TEST – NPCR AND UACI

Cryptanalysis became important after Eli Biham and Adi Shamir published a paper [13] which did a security analysis of Data Encryption Standard (DES) and various other ciphering techniques. Since then it is an important evaluation criterion to test a new algorithm.

In image encryption, algorithms strength to resist different attacks is usually evaluated using two most common quantities number of changing pixel rate (NPCR) and the unified averaged changed intensity (UACI). It is often regarded that a high NPCR and UACI value means a higher security level, but it is still not clear if that is always applicable, because there are some odd use cases where even with higher NPCR and UACI values the security level is not high.

As shown in equations 12 and 13 NPCR value focuses on the entire number of pixels where C^1 and C^2 are images before and after encryption respectively and C^1 (i, j) and C^2 (i, j) are one pixel value at i and j coordinate. T denotes the total number of pixels. If the pixel values of input and output images are same then the bipolar array D has a value 0 and if the values are different then the value is 1. The overall NPCR value ranges between [0, 1].

$$D(i,j) = \begin{cases} 0, if\, C^1(i,j) = C^2(i,j) \\ 1, if\, C^1(i,j) \neq C^2(i,j) \end{cases} \qquad (12)$$

$$NPCR: N(C^1, C^2) = \sum_{i,j} \frac{D(i,j)}{T} \times 100\% \qquad (13)$$

Another commonly used parameter for randomness test is UACI which is used to calculate the averaged difference between the input and encrypted image pixels, shown in equation 14. Here C^1 (i, j) and C^2 (i, j) denote the same as NPCR pixel values, here F denotes the largest supported pixel value, for this thesis it will be 255. Hence, even UACI value ranges from [0, 1].

$$UACI : u(C^1, C^2) = \sum_{i,j} \frac{|C^1(i,j) - C^2(i,j)|}{F\,.T} \times 100\%$$
(14)

Based on the experimental findings in [14] comparing two encryption outputs based on their test scores quantitatively is not accurate. It is noticeable that NPCR values are often close in the range of 99-100%. Hence, it is preferable to have a high NPCR value, but the differences are not very significant. But, UACI values calculated using numerical and experimental results it is clear that many image encryption methods fail UACI test because of either too low or too high scores.

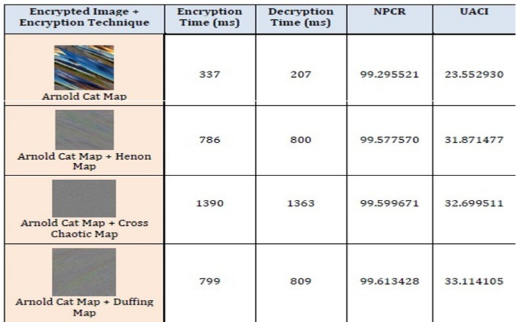
7. EXPERIMENTAL RESULTS

These experiments are performed on different image sizes using the above mentioned encryption techniques, based on which the encryption and decryption times, NPCR and UACI values are calculated. These experimental results show that to achieve higher security processing time increases substantially too.

If we combine the different chaotic maps as a combination of encryption steps the processing time will be almost equivalent to the summation of the individual encryption process. Hence, it is critical to maintaining a balance between achieving better securities and running time of the algorithm. Since chaotic maps are used for image encryption in real-time applications it is important to improve the processing time.

The following experiments (see figures 10 and 11) are performed using one chaotic map or by combining Henon, cross chaotic and duffing maps with Arnold Cat Map. Based, on the results, it is evident that as the image size increases the encryption and decryption time increases much faster, which a bottleneck is when faster response time is expected.

Figure 10: 2048*2048 input image



Encrypted Image + Encryption Technique	Encryption Time (ms)	Decryption Time (ms)	NPCR	UACI
Arnold Cat Map	337	207	99.295521	23.552930
Arnold Cat Map + Henon Map	786	800	99.577570	31.871477
Arnold Cat Map + Cross Chaotic Map	1390	1363	99.599671	32.699511
Arnold Cat Map + Duffing Map	799	809	99.613428	33.114105

Figure 11: Encryption results for 2048*2048 image size - multiple encryption steps

On comparing the NPCR and UACI values cross chaotic and duffing maps are more efficient than the others also when the encryption process comprises of multiple encryption steps using a combination of Arnold Cat map with Henon, Duffing and cross chaotic maps the results are further improved, but this increases the total encryption time even more. There is a direct co-relation between the image size, level of security and encryption time.

Using this property of key generation the overall time for encrypting multiple images can be improved. Figures 12 and 13 shows four 4096*4096 size images using one key generation step and that encrypts all the four images.

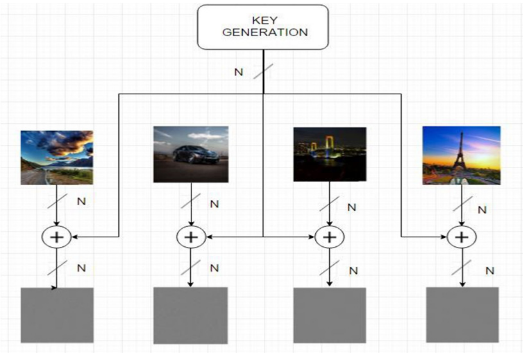
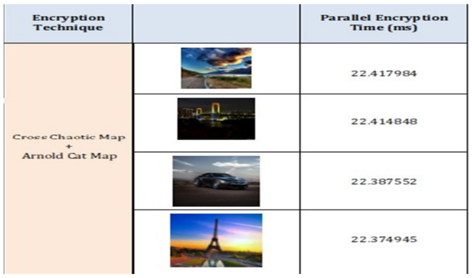


Figure 12: Parallel encrypt.ion of some images using Cross chaotic and Arnold cat map



Encryption Technique		Parallel Encryption Time (ms)
Cross Chaotic Map + Arnold Cat Map		22.417984
		22.414848
		22.387552
		22.374945

Figure 13: Encryption time for 4096*4096 image pixels without key generation time

8. CONCLUSION

From the above-discussed encryption schemes, some principles could be used for more robust and efficient image encryption algorithms. The primary focus of this paper is to understand chaos-based image encryption technique. In the past decade, there has been several a new image encryption technique developed or enhancements for the existing techniques being done. Similarly, this paper work does an analysis of the two-dimensional chaotic maps and how they are implemented as encryption algorithms.

Because of the property of image pixels being separate entities they can be encrypted independently and hence this research uses Graphic processing units (GPU) to parallel process these image pixels and then profile their behaviour based on different parameters. The results were improved substantially, because the time complexity is reduced from quadratic O (n^2) to constant O (1) time, Even though using GPU, we need to transfer the data from host system to the GPU device which is an overhead, but this varies from machine to machine, there are also efficient ways of making it less time consuming and above all when compared with the overall time improvement this overhead time is minuscule.

But, the main bottleneck is the encryption key generation algorithm, as chaos-based encryption algorithms are based on the principle of feedback system where current values are dependent upon the previous values. Hence, they can't be parallelized, to overcome this issue, multiple images are being encrypted using single key generation step. When implemented using four images of 4096 by 4096 pixels and a single key generation step speedup of ~10 times was achieved.

REFERENCES

[1] A. Sinha and K. Singh, "A technique for image encryption using digital signature," *Opt. Commun.*, 2003.

[2] D. Panchal, J. Chaita, and P. Hemin,

"IJEDR(ISSN 2321-9939)," *Int. J. Eng. Dev. Res.*, vol. 4, no. 3, pp. 561–565, 2015.

[3] Y. Mao and G. Chen, "Chaos-based image encryption," in *Handbook of Geometric Computing*, Springer, 2005, pp. 231–265.

[4] Q. Lawande, B. Ivan, and S. Dhodapkar, "Chaos based cryptography: a new approach to secure communications," *BARC Newsl.*, 2005.

[5] Wikipedia contributors., "Arnold's cat map," *The Free Encyclopedia, 9 Aug. 2015. Web. 2 Mar. 2016.* 2016.

[6] Wikipedia contributors, "Lorenz system," *The Free Encyclopedia, 8 Jan. 2016. Web. 2 Mar. 2016.* 2016.

[7] M. Hénon, "A Two-dimensional Mapping with a Strange Attractor," in *The Theory of Chaotic Attractors*, New York, NY: Springer New York, 1976, pp. 94–102.

[8] Wikipedia contributors, "Duffing map," *Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 10 Oct. 2013. Web. 2 Mar*. 2016.

[9] Z. Maotai and J. Sha, "Simulation Results," in *Communications and Information Processing International Conference*, 2012, pp. 139–141.

[10] L. Wang, Q. Ye, Y. Xiao, and Y. Zou, "An image encryption scheme based on cross chaotic map," *Image and Signal*, 2008.

[11] J. D. Owens, M. Houston, D. Luebke, S. Green, J. E. Stone, and J. C. Phillips, "GPU computing," *Proc.*, vol. 96, no. 5, pp. 879–899, 2008.

[12] S. Kumar, B. Sinha, and C. Pradhan, "Comparative Analysis of Color Image Encryption Using 2D Chaotic Maps," 2015, pp. 379–387.

[13] E. Biham and A. Shamir, "Differential cryptanalysis of DES-like cryptosystems," *J. Cryptol.*, vol. 4, no. 1, pp. 3–72, 1991.

[14] Y. Wu, J. Noonan, and S. Agaian, "NPCR and UACI randomness tests for image encryption," *Cyber journals Multidiscip.*, 2011.