An Accelerator-based Logistic Map Image Cryptosystems for Grayscale Images

M. Raviraja Holla[1,*], Alwyn R. Pais[2] and D. Suma[3]

[1]*Department of Information and Communication Technology, Manipal Institute of Technology, Manipal Academy of Higher Education (MAHE), Manipal, Karnataka, India*
[2]*Information Security Research Lab., Department of Computer Science and Engineering, National Institute of Technology Karnataka, Surathkal, India*
[3]*Department of Computer Science and Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education (MAHE), Manipal, Karnataka, India*
E-mail: raviraj.holla@manipal.edu
[*]*Corresponding Author*

Received 22 October 2020; Accepted 31 January 2021;
Publication 13 May 2021

Abstract

The logistic map is a class of chaotic maps. It is still in use in image cryptography. The logistic map cryptosystem has two stages, namely permutation, and diffusion. These two stages being computationally intensive, the permutation relocates the pixels, whereas the diffusion rescales them. The research on refining the logistic map is progressing to make the encryption more secure. Now there is a need to improve its efficiency to enable such models to fit for high-speed applications. The new invention of accelerators offers efficiency. But the inherent data dependencies hinder the use of accelerators. This paper discusses the novelty of identifying independent data-parallel tasks in a logistic map, handing them over to the accelerators, and improving their efficiency. Among the two accelerator models proposed, the first one achieves peak efficiency using coalesced memory access. The

Journal of Cyber Security and Mobility, Vol. 10_3, 487–510.
doi: 10.13052/jcsm2245-1439.1031

other cryptosystem further improves performance at the cost of more execution resources. In this investigation, it is noteworthy that the parallelly accelerated logistic map achieved a significant speedup to the larger grayscale image used. The objective security estimates proved that the two stages of the proposed systems progressively ensure security.

Keywords: Accelerator, logistic map, encryption, cryptography.

1 Introduction

A successful attack on an Information Technology (IT) system is called a security incident. Security incidents have drastically increased over the years in the world of the Internet. Due to the open unsafe communication channel and the broad clients of the Internet, it is easy for an adversary to exploit vulnerabilities and commit security incidents. Dynamically preventive security is only a viable solution than any detective or recovery security possibilities. The emerging image cryptography models provide such security solutions.

The invention of image encryption by scrambling the image dates back to 1975 [1]. The original image is converted to an unrecognizable image using a photographic approach. This reversible and reproducible transformation used a code plate comprising of unique random binary codes. A reversible logical operation decides any bit of the scrambled image on the corresponding pair of bits in the secret image and the code plate. The users have the code plates authorized to decipher the plain-image. The original image, when closely contacted with the code plate revealed the secrecy. The researchers in [2] used a chaotic map to encrypt the image. The pixel correlation in the two-dimensional representation of an image is high. So, authors in [2] used a high-dimensional map to de-correlate this correspondence by transposition of the pixels. This operation is called a permutation. Another procedure called diffusion makes the encrypted image distinctly different from the plain-image. This logistic map is a chaotic map of polynomial recurrence relation of degree 2. This function is a low-dimensional relation that can be easily implemented for an encryption [3]. In [4], the researchers analyzed the two primary operations based on a logistic map for evaluation. These operations are permutation and diffusion. The investigators interchanged the sequencing of these two operations and evaluated the effectiveness. This evaluation can be considered a reference model for selecting the effective and efficient process and the ordering of activities for chaotic map encryption.

Other than the pixel correlation in two-dimensional space, images have characteristics of high-capacity and redundancy of computations. Due to these two features, the logistic map based encryption is not affordable for on-line communications and real-time applications [5]. The last decade has witnessed an exponential growth in parallel processing power and paradigms [6]. An efficient logistic map permutation-diffusion-based image cryptosystems are proposed in this paper. These systems exploit accelerators to improve the elapsed time spent on processing. The proposed cryptosystems are objectively analyzed using various metrics to prove that the system is robustly secured.

2 Background and Motivation

The use of chaotic maps in the image cryptography is growing. Recently, the application of the Chaotic Map for the encryption is exemplified in [7] using the DNA encoding technique. The chaos is the base for the key-image. The key and secret images are encoded row by row with DNA rules. The logistic map ensures that different rows are encoded – the final cipher image obtained by repeating the operations column by column. Table 1 consolidates and compares the literature discussed so far.

The authors in [8] identified a plain-text attack vulnerability in a specific chaotic mapping cryptography algorithm. The researchers then introduced a chaos-based image replacement to withstand plain-text attacks. This novel method retained other qualities of the technique they analyzed. This article lists features that may be required by any chaotic cryptosystem. The most important of them is the flexibility of selecting any chaotic mapping, vast chaotic space, reduced redundancies or repetitions, secret key sensitivity, good security, and proper pixel distribution after encryption.

Recently, concerns about chaotic cryptosystem efficiency have begun to emerge. Here are some of the findings that inspire adding efficiency to the cryptosystem list of desired features. These findings also provide both direct and indirect reasons for that proposition. If the cryptosystem's emerging applications are the immediate causes, its features are indirect causes. Real-time [5, 9], high-protective [5, 10, 11], remote sensing [12], computer vision [13], deep learning [14], super-resolution [15, 16], and multimedia [17] applications are important indirect causes. Transmission cost [18], hardware cost, and the power consumption [19, 20], is an example of an indirect cause. The use of multi-type maps [21], the size of the images, and some redundancy [5] can also help with efficiency through parallelism.

Table 1 Comparison analysis of existing literature

Author	Features	Pros and Cons
Hines [1]	Original image transformed to a scrambled image and later restored using same code plates.	Need reduction in computational cost.
Chen et al. [2]	Used three-dimensional cat chaotic map for transposition of pixels of the original image and another map to de-correlate pixels of the original and cipher-images.	Improved security. Speedy encryption.
Liu et al. [3]	Variable parameter one-dimensional map used for pixel shuffling and dynamical reversible algorithm used for the encryption.	High security. Reduced execution time for encryption. But do not fit for real-time applications.
Wang et al. [4]	Analysed effect of the permutation and diffusion operations based on the chosen chaotic maps on security.	Concluded that effect of the order of permutation and diffusion on security is specific to the chosen.
He et al. [5]	Proposed parallel two-dimensional Arnold chaotic map for pixel shuffling and three-dimensional Liu map system for the encryption.	An efficient model that can be applied to real-time applications.
Wang and Liu [7]	The parameters needed for the encryption algorithm are derived from Piecewise Linear Map and Logistic Map. DNA rule is used for encoding.	It is a serial model. But the model is computationally intensive. So, the researchers reserved their future work making this model efficient using an accelerator.

Nonetheless, all these inventions used a modern parallel paradigm. Parallel computing has revolutionized the field of computing [22, 23]. The field cryptography has to gain a lot from concurrency. The article [22] overviews the parallel-paradigm development overt the time-line. Notably, there is a growing demand for parallel thinking for better efficiency. The main part of any structured cryptography included chaos-based cryptography [17].

However, image cryptography algorithms based on complex, chaotic maps or other designs cannot be easily parallelized [19]. These opportunities and the challenges have driven the efficiency of the logistic-map cryptography with state of the art parallel technology. This paper describes the investigation of such a unique and novel accelerated logistic map cryptosystems.

3 Method

3.1 Accelerated Logistic Map Image Cryptosystems

The architecture in the proposed systems use two steps proposed in [8] during encryption. They are the permutation and diffusion. Decryption reversely reuses these two steps. The permutation – diffusion model which uses logistic map is shown in Figure 2. Equation (1) denotes the logistic map equation used in this model.

$$x_n = rx_n(1 - x_n) \tag{1}$$

The r in Equation (1) is a positive constant. This quadratic map has a very complex behavior in various iterations, as shown in Figure 1 for the value of $r = 4$.

In the proposed systems, the initial values set to $r = 3.94$ and $x_0 = 0.4$. Equation (1) generates the logistic sequence of double-precision numbers. This sequence is placed in sequential order. The pixel position in each row is changed based on this sequence. The diffusion generates the encrypted image by performing a bitwise XOR operation on every pixel with a variable say l, where l is obtained by multiplying each element of the generated sequence with a key 987654321012345 and performing mod 255. The decryption reverses these two steps to produce the original image. Initially, we develop the CPU-based logistic map architecture as a basis to

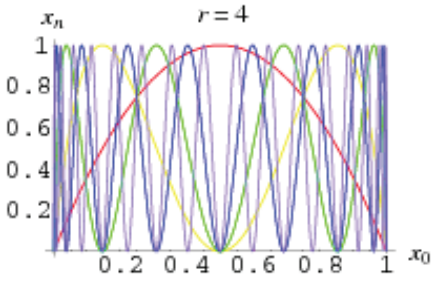

Figure 1 Logistic function (red) and its behaviour in various iterations [24].

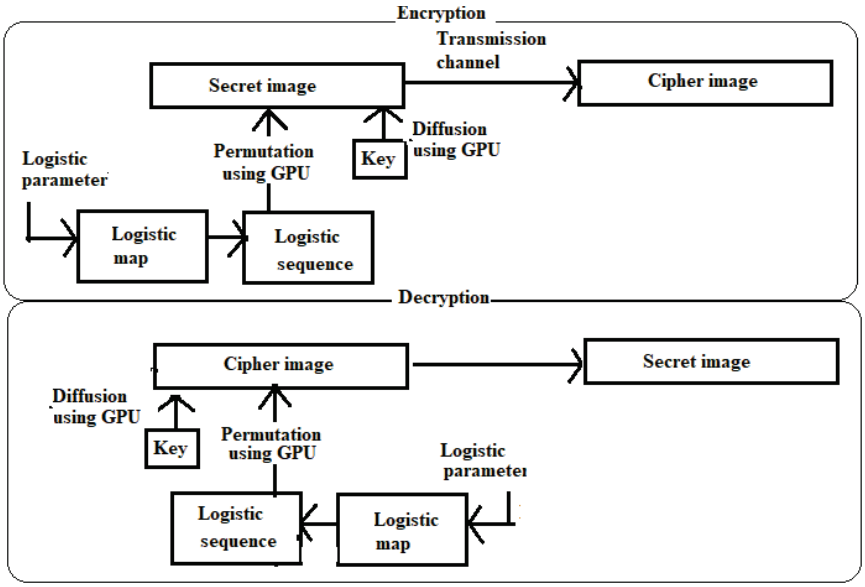

Figure 2 Permutation – diffusion logistic map architecture.

propose two efficient accelerator-based cryptosystems. In this accelerator-based category, Figure 5 shows the first version, which is a refined model shown in Figure 3. The similarity among these two models is that both these models generate threads equal to the total image height. However, in the naive model (Figure 3), neighbouring threads store the interleaved memory addresses in each iteration.

The consequence is two-fold. The interleaved references for single access lead to memory underutilization because the data available between these references remain unused for that access. This misaligned and non-coalesced access often leads to multiple memory accesses. Instead of directly processing, processing the transpose of the image guarantees coalesced memory access. By transposing the image in Figure 5, adjacent threads generate adjacent addresses. As there are now regular access patterns unlike interleaved, many accesses otherwise are coalesced to one, improving memory bandwidth utilization to achieve peak performance [25]. Figure 4 shows the coalesced and non-coalesced accesses.

In the second accelerator version, instead of image height, threads equal to the image size are created as in Figure 6. This model improves speedup further by eliminating the thread traversals.

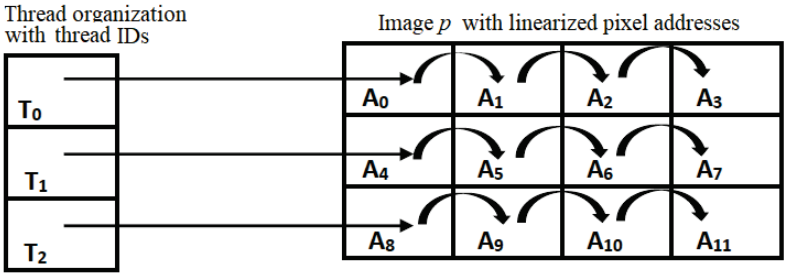

Figure 3 Thread organization and pixel mapping in general accelerator model.

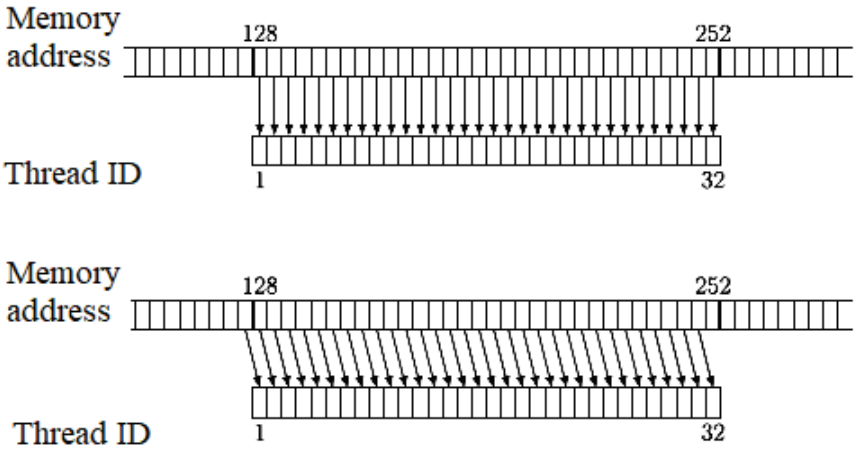

Figure 4 Global memory access [26] (a) Aligned and successive access (b) Mis-aligned access.

3.2 Accelerator Programming Using PyCUDA

Using PyCUDA in a parallel program has three benefits. They are productivity, power efficiency, and performance [27]. PyCUDA provides functions for accessing CUDA in a Python program. These include allocating accelerator memory, exchanging data with accelerator memory, and assigning workloads to the accelerator with light-weight threads to process data. In the created threads, if each thread is assigned a task to handle separate data, the job's redundancy can be avoided entirely. This approach makes complete efficiency possible. But this is not possible when there is a relationship between the data. Instead, each thread can handle a separate data block instead of just one data. Redundancy can thus significantly reduce. Then the efficiency deficit is shallow. Creating a single thread, and assigning all the data to it will keep the

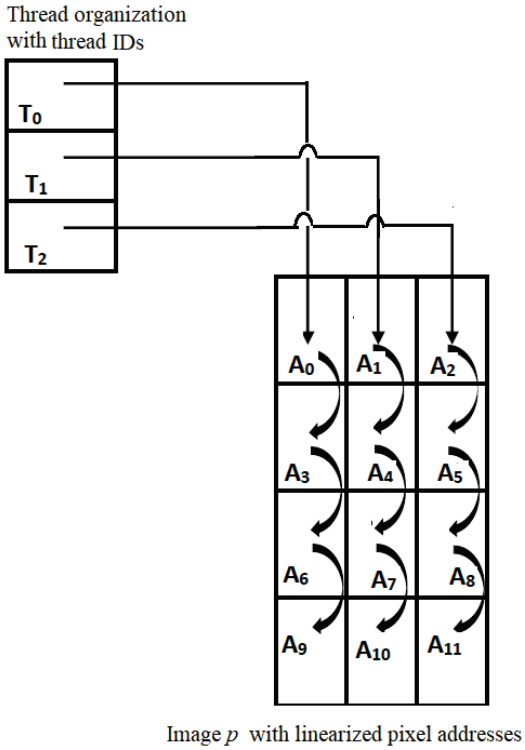

Figure 5 Thread organization and pixel mapping in accelerator based version 1.

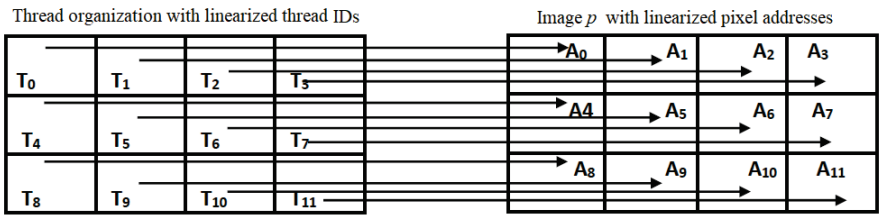

Figure 6 Thread organization and pixel mapping in accelerator based version 2.

full redundancy, results in inefficiency. Overall, efficiency rests on the amount of data and the number of threads created based on the dependency between data. Also, there are recent developments to optimize acceleration in CUDA. Coherent instructions in a function direct the work that the accelerator needs to do in parallel. That function is called the kernel. The accelerator cores by multitasking execute the threads containing these logical kernels.

Algorithm 1: Manages both CPU and GPU resources. Also, launches the kernels $encryption\_coalesced()$ and $encryption\_thread\_per\_pixel()$.

`// Host function.`
`// Assume` h `= height and` w `= width.`
Input: Grayscale secret image p (size $h \times w$), chaotic array x (scalar size w), index array z (scalar size w).
Output: Encrypted grayscale image g' (size $h \times w$).
1 Initialize x with chaotic series using the logistic map using Eq.1 and sort x, while maintaining their original indices in z.
2 Allocate accelerator memory for g, p, x, and z.
3 Transfer p, x, and z to the accelerator memory.
4 Set $configuration\_parameters$ to create n threads for the kernel.
`// For Accelerator based kernel 1 (namely` $encryption\_coalesced()$`,` $n = h$`. For Accelerator based kernel 2 (namely` $encryption\_thread\_per\_pixel()$`,` $n = h \times w$`.`
5 Launch only one among the two kernels below in each execution to obtain the corresponding result.
$encryption\_coalesced(g, p, x, z, h, w, configuration\_parameters)$.
$encryption\_thread\_per\_pixel(g, p, x, z, h, w, configuration\_parameters)$.
6 Transfer g from GPU memory to g' CPU memory.
7 Display g'.
8 End.

Algorithm 1, Algorithm 2, and Algorithm 3 show the host function, accelerator-based two versions of the proposed system, as explained in Section 3.1.

3.3 Result and the Speedup of the Proposed Systems

Table 2 shows the total execution times of permutation and diffusion of three versions explained in Section 3.1. The increase in image size significantly increases the execution time in the CPU based version. Both accelerator-based versions are almost neutral to image size. The run time rises slightly for the rise in the image's width in the accelerator-based first version. The speedup is the efficiency of the accelerator-based models over the CPU-based model. The speedup of the first version of the accelerator is increased by 3347 times while the latter is about 20000 times, for the image size 1024×1024. This speedup rises significantly as the image size increases. Figure 7 shows the speeds of the proposed system with its two variants. We also compared

Algorithm 2: $encryption\_coalesced(g, p, x, z, h, w)$: Generates an encrypted image after permutation and diffusion.

```
// Kernel function
// Accelerator based 1 with number of threads n = h
```
Input: g, p, x, z, h, w as defined in Algorithm 1
Output: Encrypted grayscale image g (size $h \times w$)
1 p' = Transpose(input image p).
```
// Coalesced access of p'.
```
2 Create per thread unique global index to point to the starting address of the each column of p' to achieve coalesced access.
```
// Each thread traverses columns successively in each iteration.
// Perform permutation.
```
3 **for** *i = 1:w* **do**
4 | Let h threads perform permutaion in p', using permuation address in z.
5 **end**
```
// Perform diffusion.
```
6 **for** *i = 1:w* **do**
7 | Let h threads perform diffusion in p'.
8 **end**
```
// Store p' in g.
```
9 g = Transpose(p').
10 Return.

Algorithm 3: $encryption\_thread\_per\_pixel(g, p, x, z, h, w)$: Generates an encrypted image after permutation and diffusion.

```
// Kernel function
// Accelerator based 2 with number of threads n = h × w
```
Input: g, p, x, z, h, w as defined in Algorithm 1
Output: Encrypted grayscale image g (size $h \times w$)
1 Create per thread unique global index pointing to the its corresponding pixel in p.
```
// Each thread processes its pixel.
```
2 Perform permutation concurrently in p.
3 Perform diffusion concurrently in p.
```
// Store p in g.
```
4 $g = p$
5 Return.

the execution times of the proposed models with the previous models in Table 3. The experimental values proved that our algorithms significantly outperform those models concerning the execution time enabling them eligible for real-time applications.

Table 2 The execution times (in seconds) for permutation and diffusion

Environment	Grayscale image size (Height x Width)		
	256×256	512×512	1024×1024
CPU based	2.1024	3.3021	6.0254
Accelerator based 1	0.0016	0.0017	0.0018
Accelerator based 2	0.0002	0.0002	0.0003

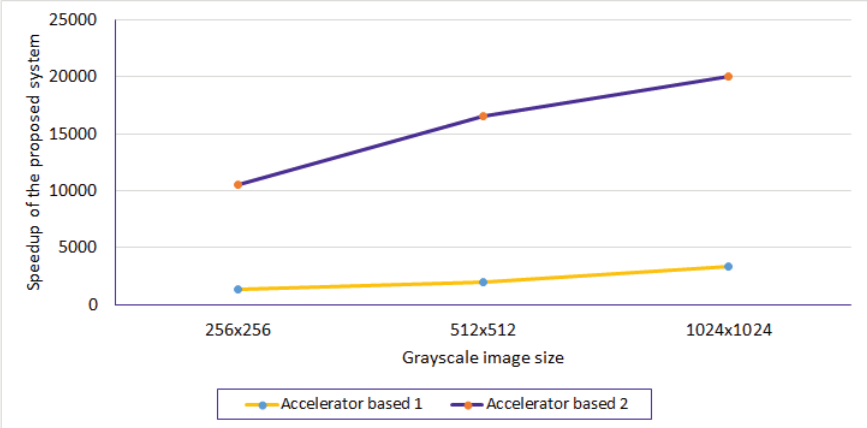

Figure 7 Speedup of the proposed systems during encryption.

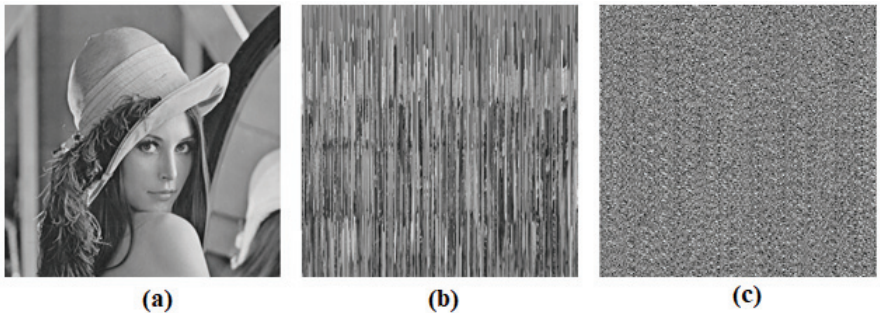

Figure 8 Image transformations during encryption (a) Secret image (b) Permuted image (c) Encrypted image.

For the grayscale image in Figure 8(a), Figure 8(b) and Figure 8(c) shows the transformation after permutation and diffusion during encryption, respectively. The decryption produces images in the reverse order.

3.4 Experimental Environment

PARAM Shavak Super Computing hardware with a CPU: Intel®Xeon ® – E5 – 2670 with two cores each containing 24 cores, 8TB RAM, 2.3 GHz. and single NVIDIA®Tesla K40 GPU: containing 2880 cores, 12 GB GDDR5, 745 MHz, PyCUDA to access NVIDIA's Compute Unified Device Architecture (CUDA) parallel API from Python and OpenCV-Python.

4 Objective Security Assessments of the Proposed Systems

Two general categories of assessing the quality of an intermediate or final image relative to a reference image are subjective and objective [28]. The humans directly perceive and evaluate the image quality in the subjective-category. The objective-category uses computation to predict image ranking. However, subjective analysis is expensive in terms of time and budget. Also, the image computing can not make use of it for optimization [28]. For cryptography, it is appropriate to consider security as an essential quality attribute. This section discusses a few objective image security assessments of the proposed systems.

4.1 Histogram Analysis

A histogram shows the number of pixels in the equal intervals between the inclusive minimum and maximum intensity values. Figure 9 shows the histograms of the secret, permuted, and encrypted images. Since the pixels in the permuted image are the relocated pixels of the secret image, its tone distribution is the same as the original image. So their histograms (Figure 9(b) and Figure 9(a)) are the same, respectively. But the histogram of the encrypted image (Figure 9(c)) is different as it changes the tone intensity of the permuted image. Also, the fact that the tone distribution is almost equal in the encrypted image authenticates how secure is the encrypted image.

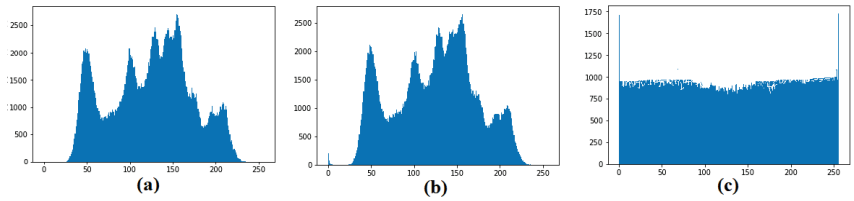

Figure 9 Histogram of (a) Secret image, (b) Permuted image, and (c) Encrypted image.

4.2 Shannon Entropy Analysis

The information entropy ($H(U)$) is a metric to quantify the randomness in an image. Equation (2) defines Shannon's entropy [33], with $p(u_j)$ representing grayscale probability distribution. For 256 grayscale values of an image, $H(U)$ lies in the range $(0 \leq H(U) \leq 8)$. $H(U) = 0$ when the image pixels have a constant value. $H(U) = 8$ for the uniform pixel distribution in the histogram. From the table, it is evident that the $H(U)$ of the encrypted image (Figure 8(c)) approches the maximum theoretic 8 [34]. The computed entropy values of the secret (Figure 8(a)) and the permuted images (Figure 8(b)) are 7.446 and 7.459, respectively. Table 4 lists the entropy values of the secret image, permuted image, and the encrypted image. Figure 10 shows the Shannon entropy, $H(U)$, analysis of the proposed system. Therefore the proposed cryptosystem is more secured.

$$H(U) = \sum_{j=0}^{255} p(u_j) \log_2 p(u_j) \tag{2}$$

Table 3 reveals that the proposed models achieved the ideal entropy estimate when compared to the existing models.

4.3 Structural Similarity Index (SSIM) Analysis

The Structural Similarity Index $(-1 \leq SSIM \leq 1)$ is a spatial domain metric that compares luminance, contrast, and structure of two images of a single capture [35, 28]. The $SSIM = 1$ for the two identical images and approaches minimum for the distinct images. The Equation (3) gives the $SSIM$ for the comparison of a distorted image, q with a reference plain image, p.

$$SSIM(p,q) = [l(p,q)]^{\alpha}.[c(p,q)]^{\beta}.[s(p,q)]^{\gamma} \tag{3}$$

Where,

$$l(p,q) = \frac{2\mu_p\mu_q + C1}{\mu_p^2 + \mu_q^2 + C2}$$
$$c(p,q) = \frac{2\sigma_p\sigma_q + C2}{\sigma_p^2 + \sigma_q^2 + C2}$$
$$s(p,q) = \frac{\sigma_{pq} + C3}{\sigma_p\sigma_q + C3}$$

Equation (4) is the simplified form of Equation (3), assuming $\alpha = \beta = \gamma = 1$ and $C3 = C2/2$.

$$SSIM(p,q) = \frac{(2\mu_p\mu_q + C1)(2\sigma_{pq} + C2)}{(\mu_p^2 + \mu_q^2 + C1)(\sigma_p^2 + \sigma_q^2 + C2)} \tag{4}$$

Table 3 Comparison of objective security assessments of the proposed systems with the previous investigations

Previous Investigations												
References (Ref.)	Execution time (seconds) for image sizes 256, 512, and 1024.			Entropy $H(U)$	$SSIM$	MSE	$PSNR$	Pearson Correlation r_{pq}			$NPCR$	$UACI$
								r_h	r_v	r_d		
Ref. [29]	0.3456	1.4016	5.3793	7.897	17.237	47.8	31.3	−0.0070	−0.0015	0.0054	98.71	31.9715
Ref. [30]	0.4139	1.6877	6.8321	7,997	16.876	50.7	31.08	0.0061	0.0067	−0.0070	99.60	32.0100
Ref. [31]	0.2854	1.1612	4.3270	7.997	19.761	40.3	32.07	−0.0042	0.0114	−0.0074	99.08	33.5612
Ref. [32]	0.0933	0.3901	1.4803	7.999	15.947	7319	9.486	−0.0133	0.0007	0.0077	100	33.5515
Ref. [8]	0.0629	0.2673	1.2157	7.993	12.116	1451	16.51	−0.0209	−0.0028	−0.0030	100	33.4183
Proposed Models												
CPU based	2.1024	3.3021	6.0254	7.998	9.018	8143	9.023	−0.0087	−0.0279	0.0246	100	33.4732
Accelerator based 1	0.0016	0.0017	0.0018	7.999	9.015	8141	9.023	0.0035	0.0037	−0.0095	100	33.4745
Accelerator based 2	0.0002	0.0002	0.0003	7.998	9.014	8143	9.023	0.0039	0.0035	−0.0098	100	33.4637

Table 4 Shannon entropy i.e., $H(U)$ analysis

Image	$H(U)$
Figure 8(a)	7.446
Figure 8(b)	7.459
Figure 8(c)	7.998

Table 5 The $SSIM$, MSE, and $PSNR$ between two images in the proposed system

Image p	Image q	$SSIM$	MSE	$PSNR$
Figure 8(a)	Figure 8(b)	0.4463	4369	11.73
Figure 8(b)	Figure 8(c)	0.4639	7783	9.219
Figure 8(a)	Figure 8(c)	0.0098	8143	9.023

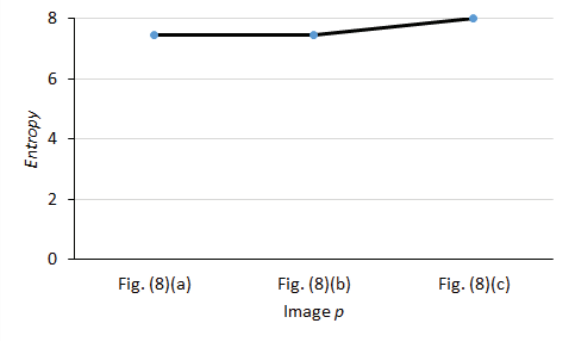

Figure 10 Shannon entropy i.e., $H(U)$ analysis.

where σ_p, σ_q are the local standard deviations of p and q, μ_p, μ_q, are the means of p and q, and σ_{pq} is the cross-covariance of p and q.

When it comes to security, the $SSIM$'s low value indicates that the image q is more different from image p and provides more protection. Table 5 lists the $SSIM$ values of the two compared images p and q. In the first row, p = Figure 8(a) (i.e., the original image) and q = Figure 8(b) (i.e., the permuted image). Similarly, it is possible to interpret the second and third rows of this table. From the table, it is evident that the $SSIM$ of the encrypted image ($q =$ Figure 8(c)) is almost zero compared to the secret image ($p =$ Figure 8(a)), indicating that the encrypted image is distinctly different from the secret image. It means that the encrypted image is least vulnerable to attacks and more secure. Figure 11 shows the $SSIM$ analysis of the proposed system. We also demonstrate in Table 3 that our algorithms outperform other existing algorithms concerning SSIM.

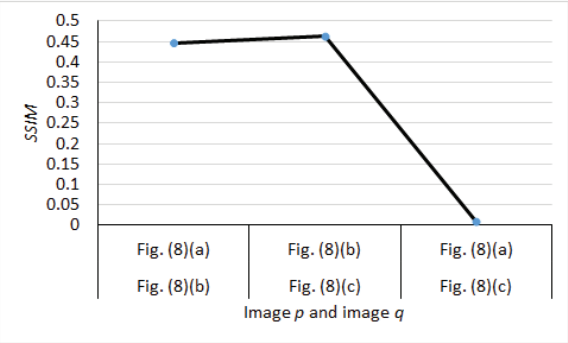

Figure 11 $SSIM$ index analysis.

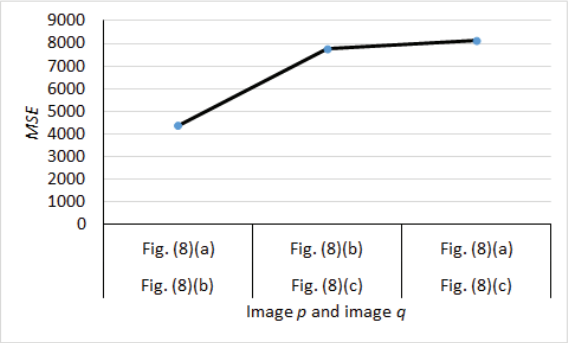

Figure 12 MSE analysis.

4.4 Mean Square Error (MSE) Analysis

There are two standard error-based related image quality measures [28]. They are the mean squared error (MSE) and the peak signal-to-noise ratio ($PSNR$). Section 4.5 covers this relationship with a discussion on $PSNR$. Let h and w represent the height and the width of the image. Let $p = \{p(i,j) \mid (0 \geq i \leq h-1), (0 \geq j \leq w-1)\}$ and $q = \{q(i,j) \mid (0 \geq i \leq h-1), (0 \geq j \leq w-1)\}$ represent the reference and the target images, respectively, where $p(i,j)$ and $q(i,j)$ represent the intensities of the pixels at (i,j) in p and q respectively. MSE is defined in Equation (5).

Higher the MSE, the better is the encryption meaning that the reference and the target images are dissimilar. Table 5 contains MSE between any two pairs of the secret image, permuted image, and the encrypted image. Figure 12 shows the plot of these values. The increasing MSE demonstrates

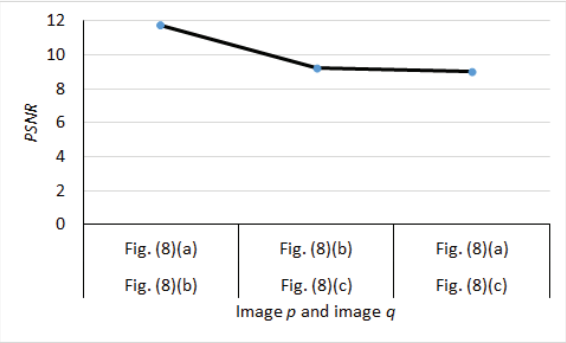

Figure 13 $PSNR$ analysis.

that security increases as the image moves from permutation to the diffusion stage. Empirical values of MSE listed in Table 3 prove that the proposed cryptosystems achieved considerably high MSE scores.

$$MSE(p,q) = \frac{1}{h\ w} \sum_{i=0}^{h-1} \sum_{j=0}^{w-1} [p(i,j) - q(i,j)]^2 \tag{5}$$

4.5 Peak-signal to Noise Ratio (PSNR) Analysis

Root Mean Squared Error ($RMSE$) of the two images, p and q is the square root of their MSE, as shown in Equation (6). Unlike MSE, it penalizes more the larger errors. $PSNR$ is an approximation to the human visual system perception of the obtained image quality. Equation (7) defines $PSNR$ using $RMSE$. It is the ratio of the maximum pixel signal to the $RMSE$. The maximum signal for the 8-bit depth grayscale image is 255. It is customary to express $PSNR$ in the logarithmic decibel (dB) scale.

Intuitive from the security perspective, the negatively-oriented $PSNR$ score offers more security. Table 5 contains $PSNR$ scores. Figure 13 shows the reduced $PSNR$ scores during encryption in the proposed system.

$$RMSE(p,q) = \sqrt{MSE(p,q)} \tag{6}$$

$$PSNR(p,q) = 20.\log_{10} \frac{255}{RMSE} \quad (dB) \tag{7}$$

When compared to $PSNR$ of the previous works in Table 3, our models scored significantly reduced $PSNR$ values.

4.6 Pearson Correlation

Equation (8) represents the Pearson correlation score r_{pq} $(-1 \leq r_{pq} \leq 1)$ used to obtain horizontal, vertical, and diagonal pixel correlation scores r_h, r_v, and r_d of the encrypted image for n random pixel samples.

$$\mathrm{r}_{pq} = \frac{\sum_{i=1}^{n}\left(\left(p_i - \frac{1}{n}\sum_{j=1}^{n} p_j\right) \times \left(q_i - \frac{1}{n}\sum_{\mathrm{j}=1}^{n} q_j\right)\right)}{\sqrt{\sum_{i=1}^{n}\left(p_i - \frac{1}{n}\sum_{j=1}^{n} p_j\right)^2 \times \sum_{i=1}^{n}\left(q_i - \frac{1}{n}\sum_{j=1}^{n} q_j\right)^2}} \tag{8}$$

Where p and q are the intensity values of an adjacent pixel pair. Optimal encryption results in highly de-correlated neighboring pixels in the cipher image. Table 3 contains the optimal r_h, r_v, and r_d indices achieved in the proposed accelerator-based systems.

4.7 NPCR and UACI

The number of Pixels Change Rate ($NPCR$) and Unified Average Changing Intensity ($UACI$) are the two estimations to evaluate the consequence of a bit change in the secret image on the encrypted image. A cryptosystem with $NPCR$ and $UACI$ values exceeding 33.4635% and 99.6094% is not vulnerable to differential abuse [32]. These two parameters are defined in Equations (9) and (10). Let q_1 and q_2 be the cipher images of the same secret image p before and after a single bit change.

$$NPCR = \frac{\sum_{i,j} q'(i,j)}{h\ w} \times 100\% \tag{9}$$

In Equation (9), $q'(i,j) = 1$, if $q_1(i,j) \neq q_2(i,j)$, otherwise $q'(i,j) = 0$.

$$UACI = \frac{1}{h\ w}\sum_{i,j}\frac{|q_1(i,j) - q_2(i,j)|}{255} \times 100\% \tag{10}$$

The Table 3 contains $NPCR$ and $UACI$ values of the proposed accelerator-based models and the previous works. These values indicate that a one-bit change of the secret image changes all bits in the encrypted image.

5 Conclusion

Just as important as how much security image encryption provides, it must do so in real-time. Although the logistic map development to secure image is

still underway, a new model to improve its efficiency is presented here. Permutation and diffusion are two main functions in the permutation-diffusion model logistic map encryption. This paper proposes a novel way of exploiting the data-parallel tasks in these two functions using an accelerator-the rest of the tasks assigned to the CPU because of their data dependency. Comparing the speeds obtained by creating two accelerator versions of the logistic map is made with the CPU-based logistic map. These models can be used for real-time applications because the speed achieved with the use of accelerators is enormous. The security metrics showed that the proposed system's security is progressive as the secret image transforms through its stages. Besides, the performance of the proposed accelerator models outperforms the results of the recent investigations. Applying these models to the color image could be interesting using reforming optimizations of the accelerator in the future.

References

[1] Marion E Hines. Image scrambling technique, October 28 1975. US Patent 3,914,877.

[2] Guanrong Chen, Yaobin Mao, and Charles K Chui. A symmetric image encryption scheme based on 3d chaotic cat maps. *Chaos, Solitons & Fractals*, 21(3):749–761, 2004.

[3] Lingfeng Liu and Suoxia Miao. A new image encryption algorithm based on logistic chaotic map with varying parameter. *SpringerPlus*, 5(1):289, 2016.

[4] Bin Wang, Yingjie Xie, Changjun Zhou, Shihua Zhou, and Xuedong Zheng. Evaluating the permutation and diffusion operations used in image encryption based on chaotic maps. *Optik*, 127(7):3541–3545, 2016.

[5] Gang He, Wenqing Wu, Li Nie, Jun Wen, Cheng Yang, and Wenxin Yu. An improved image multi-dimensional chaos encryption algorithm based on cuda. In *2019 9th International Conference on Information Science and Technology (ICIST)*, pages 183–187. IEEE, 2019.

[6] Qing Wu, Maksym Spiryagin, Colin Cole, and Tim McSweeney. Parallel computing in railway research. *International Journal of Rail Transportation*, 8(2):111–134, 2020.

[7] Xingyuan Wang and Chuanming Liu. A novel and effective image encryption algorithm based on chaos and dna encoding. *Multimedia Tools and Applications*, 76(5):6229–6245, 2017.

[8] Zhijuan Deng and Shaojun Zhong. A digital image encryption algorithm based on chaotic mapping. *Journal of Algorithms & Computational Technology*, 13:1748302619853470, 2019.

[9] Ashwin Raman. *Parallel processing of chaos-based image encryption algorithms*. PhD thesis, UC Irvine, 2016.

[10] Carlos Villaseñor, Eric F Gutierrez-Frias, Nancy Arana-Daniel, Alma Y Alanis, and Carlos Lopez-Franco. Parallel crossed chaotic encryption for hyperspectral images. *Applied Sciences*, 8(7):1183, 2018.

[11] A Sheik Abdullah, TGR Abiramie Shree, P Priyadharshini, and T Saranya. Algorithm and design techniques–a survey. *Global Journal of Computer Science and Technology*, 2019.

[12] Carlos Villaseñor, Javier Gomez-Avila, Nancy Arana-Daniel, Alma Y Alanis, and Carlos Lopez-Franco. Fast chaotic encryption for hyperspectral images. In *Processing and Analysis of Hyperspectral Data*. IntechOpen, 2019.

[13] Mouna Afif, Yahia Said, and Mohamed Atri. Computer vision algorithms acceleration using graphic processors nvidia cuda. *Cluster Computing*, pages 1–13, 2020.

[14] Sparsh Mittal and Shraiysh Vaishay. A survey of techniques for optimizing deep learning on gpus. *Journal of Systems Architecture*, 99:101635, 2019.

[15] Yuan Yuan, Xiaomin Yang, Wei Wu, Hu Li, Yiguang Liu, and Kai Liu. A fast single-image super-resolution method implemented with cuda. *Journal of Real-Time Image Processing*, 16(1):81–97, 2019.

[16] Bhabesh Deka, Sumit Datta, Helal Uddin Mullah, and Suman Hazarika. Diffusion-weighted and spectroscopic mri super-resolution using sparse representations. *Biomedical Signal Processing and Control*, 60:101941, 2020.

[17] Leila Habibpour, Shamim Yousefi, M Zolfy Lighvan, and Hadi S Aghdasi. 1d chaos-based image encryption acceleration by using gpu. *Indian journal of science and technology*, 9(6):19–25, 2016.

[18] Wai-Kong Lee, Raphael C-W Phan, Wun-She Yap, and Bok-Min Goi. Spring: a novel parallel chaos-based image encryption scheme. *Nonlinear Dynamics*, 92(2):575–593, 2018.

[19] Wai Kong Lee. *High Speed Computation Of Advanced Cryptographic Algorithms On Massively Parallel Architecture*. PhD thesis, UTAR, 2018.

[20] Aryan Saxena, Vatsal Agrawal, Rajdeepa Chakrabarty, Shubhjeet Singh, and J Saira Banu. Accelerating image encryption with aes using gpu: A

quantitative analysis. In *International Conference on Intelligent Systems Design and Applications*, pages 372–380. Springer, 2018.

[21] Lin You, Ersong Yang, and Guangyi Wang. A novel parallel image encryption algorithm based on hybrid chaotic maps with opencl implementation. *Soft Computing*, pages 1–15, 2020.

[22] Neha Kishore and Priya Raina. Parallel cryptographic hashing: Developments in the last 25 years. *Cryptologia*, 43(6):504–535, 2019.

[23] Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. Scaling neural machine translation. *arXiv preprint arXiv:1806.00187*, 2018.

[24] Eric Weisstein. *Logistic Map – from Wolfram MathWorld*, 2020(accessed June 24, 2020). https://mathworld.wolfram.com/LogisticMap.html.

[25] Huanzhou Zhu, Ligang He, Matthew Leeke, and Rui Mao. Wolfgraph: The edge-centric graph processing on gpu. *Future Generation Computer Systems*, 111:552–569, 2020.

[26] Álvaro Salinas, Claudio Torres, and Orlando Ayala. A fast and efficient integration of boundary conditions into a unified cuda kernel for a shallow water solver lattice boltzmann method. *Computer Physics Communications*, 249:107009, 2020.

[27] Håvard H Holm, André R Brodtkorb, and Martin L Sætra. Gpu computing with python: Performance, energy efficiency and usability. *Computation*, 8(1):4, 2020.

[28] Shahrukh Athar and Zhou Wang. A comprehensive performance evaluation of image quality assessment algorithms. *Ieee Access*, 7:140030–140070, 2019.

[29] Joshua Caleb Dagadu, Jian-Ping Li, Fadia Shah, Nadir Mustafa, and Kamlesh Kumar. Dwt based encryption technique for medical images. In *2016 13th International computer conference on wavelet active media technology and information processing (ICCWAMTIP)*, pages 252–255. IEEE, 2016.

[30] Nabil Ben Slimane, Kais Bouallegue, and Mohsen Machhout. Nested chaotic image encryption scheme using two-diffusion process and the secure hash algorithm sha-1. In *2016 4th International Conference on Control Engineering & Information Technology (CEIT)*, pages 1–5. IEEE, 2016.

[31] Jansher Khan, Jawad Ahmad, and Seong Oun Hwang. An efficient image encryption scheme based on: Henon map, skew tent map and s-box. In *2015 6th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO)*, pages 1–6. IEEE, 2015.

[32] M Essaid, I Akharraz, A Saaidi, and A Mouhib. A new image encryption scheme based on confusion-diffusion using an enhanced skew tent map. *Procedia Computer Science*, 127:539–548, 2018.
[33] Zhongyun Hua, Binghang Zhou, and Yicong Zhou. Sine chaotification model for enhancing chaos and its hardware implementation. *IEEE Transactions on Industrial Electronics*, 66(2):1273–1284, 2018.
[34] Xiaoling Huang and Guodong Ye. An image encryption algorithm based on time-delay and random insertion. *Entropy*, 20(12):974, 2018.
[35] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Biographies

M. Raviraja Holla is working as Assistant Professor in the Department of Information and Communication Technology Department, Manipal Institute of Technology (a constituent institution of Manipal Academy of Higher Education), Manipal. He completed B.E.(CSE) from Bangalore University, India and M.Tech.(CSE) from KSOU Mysore, India. He is currently pursuing Ph.D. from the Department of Computer Engineering, National Institute of Technology Karnataka (NITK), Surathkal. His areas of interest include Information Security, High-Performance Computing, and Semantic Web.

Alwyn R. Pais is the head of Department of Computer Engineering, National Institute of Technology Karnataka (NITK) as well as an Associate Professor. He completed his B.Tech.(CSE) from Mangalore University, India, M.Tech. (CSE) from IIT Bombay, India, and PhD (CSE) in NITK, Surthkal. His area of interest includes Information Security, Image Processing and Computer Vision.

D. Suma is working as Assistant Professor in the Department of Computer Science and Engineering, Manipal Institute of Technology (a constituent institution of Manipal Academy of Higher Education), Manipal. She completed B.E.(ECE) from Kuvempu University, India and M.Tech.(CSE) from Visvesvaraya Technological University, India. Her areas of interest include Object Oriented Programming, High-Performance Computing, and Data Mining.