IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 23 June 2024, accepted 18 July 2024, date of publication 22 July 2024, date of current version 6 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3432008

RESEARCH ARTICLE

A Hardware-Accelerated Approach to Chaotic Image Encryption: LTB Map and FPGA Implementation

MOHAMED YAMNI[1], ACHRAF DAOUI[2], PAWEŁ PŁAWIAK[3,4], OSAMA ALFARRAJ[5], AND AHMED A. ABD EL-LATIF[6,7], (Senior Member, IEEE)

[1]Dhar El Mahrez Faculty of Science, University of Sidi Mohamed Ben Abdellah, Fes 30003, Morocco
[2]National School of Applied Sciences, University of Sidi Mohamed Ben Abdellah, Fes 30003, Morocco
[3]Department of Computer Science, Faculty of Computer Science and Telecommunications, Cracow University of Technology, 31-155 Krakow, Poland
[4]Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, 44-100 Gliwice, Poland
[5]Computer Science Department, Community College, King Saud University, Riyadh 11437, Saudi Arabia
[6]Jadara University Research Center, Jadara University, Irbid 21110, Jordan
[7]Department of Mathematics and Computer Science, Faculty of Science, Menoufia University, Shebeen El-Kom 32511, Egypt

Corresponding author: Ahmed A. Abd El-Latif (ahmedabdellatif@ieee.org)

This work was supported by the Researchers Supporting Project, King Saud University, Riyadh, Saudi Arabia, under Grant RSP2024R102.

ABSTRACT The need for high-speed and secure image encryption has become increasingly critical in the digital age, driven by the rapid growth of digital data, the rapid advancement of internet technologies, and the limitations of traditional encryption algorithms. This paper addresses this need by proposing a novel high-speed and secure image encryption that leverages the flexibility and parallelism of Field-Programmable Gate Arrays (FPGAs). Firstly, we present the LTB map, an enhanced chaotic map that synergistically combines three distinct 1D chaotic maps (the Logistic, Tent, and Bernoulli shift maps) into a single framework. The resulting LTB map exhibits an expanded key space and a heightened level of complexity and unpredictability compared to its constituent maps operating independently. Furthermore, the LTB map adopts a 32-bit fixed-point representation and involves simple operations like addition, subtraction, and multiplication, resulting in a resource-efficient implementation on FPGA platform. Additionally, we propose a new image encryption algorithm based on the LTB map. The LTB map plays a pivotal role in scrambling pixel values and positions during the confusion and diffusion operations, thereby ensuring robust security. By leveraging the inherent parallelism of the LTB map and parallelizing encryption operations, coupled with efficient hardware implementation on FPGAs, our encryption method achieves high-speed performance while maintaining strong security properties.

INDEX TERMS FPGA implementation, chaotic maps, LTB map, high-speed image encryption.

I. INTRODUCTION

In recent years, securing multimedia information such as images has become a major concern due to the rapid advancement of internet technologies. Traditional encryption algorithms like the Data Encryption Standard (DES) or the Advanced Encryption Standard (AES) often prove ineffective in adequately securing digital images [1]. These images stand out due to their significant volume and high redundancy, necessitating specific encryption approaches. On the other hand, methods based on chaotic systems offer promising solutions to address the security needs of digital images. Chaotic systems possess unique properties, including sensitivity to initial conditions, unpredictability, and non-periodicity of motion trajectories. These characteristics make them well-suited for both confusion and diffusion in cryptography, thus providing robust protection against cyber-attacks and ensuring the confidentiality of data.

Discrete chaotic systems operate in discrete time steps, unlike continuous chaotic systems, which require an additional discretization step to function in digital environments. This discrete operation allows for precise control and

The associate editor coordinating the review of this manuscript and approving it for publication was Ramakrishnan Srinivasan.

synchronization, making them advantageous for digital implementation and secure communication protocols [2], [3], [4].

Multi-dimensional chaotic maps, spanning 2D, 3D, 4D systems, etc., are characterized by complex dynamic behavior and have gained prominence due to their heightened security potential. Examples include the 2D Henon map [5], 2D Sine Logistic map [6], 2D Sine Cosine Logistic chaotic map [7], 2D modular sine-cosine map (2D-MSCM) [8], 2D fractional-order Meixner polynomials map (2D-FrMP) [9], 3D Sine map [10], 3D Quantum logistic map [11], and 4D Uruk chaotic map [4]. Due to their enhanced security features, multi-dimensional chaotic maps have been adopted in recent image encryption algorithms [4], [12], [13], [14], [15], [16], [15], [17], [18].

On the other hand, 1D chaotic systems offer simplicity of structure, fast processing speed, and minimal hardware resource consumption (such as memory, physical resources, processing power, and energy), making them attractive for various practical applications in safeguarding information [19]. Notable examples include the Logistic map [20], Sine map [21], Chebychev map [22], Tent map [23], and Bernoulli shift map [24]. Despite their appeal, these systems face inherent limitations, such as (1) a restricted number of control parameters and initial conditions; (2) a limited range of parameters inducing chaotic behavior; (3) phase diagrams characterized by simple and regular trajectories; and (3) a level of randomness and instability confined within specific intervals of their control parameters. Consequently, security techniques relying on these maps, such as data encryption, data hiding, and steganography, are susceptible to brute force attacks [25].

Several researchers have examined the limitations of 1D chaotic systems and have either improved existing chaotic maps or designed new maps with enhanced chaotic characteristics [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35] (see Table 1). However, these efforts often do not take into account the hardware resource consumption (such as the number of logic elements, DSPs, memory, processing power, and energy) of chaotic maps and corresponding image encryption algorithms. Moreover, most chaotic systems and encryption algorithms resulting from these efforts are implemented on standard PCs equipped with Central Processing Units (CPUs) and Graphics Processing Units (GPUs). Indeed, traditional computing technologies, such as CPUs and GPUs, have limitations in terms of performance and flexibility. This makes them less suitable for high-performance computing (HPC) applications, which involve processing vast amounts of data and executing complex calculations at incredibly high speeds.

A promising approach would be the use of alternative hardware platforms, such as Field-Programmable Gate Arrays (FPGAs). FPGAs offer unique advantages for high-performance computing, including customizable circuit design, low latency, high degree of parallelism, and high energy efficiency. These characteristics could lead to more efficient implementations of chaotic algorithms and real-time encryption algorithms. Existing works [3], [36], [37], [38], [39], [40], [41], [42], [43], [44] have successfully implemented chaotic systems and chaotic system-based image encryption algorithms on FPGAs, taking into account hardware resource consumption and energy efficiency.

Other research efforts have also explored the implementation of chaotic systems and encryption algorithms on various hardware platforms, such as User-Programmable Analog Arrays (FPAA) [45] and the STM32 microcontroller [46], [47]. However, it is worth noting that FPAA requires an analog-to-digital converter (ADC) for digital applications, and they have limitations in terms of flexibility, precision, and processing capabilities compared to FPGAs. Additionally, STM32 microcontrollers are generally more limited in terms of computing power, available resources, and flexibility due to their closed and non-reconfigurable architecture, compared to FPGAs.

The main objective of this paper is to design and implement an enhanced 1D chaotic map, along with a new image encryption algorithm optimized for high-speed performance, on hardware using an FPGA, capitalizing on the inherent advantages offered by FPGA-based implementations.

In this paper, we present the Logistic-Tent-Bernoulli (LTB) map, an enhanced chaotic map that synergistically combines three distinct 1D chaotic maps – the Logistic, Tent, and Bernoulli shift maps – into a single framework. These maps are combined using the bitwise XOR operation between the binary representations of the outputs of the logistic map, tent map, and absolute values of Bernoulli shift map outputs. The resulting LTB map exhibits a heightened level of complexity and unpredictability compared to its constituent maps operating independently. Unlike chaotic LTB kernel maps where each map has only 2 parameters, the LTB map has 6 parameters, including 3 initial conditions and 3 control parameters. This advantage translates into a significantly expanded key space, increased complexity and enhanced security when employing the LTB map in an encryption algorithm. The robustness and unpredictability of the LTB map are confirmed through phase and entropy diagrams, as well as extensive testing using NIST and TestU01 suites. Moreover, the LTB map adopts a 32-bit fixed-point representation and involves elementary arithmetic operations such as addition, subtraction, and multiplication, resulting in a resource-efficient implementation on FPGA platform.

Subsequently, we present a high-speed image encryption algorithm based on the LTB map, implemented on an FPGA hardware platform. The algorithm employs the LTB map for pixel value and position scrambling during both confusion and diffusion operations. These operations are carried out along the rows and columns of the image, using two separate LTB maps with different keys. By leveraging the FPGA's parallel processing capabilities, confusion and diffusion are

TABLE 1. Chaotic maps and their parameters, applications, and implementation environments.

Work	Used chaotic map	Number of control parameters and initial conditions	Application	Implementation environment
Ref. [26]	Improved Logistic-sine map	3	Local image encryption	PC with a CPU
Ref. [27]	New 1D fractional chaotic map	3	Image encryption	PC with a CPU
Ref. [28]	Improved sine-tangent map	2	Image encryption	PC with a CPU
Ref. [29]	New 1D cosine fractional chaotic map	3	Image encryption	PC with a CPU
Ref. [30]	New memristive chaotic map	2	Image encryption	PC with a CPU
Ref. [31]	Quadratic map cluster	4	Image encryption	PC with a CPU
Ref. [32]	Piece-wise quadratic polynomial chaotic map	4	Image encryption	PC with a CPU
Ref. [33]	Multiparametric tent map	7	Image encryption	PC with a CPU
Ref. [11]	Quantum logistic map	5	Image encryption	PC with a CPU
Ref. [34]	Multiparametric piecewise linear chaotic map	9	Image compression-encryption	PC with a CPU
Ref. [35]	Multi-Parameter Chebyshev map	5	Image steganography	PC with a CPU
Ref. [48]	Modified Henon map	6	Audio watermarking	Raspberry Pi cluster
Ref. [49]	Continuous 3D hyperbolic sine chaotic map	3	Image encryption	NanoPC-T3 board
Ref. [46]	New 2D hyperchaotic map	5	Without app	STM32 microcontroller
Ref. [47]	Modified logistic map	4	Speech encryption	STM32 microcontroller
Ref. [45]	3D chaotic map	5	Without app	FPAA
Ref. [3]	Piecewise linear map	2	Without app	FPGA platform
Ref. [36]	TRNG based on logisitc map	2	Without app	FPGA platform
Ref. [37]	Logistic map	2	Image encryption	FPGA platform
Ref. [38]	Fractional logistic map	3	Image encryption	FPGA platform
Ref. [39]	New discrete 3D chaotic map	3	Image encryption	FPGA platform
Ref. [41]	3D Nahrain chaotic map	5	Image encryption	FPGA platform
Our work	LTB map	6	Image encryption	FPGA platform

performed simultaneously, thereby enhancing the efficiency and speed of the encryption process. This encryption method capitalizes on the inherent parallelism of the LTB map, coupled with parallelization of encryption operations and efficient hardware implementation on the FPGA platform, culminating in remarkable high-speed encryption performance.

The main contributions of this paper are:

- An enhanced 1D chaotic map is presented, denoted as the LTB map, created by combining the Logistic map, the Tent map, and the Bernoulli shift map, contributing to an expanded key space and enhanced complexity.
- The LTB map demonstrates improved robustness and unpredictability in chaotic sequence generation, evident through enhanced complexity in the phase diagram and high entropy values.
- Utilizing simple arithmetic operations like addition, subtraction, and multiplication, coupled with fixed-point representation, the LTB map ensures efficient resource usage, particularly advantageous for FPGA implementation.
- Proposal and implementation of a new image encryption algorithm based on the LTB map on an FPGA platform, offering both high encryption efficiency and high-speed processing.
- Harnessing FPGA parallelism, LTB maps and encryption operations (confusion and diffusion) are executed simultaneously, significantly enhancing both efficiency and speed of the encryption process.

The remainder of the article is organized as follows. In Section II, an introduction to the FPGA board is

provided, highlighting its advantages and the necessary steps to implement an algorithm on it. Section III presents the 1D chaotic maps used in this work and their limitations. The definition of the LTB map and its implementation on the FPGA are described in Section IV, followed by the proposed image encryption algorithm and its implementation on the FPGA in Section V. Experimental results and security analysis are presented in Section VI, while Section VII concludes this paper.

II. FPGA PLATFORM

Compared to traditional computing devices equipped with CPUs and GPUs, FPGAs offer several advantages in terms of efficiency and performance. Firstly, FPGAs provide increased flexibility and reusability; they can be programmed and reprogrammed to implement custom hardware circuits, allowing for the execution of specific algorithms much faster than CPUs and GPUs. This versatility reduces the need for purchasing additional specialized hardware for specific tasks, potentially lowering overall costs. Additionally, FPGAs are capable of parallel processing, enabling them to perform multiple calculations simultaneously. They are also more energy-efficient, consuming less power and generating less heat than CPUs and GPUs. Furthermore, FPGAs offer low-latency processing, allowing them to perform calculations with minimal delay. Finally, their customization capability allows for the creation of hardware circuits tailored to specific needs, which is not possible with the general-purpose processors of CPUs and GPUs. These advantages make FPGAs an ideal choice for real-time applications and tasks requiring high computing power while minimizing energy consumption and heat generation.

FPGAs have found applications across diverse domains, including image processing [50], Internet of Things (IoT) [51], traffic classification [52], and numerous other fields.

In this work, we utilized Intel's Cyclone II FPGA platform, featuring device number EP2C35F672C6, to implement our proposed algorithms (LTB map and image encryption algorithm). Hardware resources of this device are presented in Table 2.

The hardware implementation of our algorithms involves several necessary steps. Firstly, each algorithm is translated from high-level programming language (Matlab language) into a hardware description language (HDL). In this work, we opt Verilog-HDL to describe the hardware behavior of the algorithm. This translation involves breaking down the algorithm into fundamental operations and expressing them using synthesizable HDL constructs. Next, the designed Verilog code is synthesized using Altera's Quartus II 9.0 software. This process analyzes the code, identifies its hardware components, and optimizes for the target FPGA board. Once synthesized, the code undergoes simulation using ModelSim simulation tool to verify the functionality of the designed circuit, ensuring it performs as expected before being physically implemented on the FPGA. Finally, the verified design is downloaded onto the FPGA using Quartus II. This process configures the internal logic of the FPGA to match the designed circuit, enabling it to perform the algorithm in hardware.

TABLE 2. Hardware resources of Intel's cyclone II FPGA platform (Device EP2C35F672C6).

Resources	Available
Logic elements	33216
M4K RAM blocks	105
Total I/O pins	475
Total memory bits	483840
Embedded Multiplier 9-bit elements	70
Phase-Locked Loops (PLLs)	4
Clock inputs	50-MHz oscillator and 27-MHz oscillator

III. 1D CHAOTIC MAPS

Discrete chaotic systems, particularly 1D systems, stand out for their simple structure and fast operation compared to continuous chaotic systems. Additionally, due to their discrete nature, their electronic implementation on FPGAs consumes fewer resources than that of continuous chaotic systems, which require an additional discretization step. Among the popular 1D chaotic maps that do not involve complex mathematical operations like division or trigonometric functions, we find the Logistic map [20], the Tent map [53], and the Bernoulli shift map [24]. These maps are characterized by simple operations such as addition, subtraction, and multiplication, making their implementation on an FPGA less resource-intensive. For this reason, these maps are used in this work as kernel functions in the design of the enhanced LTB map. In this section, we provide a brief overview of these chaotic maps and highlight their behavioral characteristics.

A. LOGISTIC MAP

The logistic map is a 1-D discrete-time dynamical system defined by the following equation [20]:

$$x_{i+1} = \lambda_1 x_i (1 - x_i) \tag{1}$$

where $x_i \in (0, 1)$ is the current value in the ith iteration, x_{i+1} is the next value, and $\lambda_1 \in (0, 4]$ is a is the control parameter influences the map's behavior.

Figure 1 presents a comprehensive exploration of the dynamics of the logistic map for initial conditions $x_0 = 0.4$ and when the parameter λ_1 changes. The Figure 1.a presents the Lyapunov exponents, quantifying the chaotic behavior of the system. Positive values in the Lyapunov exponent plot indicate chaotic behavior, emphasizing regions of unpredictability. Figure 1.a shows that the logistic map exhibits chaotic behavior for certain values of λ_1, especially when it is between 3.54 and 4. This is clearly shown in Figure 1.b, which presents the bifurcation diagram of the logistic map, where the appearance of dense sets in the interval $\lambda_1 \in [3.54, 4]$ is an indicator of chaotic regions.

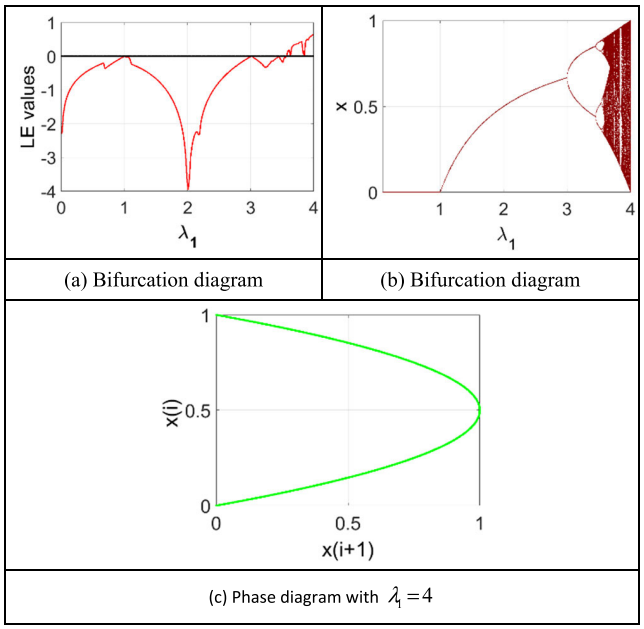


FIGURE 1. Dynamics of logistic map for initial condition $x_0 = 0.4$ and when λ_1 changes.

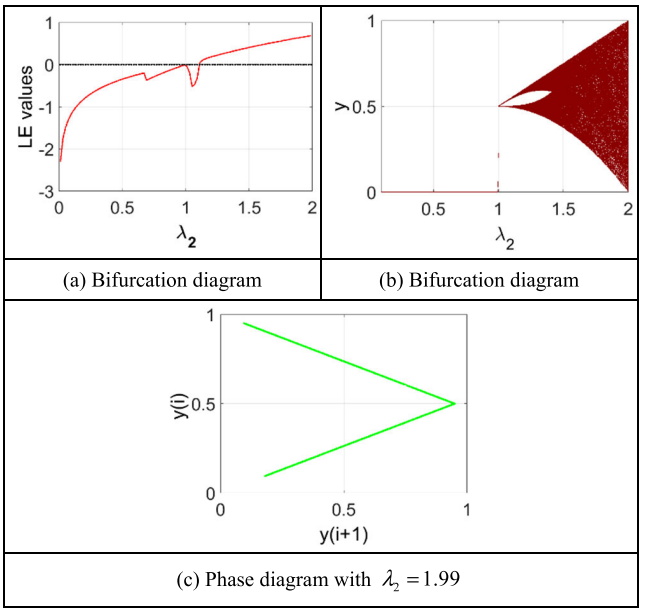


FIGURE 2. Dynamics of Tent map for initial condition $y_0 = 0.4$ and when λ_2 changes.

The Figure 1.c shows the phase diagram of the logistic map, which plots the value of x_i against x_{i+1} for each iteration ith. We can see from this figure that the phase diagram has a simple, regular and orderly trajectory, and can therefore be predicted.

B. TENT MAP

The tent map is a piecewise linear discrete dynamical system defined by [53]:

$$y_{i+1} = \begin{cases} \lambda_2 y_i & \text{if } 0 \le y_i < 0.5 \\ \lambda_2(1 - y_i) & \text{if } 0.5 \le y_i \le 1 \end{cases} \quad (2)$$

here $y_i \in (0, 1)$ represents the value in the ith iteration and $\lambda_2 \in (0, 2)$ a is the control parameter influencing the map's behavior.

Figure 2.a demonstrates that the tent map displays chaotic behavior within the interval $\lambda_2 \in [1.2, 2)$, as indicated by a positive Lyapunov exponent (LE > 0) and the presence of a dense set of points in the corresponding interval of the bifurcation diagram Figure 2.b. Finally, the phase diagram, illustrated in Figure 2.c, reveals a simplicity in the visualization of the map's dynamic evolution.

C. BERNOULLI SHIFT MAP

The Bernoulli shift map is another 1-D chaotic map defined by the following equation [24]:

$$z_{i+1} = \begin{cases} \lambda_3 z_i + 1 & \text{for } z_i < 0 \\ \lambda_3 z_i - 1 & \text{for } z_i \ge 0 \end{cases} \quad (3)$$

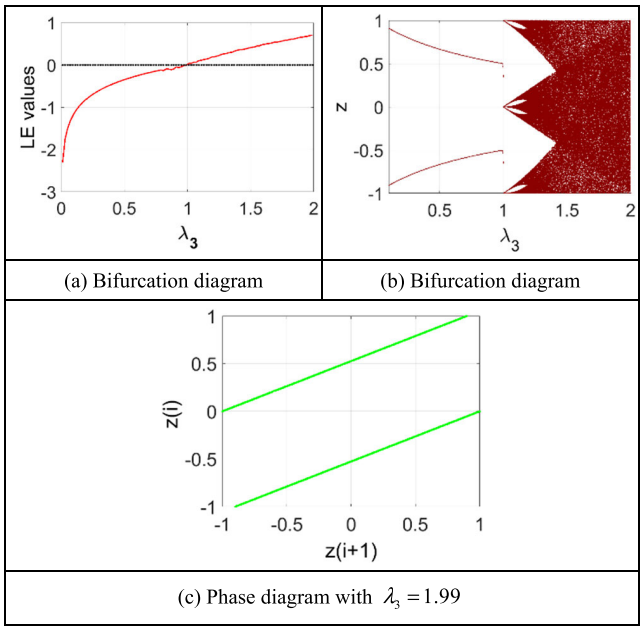
where $z_i \in [-1, 1]$ is the current value and $\lambda_3 \in (0, 2)$ is the control parameter that determines the behavior of the map.

The Lyapunov exponents graph, the bifurcation diagram, and phase diagram of the Bernoulli shift map are depicted in Figure 3 for initial condition $z_0 = 0.4$ and when the parameter λ_3 changes. The Lyapunov exponents graph (Figure 3.a) reveals positive values within the interval $\lambda_3 \in [1, 2)$, indicating chaotic behavior (LE >0). Concurrently, Figure 3.b illustrates the presence of a uniform distribution, represented by a dense set of points, within the interval $\lambda_3 \in [1.4, 2)$ in the bifurcation diagram. This alignment suggests the existence of chaotic regions characterized by unpredictability and complexity in the interval $\lambda_3 \in [1.4, 2)$. Similar to the logistic map and the tent map, the Bernoulli shift map exhibits a simple, regular, and orderly trajectory in its phase diagram, as depicted in Figure 3.c.

Despite their widespread use, the Logistic map, Tent map, and Bernoulli shift map have significant limitations that impact their effectiveness. These include a restricted number of control parameters and initial values, a limited range of parameters inducing chaotic behavior, and the simplicity, regularity, and predictability of their trajectories. As a result, information security techniques relying on these maps, such as data encryption, data hiding, and steganography, are vulnerable to brute force attacks [25].

IV. ENHANCED LTB MAP

To address the issues highlighted in Section II, this section presents an enhanced chaotic map, a cluster composed of multiple 1D chaotic systems, known as the LTB map. This LTB map combines three different 1D chaotic systems: the Logistic map, the Tent map, and the Bernoulli shift map. For this combination, the bitwise XOR operation is applied between the binary representations of the outputs of the Logistic map, the Tent map, and the absolute values of the Bernoulli shift map outputs.



(a) Bifurcation diagram (b) Bifurcation diagram

(c) Phase diagram with $\lambda_3 = 1.99$

FIGURE 3. Dynamics of Bernoulli shift for initial condition $z_0 = 0.4$ and when λ_3 changes.

A. LTB MAP MODEL

The enhanced LTB model, depicted in Figure 4, is defined as follows:

$$q_{i+1} = x_{i+1} \oplus y_{i+1} \oplus |z_{i+1}| \tag{4}$$

where $q_i \in [0, 1]$ represents the current value the LTB map, q_{i+1} is the next value, $\oplus$ denotes the bitwise XOR operation performed on the binary representations of x_{i+1} (Eq. 1), y_{i+1} (Eq. 2), and z_{i+1} (Eq. 3), and $|.|$ is floor operator (Round toward negative infinity).

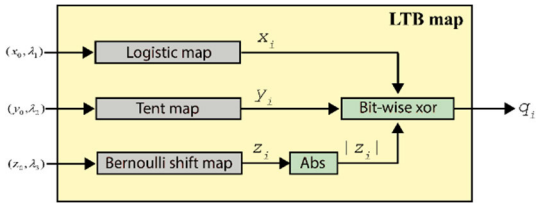
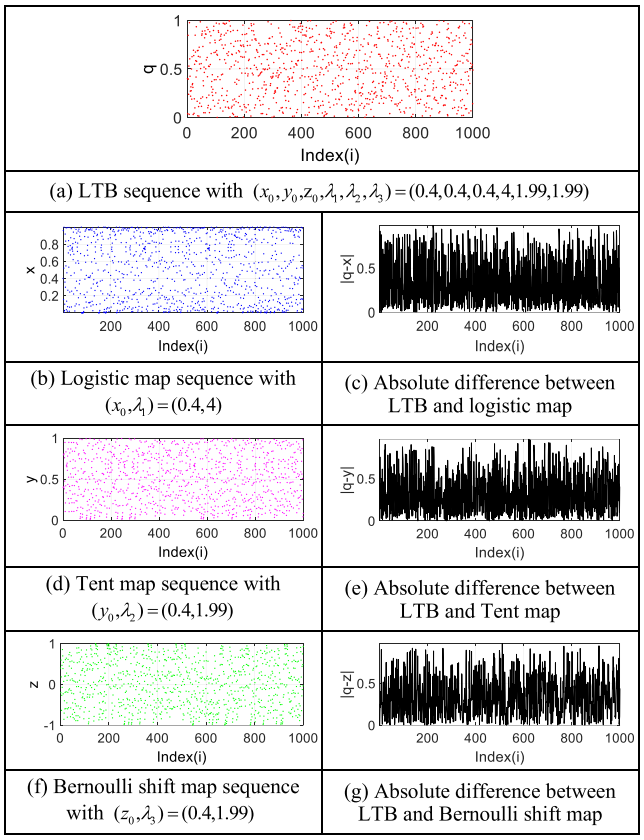


FIGURE 4. Enhanced LTB model.

By substituting x_{i+1}, y_{i+1}, and z_{i+1} with their expressions in Eq. (4), we have (5), as shown at the bottom of the next page, where λ_1, λ_2, and λ_3 are the control parameters of the LTB map with the ranges of $\lambda_1 \in (0, 4]$, $\lambda_2 \in (0, 2)$, and $\lambda_3 \in (0, 2)$, and x_0, y_0, and z_0 are the initial values of the LTB map with the range $x_0, y_0 \in (0, 1)$ and $z_0 \in [-1, 1]$.

The LTB map is designed to combine the properties of the Logistic, Tent, and Bernoulli shift maps. As shown in Figure 1 and described by Equations 4 and 5, these three chaotic maps can be executed simultaneously. For each iteration, the outputs of the three chaotic maps are generated concurrently, and the LTB output is produced in a single iteration. Therefore, the LTB map can fully exploit



(a) LTB sequence with $(x_0, y_0, z_0, \lambda_1, \lambda_2, \lambda_3) = (0.4, 0.4, 0.4, 4, 1.99, 1.99)$

(b) Logistic map sequence with $(x_0, \lambda_1) = (0.4, 4)$

(c) Absolute difference between LTB and logistic map

(d) Tent map sequence with $(y_0, \lambda_2) = (0.4, 1.99)$

(e) Absolute difference between LTB and Tent map

(f) Bernoulli shift map sequence with $(z_0, \lambda_3) = (0.4, 1.99)$

(g) Absolute difference between LTB and Bernoulli shift map

FIGURE 5. Comparison of chaotic sequences (of length 1000 iterations) generated by LTB and traditional chaotic maps using floating-point arithmetic according to the standards of the IEEE.

the parallel processing capabilities of FPGA architectures, enhancing the overall speed and efficiency of LTB map-based encryption. Additionally, through this combination, the LTB map has six parameters, including three initial conditions and three control parameters. This is a significant increase compared to the individual chaotic maps, each of which has only two parameters. This advantage translates into a substantially expanded key space, increased complexity, and enhanced security when employing the LTB map in an encryption algorithm.

Unlike many enhanced 1D chaotic maps that rely on complex mathematical operations such as division [27], [29], [32], trigonometric functions [26], [29], [28], [33], [34], [35], [30] modulus operation [54], [55], exponential function [38], Gamma function [38], which are complicated to implement on FPGA, our LTB map utilizes simple operations like addition, subtraction, and multiplication. This simplicity makes the implementation on an FPGA less resource-intensive.

Prior to implementing the enhanced LTB map on the FPGA, a series of tests were conducted using Matlab R2022b on a PC equipped with an Intel Core i3 CPU @ 2.40GHz and 6 GB of RAM. These tests utilized floating-point arithmetic conforming to IEEE standards.

Figure 5 presents the plots of the first 1000 samples generated by the LTB map with parameters (x_0, y_0, z_0, λ_1,

$\lambda_2, \lambda_3) = (0.4, 0.4, 0.4, 4, 1.99, 1.99)$, the logistic map with parameters $(x_0, \lambda_1) = (0.4, 4)$, the tent map with parameters $(y_0, \lambda_2) = (0.4, 1.99)$, and the Bernoulli shift map with parameters $(z_0, \lambda_3) = (0.4, 1.99)$. Corresponding histograms of each map are displayed in Figure 6 to observe their distributions. It is evident from Figure 5 that the LTB map generates a distinct chaotic sequence compared to the other maps. This distinction is further highlighted by the varying distributions depicted in the histograms (Figure 6), solidifying the notion that the LTB represents a new chaotic map.

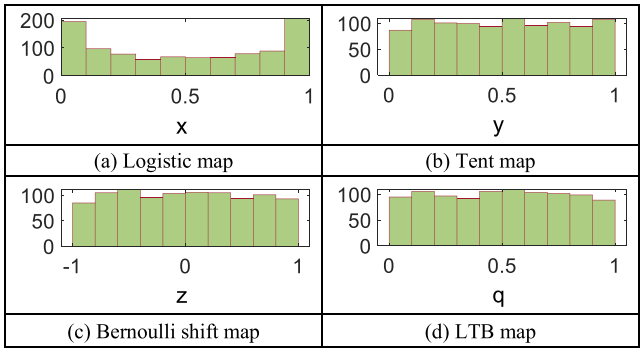


FIGURE 6. Histograms of LTB map and traditional chaotic maps for 1000 Iterations using floating-point arithmetic according to the standards of the IEEE.

Figure 7 illustrates the phase diagram of the LTB map. In contrast to the Logistic map (Figure 1.c), the Tent map (Figure 2.c), and the Bernoulli shift map (Figure 3.c), which typically exhibit simple and orderly trajectories in their phase diagrams, the phase diagram of the LTB map reveals a complex and intricate structure. Dense regions of points and an absence of discernible patterns or regularity suggest heightened levels of randomness in the dynamics of the LTB map.

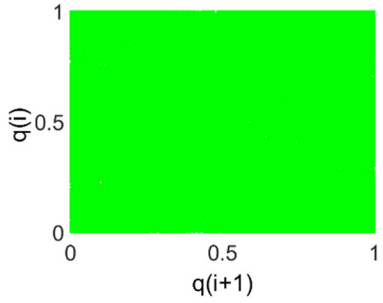


FIGURE 7. Phase diagram of the LTB map.

Figure 8 presents the results of information entropy (Eq. 12) as a function of control parameter values for the Logistic map, the tent map, the Bernoulli shift map, and the LTB map. Information entropy measures the randomness of chaotic maps and the instability of data, peaking when chaotic sequences are distributed randomly. Higher entropy values indicate a more optimal distribution pattern, ideally approaching 8. This visual comparison in Figure 8 highlights the superior distribution properties of the LTB map, where its entropy values are consistently high and close to 8 regardless of the values of its control parameters. In contrast, the logistic, tent, and Bernoulli maps demonstrate relatively high entropy values only within specific intervals of their control parameters. This suggests that the LTB map exhibits a more robust and uniformly random behavior across a wider range of parameter values, thereby affirming its superiority over traditional chaotic maps.

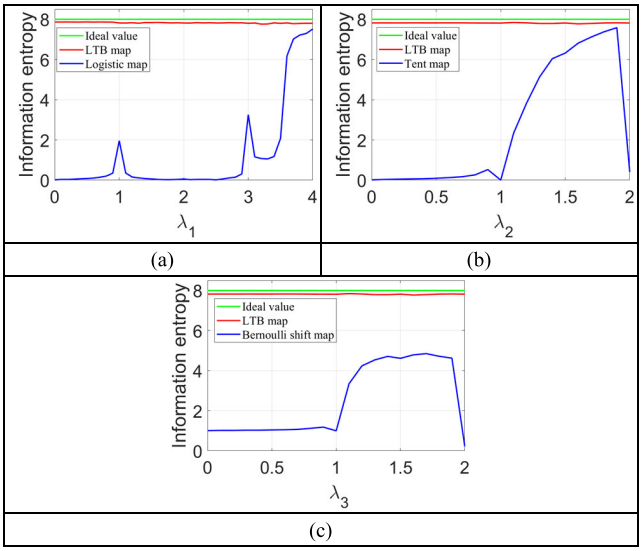
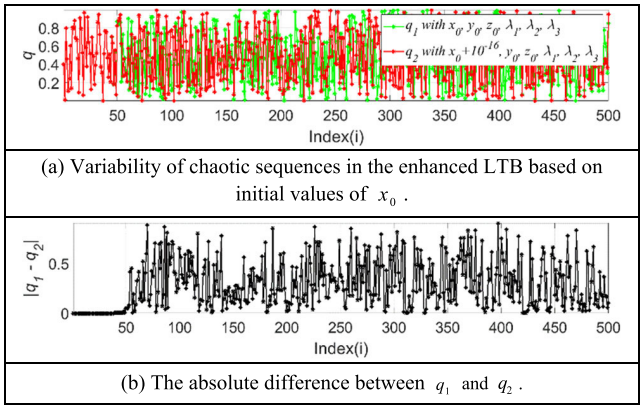


FIGURE 8. The information entropy diagram of the (a) LTB map and Logistic map, (b) LTB map and Tent map, and (c) LTB map and Bernoulli shift map.

To illustrate the sensitivity of our LTB map to initial conditions, we generate two output sequences: one with the initial conditions $(x_0, y_0, z_0, \lambda_1, \lambda_2,$ and $\lambda_3)$ and the other with a minute variation in one of the initial values. The results of this test are presented in Figures (10)-(15). From these figures, it is evident that beyond the transient regime (after the 50th sample), our system produces significantly different outputs, demonstrating consistent chaotic behavior. This occurs even with minute changes of 10^{-16} for $x_0, y_0,$ and z_0, and 10^{-15} for $\lambda_1, \lambda_2,$ and λ_3.

Table 3 compares the sensitivity to initial conditions of the enhanced LTB with that of three other chaotic

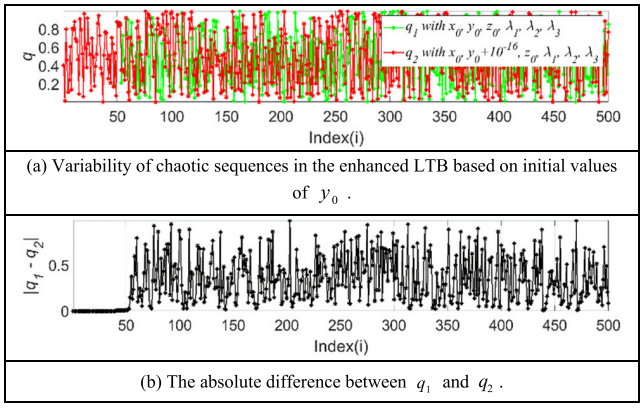
$$q_{i+1} = \begin{cases} \lambda_1 x_i(1-x_i) \oplus \lambda_2 y_i \oplus |\lambda_3 z_i + 1| & \text{if } y_i < 0.5 \& \ z_i < 0 \\ \lambda_1 x_i(1-x_i) \oplus \lambda_2 y_i \oplus |\lambda_3 z_i - 1| & \text{if } y_i < 0.5 \& \ z_i \geq 0 \\ \lambda_1 x_i(1-x_i) \oplus \lambda_2 (1-y_i) \oplus |\lambda_3 z_i + 1| & \text{if } y_i \geq 0.5 \& \ z_i < 0 \\ \lambda_1 x_i(1-x_i) \oplus \lambda_2 (1-y_i) \oplus |\lambda_3 z_i - 1| & \text{if } y_i \geq 0.5 \& \ z_i \geq 0 \end{cases} \quad (5)$$



(a) Variability of chaotic sequences in the enhanced LTB based on initial values of x_0 .

(b) The absolute difference between q_1 and q_2 .

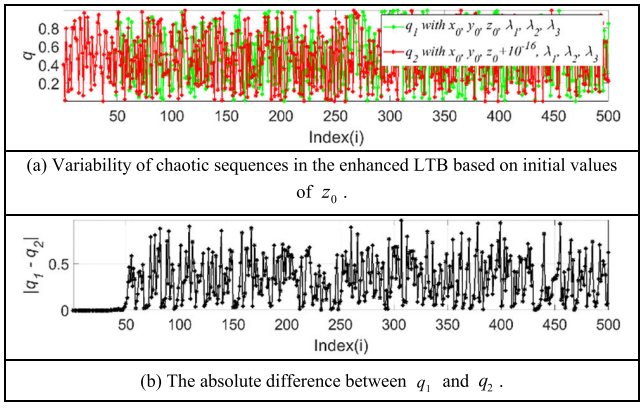
FIGURE 9. Sensitivity analysis of the enhanced LTB to parameter x_0 variation by 10^{-16}.



(a) Variability of chaotic sequences in the enhanced LTB based on initial values of y_0 .

(b) The absolute difference between q_1 and q_2 .

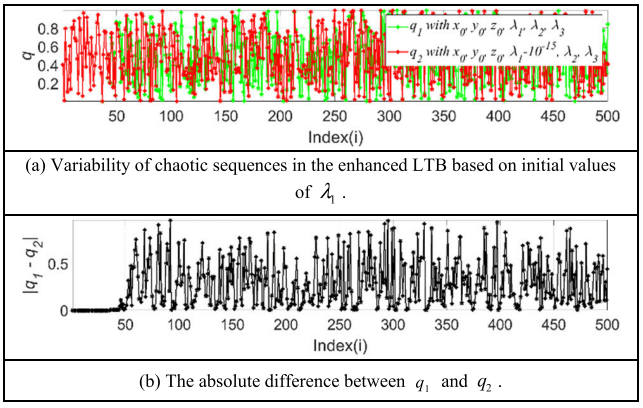
FIGURE 10. Sensitivity analysis of the enhanced LTB to parameter y_0 variation by 10^{-16}.



(a) Variability of chaotic sequences in the enhanced LTB based on initial values of z_0 .

(b) The absolute difference between q_1 and q_2 .

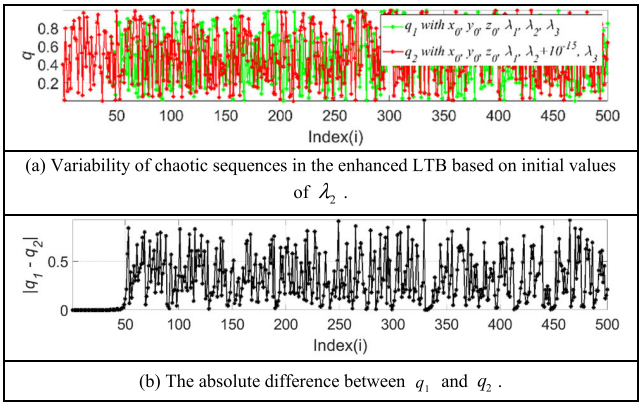
FIGURE 11. Sensitivity analysis of the enhanced LTB to parameter z_0. variation by 10^{-16}.



(a) Variability of chaotic sequences in the enhanced LTB based on initial values of λ_1 .

(b) The absolute difference between q_1 and q_2 .

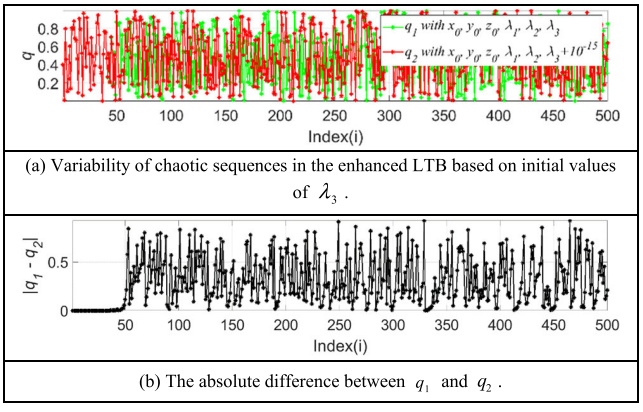
FIGURE 12. Sensitivity analysis of the enhanced LTB to parameter λ_1 variation by 10^{-15}.



(a) Variability of chaotic sequences in the enhanced LTB based on initial values of λ_2 .

(b) The absolute difference between q_1 and q_2 .

FIGURE 13. Sensitivity analysis of the enhanced LTB to parameter λ_2 variation by 10^{-15}.



(a) Variability of chaotic sequences in the enhanced LTB based on initial values of λ_3 .

(b) The absolute difference between q_1 and q_2 .

FIGURE 14. Sensitivity analysis of the enhanced LTB to parameter λ_3 variation by 10^{-15}.

maps: the logistic map, the Tent map, and the Bernoulli map. Parameters (x_0, λ_1) of the LTB have a sensitivity comparable to the corresponding parameters of the other maps. However, the LTB has four additional parameters that exhibit high sensitivity to infinitesimal variations. This property gives the LTB an increased potential for resistance to statistical analysis attacks and brute force attacks, making it more suitable for cryptographic security applications.

B. NUMBER REPRESENTATION AND ITS IMPACT ON THE LTB MAP

The choice of digital format is crucial for implementing the LTB map on an FPGA. Indeed, hardware such as the FPGA recognizes only fields with specific bit widths. Generally, two main formats are used to represent real numbers: fixed-point, which uses a fixed number of bits after the decimal point, and floating-point, which uses a mantissa and an exponent to represent real numbers with

TABLE 3. Sensitivity to the initial conditions of the enhanced LTB map and others chaotic maps.

LTB map		Logistic map		Tent map		Bernoulli shift map	
Parameter	Sensitivity	Parameter	Sensitivity	Parameter	Sensitivity	Parameter	Sensitivity
x_0	10^{-16}	x_0	10^{-16}	y_0	10^{-16}	z_0	10^{-16}
y_0	10^{-16}	λ_1	10^{-15}	λ_2	10^{-15}	λ_3	10^{-15}
z_0	10^{-16}						
λ_1	10^{-15}						
λ_2	10^{-15}						
λ_3	10^{-15}						

variable precision. In this work, we choose to implement the LTB map on an FPGA using fixed-point representation. This choice offers several advantages: firstly, fixed-point representation allows for efficient hardware implementation compared to floating-point representation because it requires fewer hardware resources and reduces the complexity of arithmetic operations. Secondly, the LTB map, as shown in equation (5), involves simple addition, subtraction, and multiplication operations, making it suitable for fixed-point representation. Thirdly, by using fixed-point representation on an FPGA, we can achieve high-performance and low-latency implementation suitable for real-time applications such as cryptography.

TABLE 4. Details of precision architectures used for hardware design of the LTB map.

	Architecture		
	16-bit	32-bit	64-bit
Sign bits	1	1	1
Integer part bits	8	8	8
Fractional part bits	7	23	55

To ensure that our hardware design effectively captures the complex dynamics of the LTB map while adapting to its specified intervals, we used three common architectures (see Table 4): 16, 32, and 64 bits. These architectures were chosen to achieve the sufficient precision required to preserve the chaotic behavior of the LTB map, thus striking a balance between precision and efficiency. Figure 15 presents chaotic sequences generated from the LTB map, logistic map, tent map, and Bernoulli shift map, using fixed-point representation with 16-bit, 32-bit, and 64-bit architectures. Figure 16 shows the histograms of these sequences. From these figures, we observe that the LTB map (as well as its kernel maps), when using fixed-point representation with the 16-bit architecture, generates chaotic sequences with specific values (fixed points) within the map's domain, representing a significant difference compared to chaotic sequences generated with floating-point representation (see Figure 5). However, the chaotic sequences generated with 32-bit and 64-bit architectures are relatively similar to their counterparts generated using floating-point representation. These results demonstrate that 32-bit and 64-bit architectures can be adopted while preserving the dynamic behavior of the chaotic maps. In this work, the 32-bit architecture is chosen for the implementation of LTB on FPGA.

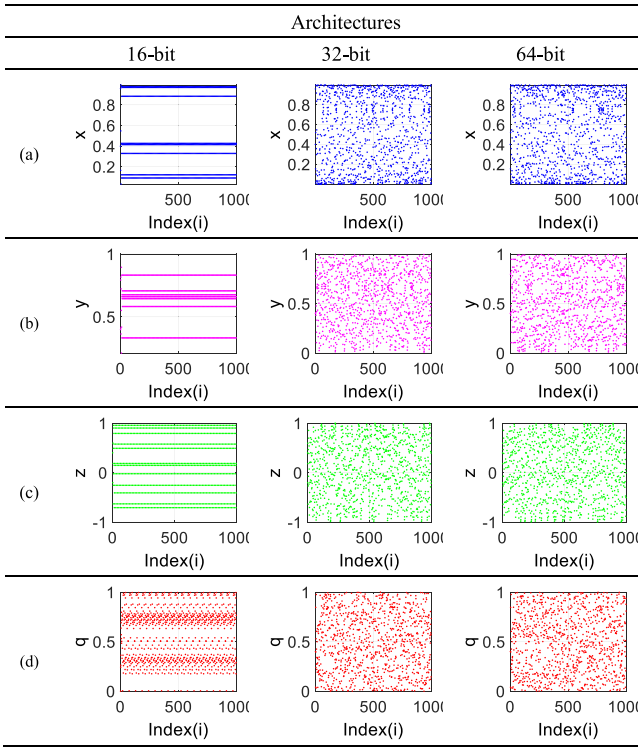


FIGURE 15. Chaotic sequences (of length 1000) generated using fixed-point representation. (a) Logistic map with $(x_0, \lambda_1) = (0.4, 4)$, (b) Tent map with $(y_0, \lambda_2) = (0.4, 1.99)$, (c) Bernoulli shift map $(z_0, \lambda_3) = (0.4, 1.99)$, and (d) LTB with $(x_0, y_0, z_0, \lambda_1, \lambda_2, \lambda_3) = (0.4, 0.4, 0.4, 4, 1.99, 1.99)$.

C. FPGA IMPLEMENTATION OF THE ENHANCED LTB MAP

The LTB map was designed using Quartus II Version 9.0 and implemented on the Altera DE2 FPGA board (Cyclone II EP2C70F896) using Verilog as the hardware description language (HDL). The LTB map algorithm was translated into Verilog-HDL to describe the behavior of the circuit. Each component of the LTB map, including the logistic map, tent map, and Bernoulli shift map, was implemented as separate Verilog modules. Next, the Verilog code of the LTB map was simulated using the ModelSim simulation tool to verify its functionality and detect errors early in the design process. During this step, a comparison was made between the Matlab code and the Verilog code for validation. Subsequently, the Verilog code was synthesized using Quartus II 9.0, Altera's design software. Synthesis mapped our RTL code to the target FPGA architecture, generating a netlist of logical gates.

Figure 17 illustrates the RTL Viewer schematic of the LTB map generated from the RTL code during the synthesis

process of the design. Our hardware design is based on a 32-bit fixed-point architecture (Table 4). The design integrates multipliers (qmult), adders (qadder), comparators (LESS_THAN), multiplexers (mux), and registers (DFFE), allowing multiplication, addition, comparison, selection, and storage of intermediate values for each iteration of the algorithm. The adders and multipliers components are able to perform addition and multiplication using fixed-point arithmetic. Additionally, the Abs component (inst4) computes the absolute value of the output of the Bernoulli shift output, while the bitwise XOR module (inst5) performs the XOR operation on the outputs of the three maps. Clock (clk) and reset (rst) signals were defined to control the timing and initialization of the LTB map circuit.

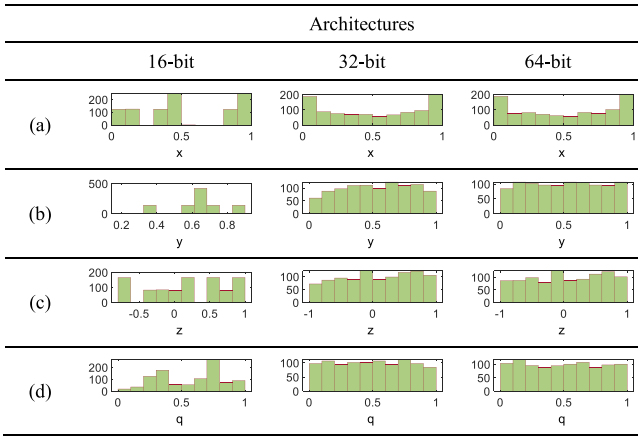


FIGURE 16. Histograms of the chaotic sequences from Figure 15. (a) Logistic map with $(x_0, \lambda_1) = (0.4, 4)$, (b) Tent map with $(y_0, \lambda_2) = (0.4, 1.99)$, (c) Bernoulli shift map $(z_0, \lambda_3) = (0.4, 1.99)$, and (d) LTB with $(x_0, y_0, z_0, \lambda_1, \lambda_2, \lambda_3) = (0.4, 0.4, 0.4, 4, 1.99, 1.99)$.

The resources of the Cyclone II FPGA board utilized by the LTB map are presented in Table 5. This table indicates that the LTB map design effectively utilizes FPGA resources while meeting timing requirements.

Table 6 displays the output values of the LTB map acquired from both the FPGA, the simulation on ModelSim, and the simulation on MATLAB. This table showcases the initial five and final five values of a chaotic sequence with a length of 1000, utilizing identical parameters as outlined in the preceding section. Observation reveals consistent results across all iterations in FPGA, ModelSim, and MATLAB simulations. Hence, it can be deduced that the programming, design, and implementation of the LTB map on the FPGA are adequate.

In the current test, we present the results of comprehensive statistical tests conducted on the sequences generated by the LTB map to validate its chaotic properties and randomness. We employed two widely recognized statistical test suites: the NIST [56] (National Institute of Standards and Technology) Statistical Test Suite (STS) and TestU01 [57], to rigorously evaluate the quality and unpredictability of the LTB map's output.

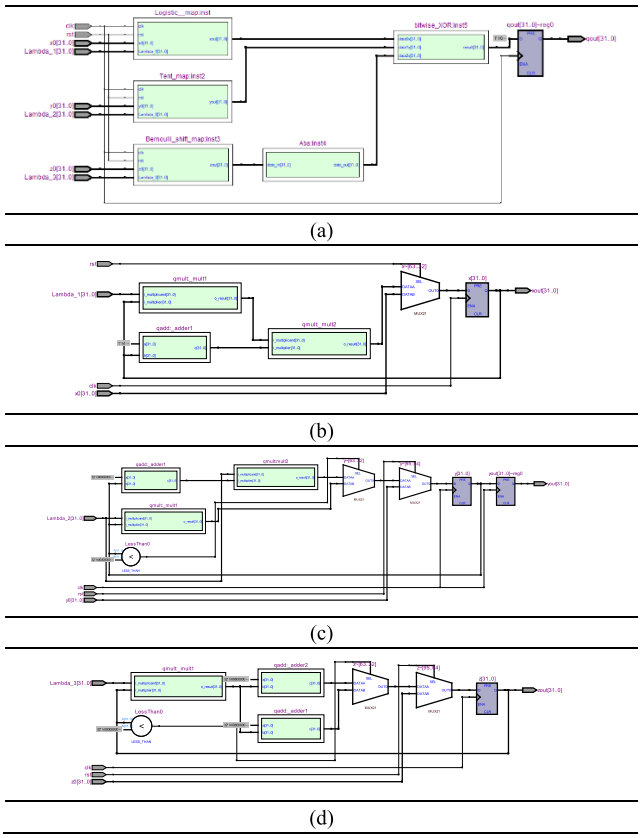


FIGURE 17. RTL Viewer schematic of the enhanced LTB map generated using the Quartus software. (a) LTB map, (b) Logistic map, (c) Tent map, (d) Bernoulli shift map.

TABLE 5. Hardware resources utilized for LTB map implementation on Cyclone II EP2C35F672C6 FPGA.

Resources	Used
Met timing requirements	Yes
Total logic elements	648/33,216(2%)
Total combinational functions	631/33,216(2%)
Dedicated logic registers	126/33,216(< 1%)
Total registers	126
Total memory bits	0/483,840(0%)
Embedded Multiplier 9-bit elements	14/70 (20%)

TABLE 6. Comparison of LTB map output values (in hex) from FPGA board and simulation on ModelSim and Matlab.

Iteration	Matlab – Value	ModelSim–Value	FPGA – Value
1	00333333	00333333	00333333
2	0006a4aa	0006a4aa	0006a4aa
3	006ce673	006ce673	006ce673
4	003ca91b	003ca91b	003ca91b
5	00027344	00027344	00027344
996	00185e0c	00185e0c	00185e0c
997	004376b8	004376b8	004376b8
998	0006803a	0006803a	0006803a
999	0073b780	0073b780	0073b780
1000	006067c1	006067c1	006067c1

The NIST-STS comprises 15 tests designed to evaluate the randomness of binary sequences, necessitating diverse bit

sequences. In this study, 150 binary sequences, each with a length of 10^6 bits, are generated using LTB on FPGA, with parameters configured within the chaotic regime. The average p-values for these sequences and the proportion of passing sequences are presented in Table 7. A sequence is deemed to pass the test if its p-value is greater than or equal to $\alpha = 0.01$, representing the significance level, indicating randomness [2]. The outcomes in Table 7 reveal that a significant majority of sequences generated by the LTB map successfully pass all tests within the NIST suite, affirming their statistical randomness and their viability for cryptographic applications.

To further validate the randomness of the sequences generated by the LTB map, we employed the TestU01 suite, which is renowned for its comprehensive and rigorous testing capabilities. TestU01 includes several batteries of tests, namely SmallCrush (15 tests), Crush (144 tests), and BigCrush (160 tests). We utilized the default settings provided by TestU01 for these batteries without any modifications. The results of these tests are summarized in Table 8. The results presented in Table 8 demonstrate that the LTB map's sequences successfully passed all the tests included in the SmallCrush, Crush, and BigCrush batteries. This rigorous validation confirms the high quality of randomness and unpredictability of the sequences generated by the LTB map.

The results from both the NIST and TestU01 statistical test suites demonstrate that the LTB map generates sequences that exhibit excellent statistical randomness. This validates the chaotic nature of the LTB map and confirms its effectiveness for secure cryptographic applications. These tests provide strong evidence that the LTB map is a robust and reliable choice for high-performance image encryption on FPGA platforms.

TABLE 7. NIST-STS results for LTB map sequences.

Test	Average p-value	Passed (Yes/no)	Proportion of passing sequences
Frequency (Monobit)	0.8494	Yes	0.9867
Block Frequency	0.4259	Yes	0.9933
Runs	0.5130	Yes	0.9933
Long Runs of Ones	0.6447	Yes	1.0000
Rank	0.4804	Yes	0.9867
Spectral DFT	0.6650	Yes	0.9867
Non-overlapping Template Matching	0.2316	Yes	0.9867
Overlapping Template Matching	0.8753	Yes	0.9933
Maurer's "Universal Statistical"	0.8548	Yes	0.9867
Linear complexity	0.2057	Yes	0.9933
Serial 1	0.3793	Yes	0.9933
Serial 2	0.5481	Yes	0.9867
Approximate Entropy	0.2177	Yes	0.9933
Cusum-Forward	0.3283	Yes	0.9933
Cusum-Reverse	0.3938	Yes	0.9867
Random Excursions	0.5268	Yes	0.9800
Random Excursions Variant	0.2348	Yes	0.9933

TABLE 8. TestU01 results for LTB map sequences.

Test battery	Number of tests	Number of passed tests
SmallCrush	15	15
Crush	144	144
BigCrush	160	160

V. PROPOSED LTB-BASED IMAGE ENCRYPTION ALGORITHM AND ITS IMPLEMENTATION ON FPGA

A. DESCRIPTION OF THE ALGORITHM

In this section, we present the proposed image encryption algorithm and the purposes of its main steps. This algorithm is based on LTB (Eq. 5) and shifting and XOR operations (only basic operations). LTB map is used to perform confusion and diffusion operations, which are essential components of many encryption algorithms. Confusion changes the positions of pixels in the image to make the relationship between pixels of plain image and those of encrypted image as complex as possible. On the other hand, diffusion modifies the pixel values of the image to ensure resistance to statistical analysis.

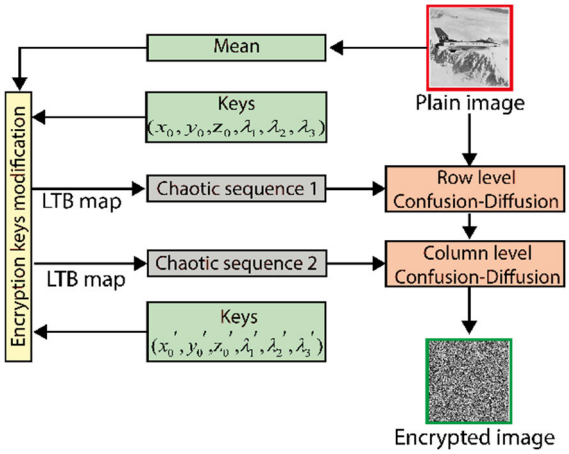


FIGURE 18. The flowchart of the proposed LTB-based encryption algorithm.

Figure 18 illustrates the encryption algorithm through a flowchart. The proposed algorithm initiates by performing a confusion-diffusion process along the rows of the plaintext image. The confusion involves shifting the pixels within each row of the plaintext image in a chaotic manner using a chaotic sequence **q** (generated by LTB map with key parameters $x_0, y_0, z_0, \lambda_1, \lambda_2, \lambda_3$). Simultaneously, each pixel that has been shifted during the confusion undergoes a XOR operation (diffusion) with the corresponding chaotic value in the **q** sequence.

A similar confusion-diffusion process is then applied along the columns, but with a new chaotic sequence **q'** (generated by LTB map with key parameters $x_0', y_0', z_0', \lambda_1', \lambda_2', \lambda_3'$). This enhances the encryption by adding another layer of complexity and security.

To enhance the robustness of our algorithm against potential differential attacks, we use a strategy that involves adjusting the encryption keys based on the statistical characteristics of the plaintext image. Specifically, before utilizing these keys, such as the initial conditions of the LTB map, they undergo modification according to the statistical mean of the input image. By incorporating information derived from the mean of the plaintext image, our algorithm dynamically adapts to the specific features and variations present within the image data. This proactive adjustment

mechanism strengthens the algorithm's resilience against differential attacks, thereby enhancing the security of the encryption process.

The steps below explain more the two processes: encryption and decryption:

Let **I** represent the plaintext image to be encrypted, and **EI** the resulting encrypted image. Additionally, let $(x_0, y_0, z_0, \lambda_1, \lambda_2, \lambda_3; x'_0, y'_0, z'_0, \lambda'_1, \lambda'_2, \lambda'_3)$ denote the parameters of the LTB map serving as the encryption security key.

Encryption Process:

Step 1: Insert the plaintext image **I** of size $m \times n$.

Step 2: Calculate the average value (denoted as A) of the plaintext image pixels.

Step 3: Adjust the encryption keys, particularly the initial conditions of the LTB map, based on the statistical mean of the plaintext image using the following equations:

$$\overline{x_0} = A \times x_0 - floor(A \times x_0), \overline{x'_0} = A \times x'_0 - floor(A \times x'_0) \tag{6}$$

$$\overline{y_0} = A \times y_0 - floor(A \times y_0), \overline{y'_0} = A \times y'_0 - floor(A \times y'_0) \tag{7}$$

$$\overline{z_0} = A \times z_0 - floor(A \times z_0), \overline{z'_0} = A \times z'_0 - floor(A \times z'_0) \tag{8}$$

Step 3: Compute LTB map sequence **q**

Use Eq. (6) with parameters $(\overline{x_0}, \overline{y_0}, \overline{z_0}, \lambda_1, \lambda_2, \lambda_3)$ to generate a chaotic sequence of the LTB map denoted as **q**.

Scale the chaotic sequence **q** to the range $[0, m]$ and round the result to obtain **u**:

$$\mathbf{u} = m \times \mathbf{q}$$
$$\mathbf{u} = floor\ (\mathbf{u}) \tag{9}$$

Scale the chaotic sequence **q** to the range [0, 255] and round the result to obtain **v**:

$$\mathbf{v} = 255 \times \mathbf{q}$$
$$\mathbf{v} = floor\ (\mathbf{v}) \tag{10}$$

The sequences **u** and **v** are used for the confusion and diffusion processes, respectively.

Step 4: Perform confusion-diffusion along the rows:

Initialize an empty matrix **M**.

Shift the pixels of each row of **I** chaotically by values from **u** and perform bitwise XOR operation with corresponding values from **v**. Store the resulting values in **M**.

Step 5: Compute LTB map sequence **q'**:

Similar to step 3, use parameters $(\overline{x'_0}, \overline{y'_0}, \overline{z'_0}, \lambda'_1, \lambda'_2, \lambda'_3)$ to generate LTB map sequence $\mathbf{q}'$ and corresponding scaled sequences, $\mathbf{u}'$ and $\mathbf{v}'$.

Step 6: Perform confusion-diffusion along the columns **:**

Initialize an empty matrix **EI**.

Shift the pixels of each column of **M** chaotically by values from $\mathbf{u}'$ and perform bitwise XOR operation with corresponding values from $\mathbf{v}'$. Store the resulting values in **EI**.

Algorithm 1 LTB-Based Image Encryption Algorithm

Input: Plaintext image **I** of size m × n.
Parameters of LTB map $(x_0, y_0, z_0, \lambda_1, \lambda_2, \lambda_3)$ and $(x'_0, y'_0, z'_0, \lambda'_1, \lambda'_2, \lambda'_3)$
Output: Encrypted Image **EI**.
// Adjust the encryption keys based on the mean of plaintext image
1 Compute A the mean value of **I.**
2 Use Eqs.(6)-(8) to modify the initial conditions $(x_0, y_0, z_0; \overline{x'_0}, \overline{y'_0}, \overline{z'_0})$. The new values are denoted as $(\overline{x_0}, \overline{y_0}, \overline{z_0}; \overline{x'_0}, \overline{y'_0}, \overline{z'_0})$.
// Performing confusion-diffusion along the rows
3 Generate a chaotic sequence of the LTB map denoted as **q**(i), where $1 \leq i \leq m \times n$, using Eq. (6) with the parameters $(\overline{x_0}, \overline{y_0}, \overline{z_0}, \lambda_1, \lambda_2, \lambda_3)$.
4 Use Eq. (9) to scale the chaotic sequence **q** to the range [0,m] and round the result to obtain **u**.
5 Use Eq. (10) to scale the chaotic sequence **q** to the range [0,255] and round the result to obtain **v**.
6 Let **M** be an empty matrix of size m × n
7 Set p = 1;
8 **for** i = 1 to m **do**
9 k = **u**(i);
10 **for** j = 1 to n **do**
11 **if**(k > n) **then**
12 Set k = 1;
13 **end**
14 **M**(k,i) = bitxor(**I**(j,i), **v**(p));
15 Set k = k+1;
16 Set p = p+1;
17 **end**
18 **end**
// Performing confusion-diffusion along the columns.
19 Generate a chaotic sequence of the LTB map denoted as $\mathbf{q}'$(i), where $1 \leq i \leq m \times n$, using Eq.(6) with the parameters $(\overline{x'_0}, \overline{y'_0}, \overline{z'_0}, \lambda'_1, \lambda'_2, \lambda'_3)$.
20 Use Eq.(9) and Eq. (10) to obtain the corresponding scaled sequences $\mathbf{u}'$ and $\mathbf{v}'$ from the chaotic sequence $\mathbf{q}'$, respectively.
21 Let **EI** be an empty matrix of size $m \times n$
22 Set p = 1;
23 **for** i = 1 to m **do**
24 k = $\mathbf{u}'$(i);
25 **for** j = 1 to n **do**
26 **if**(k > n)then
27 Set k = 1;
28 **end**
29 **EI**(i,k) = bitxor(**M**(i,j), $\mathbf{v}'$(p));
30 Set k = k +1;
31 Set p = p +1;
32 **end**
33 **end**
34 **Return EI** the encrypted image.

The detail description of the proposed encryption process is given in Algorithm 1.

Decryption Process: The proposed encryption algorithm described above is symmetric, meaning that similar operations to those used for encryption are applied but in reverse order to recover the original image from the encrypted one.

B. FPGA IMPLEMENTATION OF THE LTB–BASED ENCRYPTION ALGORITHM

The image encryption algorithm is implemented on Intel's Cyclone II FPGA platform, with device number EP2C35F672C6. Algorithm is translated from MATLAB to Verilog-HDL, synthesized, and simulated using Quartus II and ModelSim to verify functionality before deployment on the FPGA. Finally, the verified design is downloaded onto the FPGA, configuring its internal logic to perform the image encryption algorithm in hardware.

Figure 19 shows the RTL Viewer schematic of the proposed encryption algorithm generated during synthesis of the design. This comprehensive schematic comprises several key modules: (1) The "Encryption Key Modification" module, which dynamically adjusts input key parameters based on the average of the plaintext image. This adaptation enhances the robustness of our algorithm against potential differential attacks. (2) Two "LTB" modules (LTB 1 and LTB 2), each generating distinct outputs owing to their different input parameters. (3) Two multipliers are responsible for scaling the chaotic output generated by LTB modules (LTB 1 and LTB 2) to the ranges of [0, m] and [0, 255]. (4) The "Encryption process" module encrypts the data according to the processes of confusion-diffusion along the rows (phase 1) and confusion-diffusion along the columns (phase 2).

The image to be encrypted, sized 256 × 256 pixels, is initially stored in the RAM memory of the FPGA board. The program reads the pixels from RAM, totaling 65,536 pixels. To facilitate this process, a counter (counter 1) is utilized to address the memory sequentially. The counter reads the pixels sequence by sequence in blocks of 1024 pixels (32 × 32 pixels per block) and sends them to the encryption module. In the encryption_process module, the received 1024 pixels are encrypted using chaotic sequences ($\mathbf{u}$ and $\mathbf{v}$) obtained from either LTB1 (for row-level confusion-diffusion) or those ($\mathbf{u}'$ and $\mathbf{v}'$) obtained from LTB2 (for column-level confusion-diffusion), based on a signal called phase_signal. A multiplexer selects between LTB1 and LTB2 for this purpose. Upon initiation of the encryption process, counter 1 halts incrementing (ceases reading from RAM) until the encryption process concludes. The start_encryption signal controls the encryption process. The resulting blocks are then combined to form the encrypted image, which is then transmitted from the FPGA to a PC using the JTAG interface and the URAT protocol. Numerous experiments were conducted utilizing the Altera FPGA development board, with key parameters provided in Table 9.

Table 10 provides information about the resources used by our encryption system within the Cyclone II FPGA device EP2C35F672C6. This table indicates that timing requirements are met, and the utilization of logic elements, registers, memory bits, and embedded multipliers is within specified limits.

TABLE 9. LTB-based encryption system parameters.

Parameter	Value
Keys of row-level encryption $(x_0, y_0, z_0, \lambda_1, \lambda_2, \lambda_3)$	(0.4, 0.4, 0.4, 4, 1.99, 1.99)
Keys of column-level encryption $(x_0', y_0', z_0', \lambda_1', \lambda_2', \lambda_3')$	(0.1, 0.1, 0.1, 4, 1.99, 1.99)
Size of plaintext image	256×256
Mean of Airplane image	179.1850
Mean of Peppers image	120.2212
Mean of House image	159.3376

TABLE 10. Hardware resources utilized for LTB-based image encryption algorithm implementation on Cyclone II EP2C35F672C6 FPGA.

Resources	Used
Met timing requirements	Yes
Total logic elements	11,564/33,216 (35%)
Total combinational functions	7,465/33,216 (22%)
Dedicated logic registers	8,393/33,216 (25%)
Total registers	8393
Total memory bits	8,592/483,840 (2%)
Embedded Multiplier 9-bit elements	27/70 (39%)

VI. EXPERIMENTAL RESULTS

The experiments were carried out using the Cyclone II EP2C35F672C6 FPGA. Various grayscale images sized 256 × 256 were employed as test images for the experiments. Each image was initially loaded into the FPGA board's RAM, where it was divided into blocks of 32 × 32 pixels. These blocks were then transmitted to the cryptosystem sequentially. The encryption system's parameters, outlined in Table 9, are considered local parameters. The proposed system was evaluated across several metrics key space analysis, security assessments, statistical analyses, and computational time.

A. SECRET KEY SPACE ANALYSIS

The key space of an encryption algorithm denotes the collection of unique value combinations viable for encrypting and decrypting data. For robustness against brute-force attacks, this space must encompass at least 2^{100} possibilities [58]. Such attacks entail systematically trying all conceivable keys until the correct one is discovered. Our encryption methodology relies on the chaotic LTB map, which is

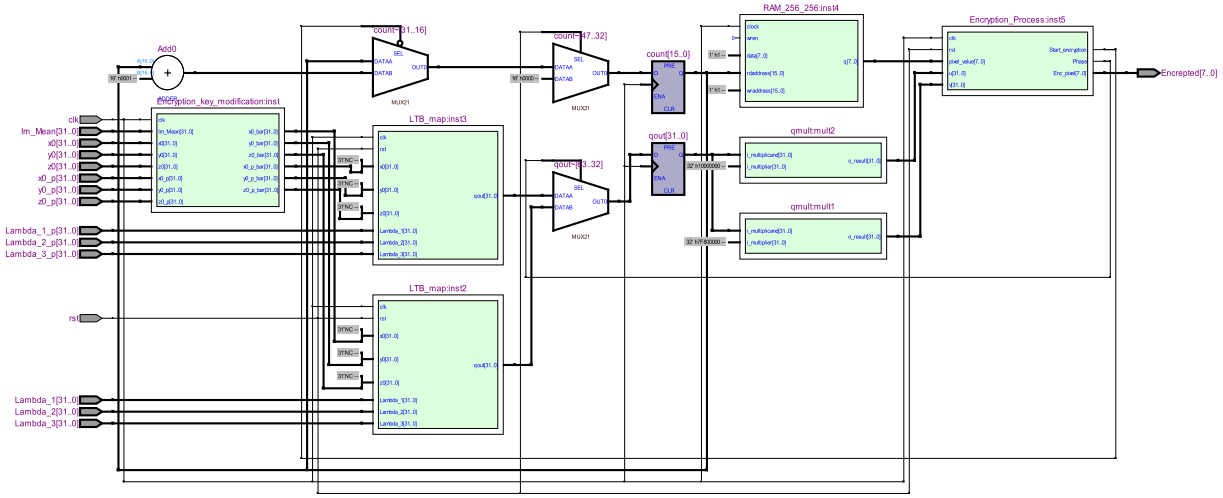


FIGURE 19. RTL Viewer schematic of the proposed encryption algorithm generated using the Quartus software.

responsive to parameters and initial conditions, which are used as private keys. Our algorithm integrates a total of 12 parameters (x_0, y_0, z_0, λ_1, λ_2, λ_3; x'_0, y'_0, z'_0, λ'_1, λ'_2, λ'_3). Employing fixed-point arithmetic with a 32-bit architecture, where 23 bits are allocated for the fractional part, ensures each parameter maintains a precision of 2^{23}. Consequently, the key space of our proposed algorithm equates to $2^{23\times12} = 2^{276}$. This extensive key space fortifies our encryption algorithm against brute-force attacks. Illustratively, Figure 20 depicts the decryption of the Airplane image using a key differing by a single bit from the encryption key. Notably, the decryption process fails, underscoring our algorithm's sensitivity to the encryption key. In contrast, the logistic map, tent map, and Bernoulli shift map, characterized by only two control parameters, offer a lower key space compared to the LTB map and therefore exhibit lower resistance to brute-force attacks. For example, an encryption method based on the Logistic map, Tent map, or Bernoulli shift map (instead of the enhanced LTB map) provides a security key space equal to 2^{92}, which is much lower than the key space obtained from the enhanced LTB map.

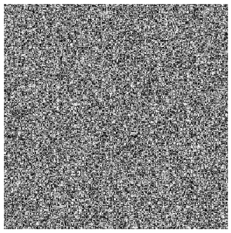


FIGURE 20. Decryption of Airplane image with one-bit difference in encryption key.

According to the comparison results presented in Table 11, our encryption algorithm offers a key space greater than that of [26], [27], [49], but lower than those of [31] and [39]. It should be noted that all methods examined in this comparison have a key space well above 2^{100}.

TABLE 11. Comparison of secret key space.

Method	Proposed	Ref. [26]	Ref. [27]	Ref. [31]	Ref. [49]	Ref. [39]
Key space	2^{276}	2^{186}	2^{224}	2^{340}	2^{159}	2^{356}

B. HISTOGRAM ANALYSIS

The histogram of an image provides insight into the distribution of pixel values, which can be indicative of its visual content. Analyzing the histogram is crucial as it allows for predictions about the image's characteristics. Consequently, an effective encryption algorithm should produce encrypted images with flat histograms to prevent revealing discernible information about the original content. In this context, we evaluate the performance of our encryption algorithm by examining the histograms of the generated encrypted images. Figure 21 illustrates that regardless of the input plaintext image, our algorithm consistently produces encrypted images with flat histograms. This observation signifies that our encryption algorithm successfully conceals the underlying content, thereby enhancing the security and privacy of the data.

C. CORRELATION COEFFICIENT ANALYSIS

The correlation coefficient (CC) of an image serves as a measure of the linear relationship between adjacent pixels in the image. It can be calculated in three directions: vertical, horizontal, and diagonal. An effective encryption algorithm aims to generate encrypted images with reduced correlation compared to the plaintext image, where correlation is typically high. A correlation coefficient close to 0 indicates little or no linear relationship between adjacent pixels in the image. The CC is defined as follows [59]:

$$CC = \frac{\sum (y_i - \bar{y})(x_i - \bar{x})}{\sqrt{\sum (y_i - \bar{y})^2 \sum (x_i - \bar{x})^2}} \tag{11}$$

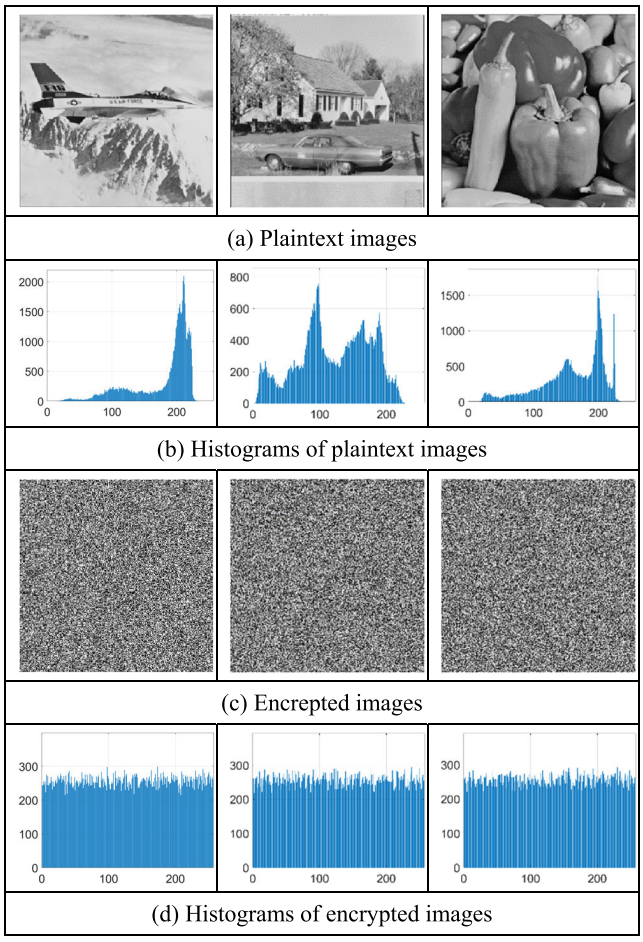


FIGURE 21. Original vs. encrypted images with histograms.

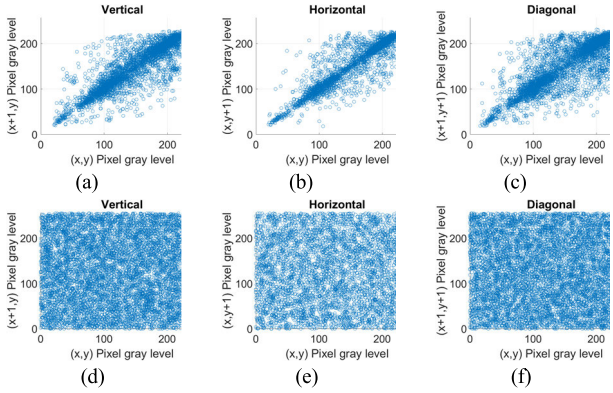


FIGURE 22. Correlation analysis of the airplane image. (a), (b), (c) Distribution of adjacent pixels in the plaintext image. (d), (e), (f) Distribution of adjacent pixels in the encrypted image.

here, y and x represent two sets of adjacent pixels, while $\bar{y}$ and $\bar{x}$ denote their means, respectively.

Figures (22)-(24) depict the distribution of adjacent pixels in the three directions of the encrypted images, showcasing a dispersed pattern. This dispersion indicates that the proposed encryption algorithm effectively reduces the correlation between adjacent pixels. Furthermore, Table 12 provides correlation results obtained from the analysis, confirming

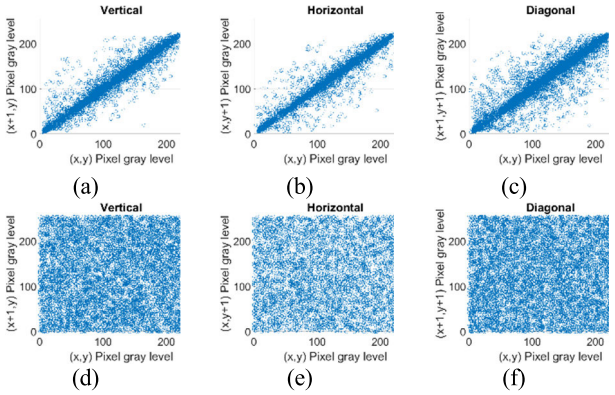


FIGURE 23. Correlation analysis of the peppers image. (a), (b), (c) Distribution of adjacent pixels in the plaintext image. (d), (e), (f) Distribution of adjacent pixels in the encrypted image.

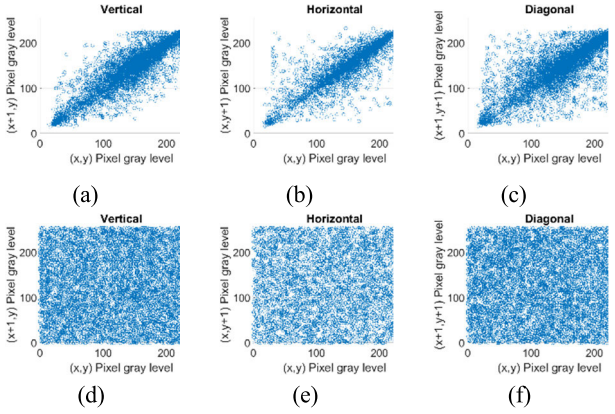


FIGURE 24. Correlation analysis of the house image. (a), (b), (c) Distribution of adjacent pixels in the plaintext image. (d), (e), (f) Distribution of adjacent pixels in the encrypted image.

the low correlation between adjacent pixels in the encrypted images.

TABLE 12. Correlation coefficient test.

Image	Direction	Plaintext image	Encrypted image
Airplane	Vertical	0.9299	0.00130
	Horizontal	0.9317	-0.00172
	Diagonal	0.8746	0.00164
Peppers	Vertical	0.9685	-0.00188
	Horizontal	0.9588	0.00148
	Diagonal	0.9370	0.0021
House	Vertical	0.9110	0.0023
	Horizontal	0.8856	0.00156
	Diagonal	0.8402	0.0022

D. ENTROPY ANALYSIS

Entropy analysis is used to assess the disorder, uncertainty, and unpredictability of data. An effective encryption algorithm should aim to maximize the entropy of encrypted

images, which is equal to 8 for 8-bit grayscale images. A low entropy value suggests more predictable and structured patterns. The entropy is defined as follows [54]:

$$H = -\sum_{i=1}^{L} p(y_i) \times \log_2 (p(y_i)) \tag{12}$$

where L is the total number of pixels and $p(y_i)$ is the probability of occurrence of pixel y_i.

Table 13 presents the entropy results obtained from the analysis, providing quantitative evidence of the effectiveness of the proposed encryption algorithm. The results show that entropy is close to 8 for all test images, indicating the algorithm's success in achieving optimal randomness and unpredictability.

TABLE 13. Entropy test.

	Airplane	Peppers	House
Plaintext image	6.7300	7.5808	7.2389
Encrypted image	7.9979	7.9976	7.9973

E. ROBUSTNESS TO NOISE

An encryption algorithm must exhibit robustness to noise, as encrypted images transmitted over communication channels may be susceptible to noise interference. It is essential for the encryption algorithm to effectively recover plaintext images from encrypted counterparts affected by noise.

Figure 25 illustrates the images recovered from encrypted counterparts affected by various types of noise: Gaussian noise (1% variance), salt-and-pepper noise (1% density), and speckle noise (1% variance). The recovered images depicted in the figure remain recognizable despite the presence of different noise types in the encrypted images. This demonstrates the effectiveness of the encryption algorithm in recovering clear text images despite potential alterations during transmission.

TABLE 14. UACI and NPCR results.

	Airplane	House	Pappers
UACI (%)	33.4628	33.4636	33.4700
NPCR (%)	99.6077	99.6078	99.5988

F. DIFFUSION ANALYSIS TEST

Diffusion analysis is employed to assess the algorithm's robustness against differential attacks. An efficient encryption algorithm should produce two distinct encryption images if only one bit in the input plaintext is modified. To assess resistance against differential attacks, we measured the NPCR (Number of Pixels Change Rate) and the UACI (unified average changed intensity).

Let EI_1 and EI_2 represent two encrypted images of size $m \times n$, generated from two grayscale plaintext images that differ by only one bit. NPCR and UACI are defined as follows [59]:

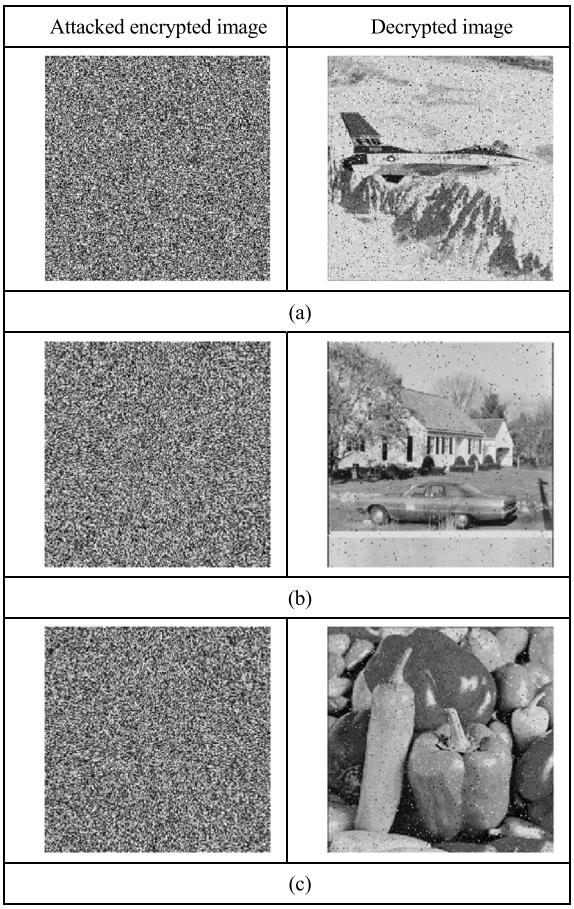


FIGURE 25. Attacked encrypted images and its decrypted versions. (a) Gaussian noise (1%), (b) salt and pepper (1%), and (c) speckle noise (1%).

$$NPCR(\%) = \frac{N_C}{m \times n} \times 100 \tag{13}$$

$$UACI(\%) = \frac{1}{255 \times m \times n} \times \sum_{i=1}^{m} \sum_{j=1}^{n} |EI_1(j,i) - EI_2(j,i)| \times 100 \tag{14}$$

where N_C is the number of pixels that change between EI_1 and EI_2.

In the case of a grayscale image, the ideal expected values for NPCR and UACI are 99.6094% and 33.4635% respectively [60].

Figure 26 depicts the encrypted images generated from two grayscale plaintext images differing by only one bit, along with the difference between them. The results of NPCR and UACI are listed in Table 14. Figure 26 and the results in Table 14 demonstrate that despite the plaintext images used for encryption being highly similar, with only a single-bit difference between them, the resulting encrypted images are significantly different. These findings underscore the efficacy of the proposed algorithm in resisting against differential attacks.

TABLE 15. Performance comparison of encryption algorithms.

Algorithm	Average correlation in absolute value			Average entropy	Average NPCR(%)	Average UACI(%)
	Vertical	Horizontal	Diagonal			
Ideal value	≈0	≈0	≈0	≈8	≈99.6094	≈33.4635
Ref. [61]	0.0046	0.0012	0.0054	7.9892	99.6264	34.5865
Ref. [11]	0.0123	0.0148	0.0102	7.9993	99.6113	33.4407
Ref. [31]	0.0012	0.0016	0.0003	7.9971	99.61	33.55
Ref. [49]	0.0146	0.0081	0.0377	7.9953	99.5975	33.6027
Ref. [39]	0.0020	0.0022	0.0031	7.9972	99.4967	33.4600
Proposed	**0.0018**	**0.0016**	**0.0020**	**7.9976**	**99.6048**	**33.4655**

TABLE 16. Time comparison (in seconds) of encryption algorithms for grayscale image of size 256 × 256.

Method	Implementation environment	Time encryption in s
Ref. [11]	PC equipped with an AMD Ryzen 7 1700 octa-core CPU and 8 GB of memory	0.3949
Ref. [31]	Undefined	0.1450
Ref. [30]	PC equipped with an i7-8550U processor.	0.4914
Ref. [32]	PC with an Intel(R) Core i7-9700 @ 3.00 GHz CPU and 16.0 GB memory	0.0888
Ref. [49]	NanoPC-T3 board equipped with a Cortex-A53 Samsung S5P6818 processor	0.02514
Ref. [39]	Zybo Z20 FPGA board equipped with a ZYNQ processor running at 666.67 MHz	0.0860
Ref. [41]	Cyclone V GX Starter Kit FPGA platform with a clock frequency of 50 MHz	1.415
Proposed	**EP2C70F896C6 Cyclone II FPGA board with a clock frequency of 50 MHz**	**0.00656**

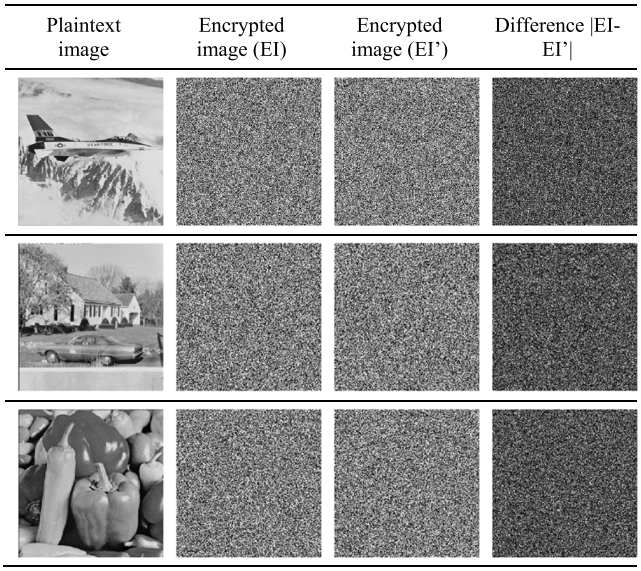


FIGURE 26. Comparison of encrypted images from plaintext images with one-bit difference.

According to the comparison results presented in Table 15, our encryption algorithm generally demonstrates good performance in terms of correlation, entropy, NPCR, and UACI compared to the algorithms selected in this comparison. Only algorithm [31] shows a relatively lower correlation than our method.

G. COMPUTATION TIME

Computation time is a crucial factor for encryption algorithms, especially in practical contexts where efficiency and speed are paramount. A shorter computation time implies that encryption and decryption processes can be executed more quickly, enabling faster data processing. This is particularly important in real-time applications where large volumes of data need to be encrypted and decrypted within strict deadlines, such as in secure communication systems. Therefore, optimizing computation time is often a key consideration in the design and evaluation of encryption algorithms.

Our encryption algorithm is deployed on the EP2C70F896C6 Cyclone II FPGA board, operating at a clock frequency of 50 MHz. The encryption/decryption time is calculated as follows:

$$\text{Time} = \frac{\text{Total number of cycles}}{\text{Clock Frequency}} \tag{15}$$

In our experiments, we used a 256 × 256-sized image for testing, and found that the encryption and decryption procedures required approximately 327984 cycles. Consequently, the encryption and decryption times amounted to 6.5597 milliseconds ($\frac{327984}{50\times10^6}$). This result highlights the effectiveness of our image encryption algorithm, demonstrating its

potential for real-world applications requiring secure and fast data processing.

According to the comparison results presented in Table 16, we observe that our proposed algorithm is significantly faster than several recent algorithms, such as those referenced in [11], [30], [31], [32], [39], [41], [49]. This demonstrates a preference for our work.

VII. CONCLUSION

In this paper, we proposed and implemented a high-speed image encryption algorithm based on the enhanced LTB chaotic map, utilizing FPGA technology. The LTB map has demonstrated improved robustness and unpredictability, characterized by a complex phase diagram and high entropy values compared to its constituent maps operating independently. With six sensitive parameters instead of the two found in its constituent maps, the LTB map offers a considerably extended key space and enhanced security when employed in encryption algorithms. Furthermore, the LTB-based encryption method exhibits an unpredictable nature, with high sensitivity to the encryption key and strong resistance to brute force and differential attacks. Our proposed method also showed notable speed improvements over several recent encryption methods, attributed to the inherent parallelism of the LTB map, coupled with parallelization of encryption operations (confusion and diffusion) and efficient hardware implementation on the FPGA platform. For instance, using the EP2C70F896C6 Cyclone II FPGA board operating at a clock frequency of 50 MHz, our method achieves encryption and decryption times of 6.56 milliseconds for grayscale images sized 256×256 pixels. These results underscore the efficacy and preference for our approach.

AUTHOR CONTRIBUTIONS

Conceptualization, M. Yamni and A. Daoui; methodology, A. A. Abd El-Latif; software, M. Yamni and A. Daoui; validation, P. Pławiak, and O. Alfarraj; formal analysis, M. Yamni and A. A. Abd El-Latif; investigation, A. Daoui and P. Pławiak; resources, O. Alfarraj; data curation, P. Pławiak and O. Alfarraj; writing—original draft preparation, M. Yamni, A. Daoui, and A. A. Abd El-Latif; writing—review and editing, M. Yamni, A. Daoui, and A. A. Abd El-Latif; visualization, M. Yamni and A. Daoui; supervision, A. A. Abd El-Latif; project administration, O. Alfarraj; funding acquisition, P. Pławiak. All authors have read and agreed to the published version of the manuscript.

INFORMED CONSENT STATEMENT

Not applicable

DATA AVAILABILITY STATEMENT

All data will be available upon reasonable request.

CONFLICTS OF INTEREST

The authors declare no conflicts of interest.

REFERENCES

[1] W. Zhou, X. Wang, M. Wang, and D. Li, "A new combination chaotic system and its application in a new bit-level image encryption scheme," *Opt. Lasers Eng.*, vol. 149, Feb. 2022, Art. no. 106782, doi: 10.1016/j.optlaseng.2021.106782.

[2] X. Wang, Y. Feng, and Y. Chen, "A new four-dimensional chaotic system and its circuit implementation," *Frontiers Phys.*, vol. 10, Apr. 2022, Art. no. 906138, doi: 10.3389/fphy.2022.906138.

[3] C. García-Grimaldo, C. F. Bermudez-Marquez, E. Tlelo-Cuautle, and E. Campos-Cantón, "FPGA implementation of a chaotic map with no fixed point," *Electronics*, vol. 12, no. 2, p. 444, Jan. 2023, doi: 10.3390/electronics12020444.

[4] A. A. Abdul-Kareem and W. A. Mahmoud Al-Jawher, "Uruk 4D discrete chaotic map for secure communication applications," *J. Port Sci. Res.*, vol. 5, no. 3, pp. 131–142, Oct. 2022.

[5] M. Hénon, "A two-dimensional mapping with a strange attractor," in *The Theory of Chaotic Attractors*. Berlin, Germany: Springer, 1976, pp. 94–102.

[6] B. Khokhar, S. Dahiya, and K. P. S. Parmar, "Load frequency control of a microgrid employing a 2D sine logistic map based chaotic sine cosine algorithm," *Appl. Soft Comput.*, vol. 109, Sep. 2021, Art. no. 107564, doi: 10.1016/j.asoc.2021.107564.

[7] A. Daoui, M. Yamni, H. Karmouni, M. Sayyouri, H. Qjidaa, S. Motahhir, O. Jamil, W. El-Shafai, A. D. Algarni, N. F. Soliman, and M. H. Aly, "Efficient biomedical signal security algorithm for smart Internet of Medical Things (IoMTs) applications," *Electronics*, vol. 11, no. 23, p. 3867, Nov. 2022.

[8] A. Daoui, M. Yamni, T. Altameem, M. Ahmad, M. Hammad, P. Plawiak, R. Tadeusiewicz, and A. A. Abd El-Latif, "AuCFSR: Authentication and color face self-recovery using novel 2D hyperchaotic system and deep learning models," *Sensors*, vol. 23, no. 21, p. 8957, Nov. 2023.

[9] A. Daoui, M. Yamni, H. Karmouni, M. Sayyouri, H. Qjidaa, M. Ahmad, and A. A. Abd El-Latif, "Biomedical multimedia encryption by fractional-order meixner polynomials map and quaternion fractional-order meixner moments," *IEEE Access*, vol. 10, pp. 102599–102617, 2022.

[10] M. Y. Valandar, M. J. Barani, P. Ayubi, and M. Aghazadeh, "An integer wavelet transform image steganography method based on 3D sine chaotic map," *Multimedia Tools Appl.*, vol. 78, no. 8, pp. 9971–9989, Apr. 2019, doi: 10.1007/s11042-018-6584-2.

[11] G. Ye, H. Wu, K. Jiao, and D. Mei, "Asymmetric image encryption scheme based on the quantum logistic map and cyclic modulo diffusion," *Math. Biosciences Eng.*, vol. 18, no. 5, pp. 5427–5448, 2021.

[12] Y. Hu, H. Wu, and L. Zhou, "Color image encryption base on a 2D hyperchaotic enhanced Henon map and cross diffusion," *Alexandria Eng. J.*, vol. 73, pp. 385–402, Jul. 2023, doi: 10.1016/j.aej.2023.04.060.

[13] S. Zhou, Y. Qiu, X. Wang, and Y. Zhang, "Novel image cryptosystem based on new 2D hyperchaotic map and dynamical chaotic S-box," *Nonlinear Dyn.*, vol. 111, no. 10, pp. 9571–9589, May 2023, doi: 10.1007/s11071-023-08312-1.

[14] A. Daoui, M. Yamni, H. Karmouni, M. Sayyouri, H. Qjidaa, M. Ahmad, and A. A. A. El-Latif, "Color stereo image encryption and local zero-watermarking schemes using octonion Hahn moments and modified Henon map," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 10, pp. 8927–8954, Nov. 2022.

[15] Z. Hua, F. Jin, B. Xu, and H. Huang, "2D logistic-sine-coupling map for image encryption," *Signal Process.*, vol. 149, pp. 148–161, Aug. 2018.

[16] B. Sinha, S. Kumar, and C. Pradhan, "Comparative analysis of color image encryption using 3D chaotic maps," in *Proc. Int. Conf. Commun. Signal Process. (ICCSP)*, Apr. 2016, pp. 332–335. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7754150/

[17] Z. Hua, Y. Zhou, C.-M. Pun, and C. L. P. Chen, "2D sine logistic modulation map for image encryption," *Inf. Sci.*, vol. 297, pp. 80–94, Mar. 2015.

[18] C. Cao, K. Sun, and W. Liu, "A novel bit-level image encryption algorithm based on 2D-LICM hyperchaotic map," *Signal Process.*, vol. 143, pp. 122–133, Feb. 2018.

[19] M. A. Murillo-Escobar, C. Cruz-Hernández, F. Abundiz-Pérez, and R. M. López-Gutiérrez, "Implementation of an improved chaotic encryption algorithm for real-time embedded systems by using a 32-bit microcontroller," *Microprocessors Microsystems*, vol. 45, pp. 297–309, Sep. 2016.

[20] J. Arif, M. A. Khan, B. Ghaleb, J. Ahmad, A. Munir, U. Rashid, and A. Y. Al-Dubai, "A novel chaotic permutation-substitution image encryption scheme based on logistic map and random substitution," *IEEE Access*, vol. 10, pp. 12966–12982, 2022.

[21] K. M. Hosny and M. M. Darwish, "Reversible color image watermarking using fractional-order polar harmonic transforms and a chaotic sine map," *Circuits, Syst., Signal Process.*, vol. 40, no. 12, pp. 6121–6145, Dec. 2021.

[22] R. Qi, S. Ji, J. Shen, P. Vijayakumar, and N. Kumar, "Security preservation in industrial medical CPS using Chebyshev map: An AI approach," *Future Gener. Comput. Syst.*, vol. 122, pp. 52–62, Sep. 2021.

[23] S. Y. Dezfuli Nezhad, N. Safdarian, and S. A. H. Zadeh, "New method for fingerprint images encryption using DNA sequence and chaotic tent map," *Optik*, vol. 224, Dec. 2020, Art. no. 165661.

[24] J. C. Dagadu, J.-P. Li, and E. O. Aboagye, "Medical image encryption based on hybrid chaotic DNA diffusion," *Wireless Pers. Commun.*, vol. 108, no. 1, pp. 591–612, Sep. 2019.

[25] C. Han, "An image encryption algorithm based on modified logistic chaotic map," *Optik*, vol. 181, pp. 779–785, Mar. 2019.

[26] J. Wang, L. Liu, M. Xu, and X. Li, "A novel content-selected image encryption algorithm based on the LS chaotic model," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 10, pp. 8245–8259, Nov. 2022.

[27] M. Z. Talhaoui and X. Wang, "A new fractional one dimensional chaotic map and its application in high-speed image encryption," *Inf. Sci.*, vol. 550, pp. 13–26, Mar. 2021.

[28] A. Belazi, S. Kharbech, M. N. Aslam, M. Talha, W. Xiang, A. M. Iliyasu, and A. A. A. El-Latif, "Improved sine-tangent chaotic map with application in medical images encryption," *J. Inf. Secur. Appl.*, vol. 66, May 2022, Art. no. 103131.

[29] M. Z. Talhaoui, X. Wang, and A. Talhaoui, "A new one-dimensional chaotic map and its application in a novel permutation-less image encryption scheme," *Vis. Comput.*, vol. 37, no. 7, pp. 1757–1768, Jul. 2021, doi: 10.1007/s00371-020-01936-z.

[30] L. Zhu, D. Jiang, J. Ni, X. Wang, X. Rong, and M. Ahmad, "A visually secure image encryption scheme using adaptive-thresholding sparsification compression sensing model and newly-designed memristive chaotic map," *Inf. Sci.*, vol. 607, pp. 1001–1022, Aug. 2022.

[31] L. Liu and J. Wang, "A cluster of 1D quadratic chaotic map and its applications in image encryption," *Math. Comput. Simul.*, vol. 204, pp. 89–114, Feb. 2023.

[32] S. Zhu, X. Deng, W. Zhang, and C. Zhu, "Secure image encryption scheme based on a new robust chaotic map and strong S-box," *Math. Comput. Simul.*, vol. 207, pp. 322–346, May 2023.

[33] A. Daoui, M. Yamni, S. A. Chelloug, M. A. Wani, and A. A. A. El-Latif, "Efficient image encryption scheme using novel 1D multiparametric dynamical tent map and parallel computing," *Mathematics*, vol. 11, no. 7, p. 1589, Mar. 2023, doi: 10.3390/math11071589.

[34] A. Daoui, H. Mao, M. Yamni, Q. Li, O. Alfarraj, and A. A. Abd El-Latif, "Novel integer shmaliy transform and new multiparametric piecewise linear chaotic map for joint lossless compression and encryption of medical images in IoMTs," *Mathematics*, vol. 11, no. 16, p. 3619, Aug. 2023, doi: 10.3390/math11163619.

[35] M. Yamni, A. Daoui, and A. A. Abd El-Latif, "Efficient color image steganography based on new adapted chaotic dynamical system with discrete orthogonal moment transforms," *Math. Comput. Simul.*, Feb. 2024, doi: 10.1016/j.matcom.2024.01.023.

[36] E. Avaroğlu, T. Tuncer, A. B. Özer, B. Ergen, and M. Türk, "A novel chaos-based post-processing for TRNG," *Nonlinear Dyn.*, vol. 81, nos. 1–2, pp. 189–199, Jul. 2015, doi: 10.1007/s11071-015-1981-9.

[37] M. P. Leong, S. Z. M. Naziri, and S. Y. Perng, "Image encryption design using FPGA," in *Proc. Int. Conf. Electr., Electron. Syst. Eng. (ICEESE)*, Dec. 2013, pp. 27–32. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6895037/

[38] O. A. Aboulseoud and S. M. Ismail, "FPGA floating point fractional-order chaotic map image encryption," in *Proc. 31st Int. Conf. Microelectron. (ICM)*, Dec. 2019, pp. 134–137. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9021500/

[39] M. Maazouz, A. Toubal, B. Bengherbia, O. Houhou, and N. Batel, "FPGA implementation of a chaos-based image encryption algorithm," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 10, pp. 9926–9941, Nov. 2022, doi: 10.1016/j.jksuci.2021.12.022.

[40] Y. Luo, C. Fan, C. Xu, and X. Li, "Design and FPGA implementation of a high-speed PRNG based on an n-D non-degenerate chaotic system," *Chaos, Solitons Fractals*, vol. 183, Jun. 2024, Art. no. 114951.

[41] H. A. Abdullah and H. N. Abdullah, "FPGA implementation of color image encryption using a new chaotic map," *Indonesian J. Electr. Eng. Comput. Sci.*, vol. 13, no. 1, pp. 129–137, Jan. 2019.

[42] B. Bouteghrine, C. Tanougast, and S. Sadoudi, "Design and FPGA implementation of new multidimensional chaotic map for secure communication," *J. Circuits, Syst. Comput.*, vol. 30, no. 15, Dec. 2021, Art. no. 2150280, doi: 10.1142/s0218126621502807.

[43] Q. Wang, S. Yu, C. Li, J. Lü, X. Fang, C. Guyeux, and J. M. Bahi, "Theoretical design and FPGA-based implementation of higher-dimensional digital chaotic systems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 63, no. 3, pp. 401–412, Mar. 2016.

[44] E. Tlelo-Cuautle, L. G. de la Fraga, V.-T. Pham, C. Volos, S. Jafari, and A. D. J. Quintas-Valles, "Dynamics, FPGA realization and application of a chaotic system with an infinite number of equilibrium points," *Nonlinear Dyn.*, vol. 89, no. 2, pp. 1129–1139, Jul. 2017, doi: 10.1007/s11071-017-3505-2.

[45] H. A. Abdullah and H. N. Abdullah, "FPAA implementation of chaotic modulation based on Nahrain map," *Iraqi J. Inf. Commun. Technol.*, vol. 1, no. 3, pp. 17–30, Feb. 2019.

[46] X. Zhou, C. Li, X. Lu, T. Lei, and Y. Zhao, "A 2D hyperchaotic map: Amplitude control, coexisting symmetrical attractors and circuit implementation," *Symmetry*, vol. 13, no. 6, p. 1047, Jun. 2021, doi: 10.3390/sym13061047.

[47] N. Charalampidis, A. Iatropoulos, and C. Volos, "Chaos based speech encryption using microcontroller," *Integration*, vol. 95, Mar. 2024, Art. no. 102128.

[48] M. Yamni, A. Daoui, H. Karmouni, M. Sayyouri, H. Qjidaa, S. Motahhir, O. Jamil, W. El-Shafai, A. D. Algarni, N. F. Soliman, and M. H. Aly, "An efficient watermarking algorithm for digital audio data in security applications," *Sci. Rep.*, vol. 13, no. 1, p. 18432, Oct. 2023.

[49] Z. Lin, J. Liu, J. Lian, Y. Ma, and X. Zhang, "A novel fast image encryption algorithm for embedded systems," *Multimedia Tools Appl.*, vol. 78, no. 14, pp. 20511–20531, Jul. 2019, doi: 10.1007/s11042-018-6824-5.

[50] M. I. AlAli, K. M. Mhaidat, and I. A. Aljarrah, "Implementing image processing algorithms in FPGA hardware," in *Proc. IEEE Jordan Conf. Appl. Electr. Eng. Comput. Technol. (AEECT)*, Dec. 2013, pp. 1–5, doi: 10.1109/AEECT.2013.6716446.

[51] A. Magyari and Y. Chen, "Review of state-of-the-art FPGA applications in IoT networks," *Sensors*, vol. 22, no. 19, p. 7496, Oct. 2022, doi: 10.3390/s22197496.

[52] M. Elnawawy, A. Sagahyroon, and T. Shanableh, "FPGA-based network traffic classification using machine learning," *IEEE Access*, vol. 8, pp. 175637–175650, 2020.

[53] C. Li, G. Luo, K. Qin, and C. Li, "An image encryption scheme based on chaotic tent map," *Nonlinear Dyn.*, vol. 87, no. 1, pp. 127–133, Jan. 2017.

[54] C. Pak, J. Kim, K. An, C. Kim, K. Kim, and C. Pak, "A novel color image LSB steganography using improved 1D chaotic map," *Multimedia Tools Appl.*, vol. 79, nos. 1–2, pp. 1409–1425, Jan. 2020.

[55] C. Pak, K. An, P. Jang, J. Kim, and S. Kim, "A novel bit-level color image encryption using improved 1D chaotic map," *Multimedia Tools Appl.*, vol. 78, no. 9, pp. 12027–12042, May 2019, doi: 10.1007/s11042-018-6739-1.

[56] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, and J. Dray, *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*, vol. 22. Gaithersburg, MD, USA: U.S. Department of Commerce, Technology Administration, National Institute of Standards and Technology, 2001.

[57] P. L'Ecuyer and R. Simard, "TestU01: AC library for empirical testing of random number generators," *ACM Trans. Math. Softw.*, vol. 33, no. 4, pp. 1–40, Aug. 2007, doi: 10.1145/1268776.1268777.

[58] G. Alvarez and S. Li, "Some basic cryptographic requirements for chaos-based cryptosystems," *Int. J. Bifurcation Chaos*, vol. 16, no. 8, pp. 2129–2151, Aug. 2006.

[59] S. El Assad and M. Farajallah, "A new chaos-based image encryption system," *Signal Process., Image Commun.*, vol. 41, pp. 144–157, Feb. 2016.

[60] X. Wang, N. Guan, H. Zhao, S. Wang, and Y. Zhang, "A new image encryption scheme based on coupling map lattices with mixed multi-chaos," *Sci. Rep.*, vol. 10, no. 1, p. 9784, Jun. 2020.

[61] G. Kaur, R. Agarwal, and V. Patidar, "Chaos based multiple order optical transform for 2D image encryption," *Eng. Sci. Technol., Int. J.*, vol. 23, no. 5, pp. 998–1014, Oct. 2020, doi: 10.1016/j.jestch.2020.02.007.

MOHAMED YAMNI was born in Fes, Morocco, in 1993. He received the B.Eng. degree in electrical engineering and the M.S. degree in engineering science from FSDM, University of Sidi Mohammed Ben Abdellah, Fes, and the Ph.D. degree in electrical engineering from the University of Sidi Mohammed Ben Abdellah, in 2022. He has published approximately 40 papers published in journals and conference proceedings, comprising journal articles, book chapters, and conference papers, with over 600 citations. His research interests include signal processing, image processing, pattern recognition, multimedia data security on embedded systems, and hardware implementation. Beyond his research contributions, he actively serves the academic community as a reviewer for several prestigious journals with high impact factors.

ACHRAF DAOUI was born in Taounate, Morocco. He received the B.Eng. degree in electrical engineering and the M.S. degree in engineering science from the Faculty of Science, University of Sidi Mohammed Ben Abdellah, Fes, Morocco, in 2013 and 2018, respectively, and the Ph.D. degree in electrical engineering from the Laboratory of Engineering, Systems, and Applications, National School of Applied Sciences, University of Sidi Mohamed Ben Abdellah, in 2022. He has made significant contributions to the field, with over 48 publications, including book chapters, journal articles, and conference papers. His research interests include signal processing, image processing, pattern recognition, and robotic control. He serves as a Reviewer for several high-impact factor journals, such as *Pattern Recognition*, *Expert Systems with Applications*, *Artificial Intelligence Review*, and *Journal of Ambient Intelligence and Humanized Computing*.

PAWEŁ PŁAWIAK was born in Ostrowiec, Poland, in 1984. He received the B.Eng. and M.Sc. degrees in electronics and telecommunications and the Ph.D. degree (Hons.) in biocybernetics and biomedical engineering from the AGH University of Science and Technology, Kraków, Poland, in 2012 and 2016, respectively, and the D.Sc. degree in technical computer science and telecommunications from the Silesian University of Technology, Gliwice, Poland, in 2020. He is currently the Dean of the Faculty of Computer Science and Telecommunications; an Associate Professor with the Cracow University of Technology, Kraków; the Deputy Director for Research with the National Institute of Telecommunications, Warsaw; and an Associate Professor with the Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, Gliwice. He has published more than 50 articles in refereed international SCI-IF journals. His research interests include machine learning and computational intelligence (e.g., artificial neural networks, genetic algorithms, fuzzy systems, support vector machines, k-nearest neighbors, and hybrid systems), ensemble learning, deep learning, evolutionary computation, classification, pattern recognition, signal processing and analysis, data analysis and data mining, sensor techniques, medicine, biocybernetics, biomedical engineering, and telecommunications. He is an academic editor and a reviewer of many prestigious and reputed journals.

OSAMA ALFARRAJ received the master's and Ph.D. degrees in information and communication technology from Griffith University, in 2008 and 2013, respectively. He is currently a Professor of computer sciences with King Saudi University, Riyadh, Saudi Arabia. He was a Consultant and a member of the Saudi National Team for Measuring E-Government, Saudi Arabia, for two years. His current research interests include eSystems (eGov, eHealth, and ecommerce), cloud computing, and big data.

AHMED A. ABD EL-LATIF (Senior Member, IEEE) received the Ph.D. degree from Harbin Institute of Technology, China, in 2013. Since then, he has led and participated in several successful research projects and secured grants in Egypt, the Russian Federation, Saudi Arabia, China, Malaysia, and Tunisia. Currently, he holds staff positions with Menoufia University, Egypt, and Prince Sultan University, Saudi Arabia. Since 2022, he has been the Head of the MEGANET 6G Laboratory Research in the Russian Federation. He holds several leadership positions, including the Vice Chair of the EIAS Research Laboratory; and a Founder and the Deputy Director of the Center of Excellence in Quantum and Intelligent Computing, Prince Sultan University, Saudi Arabia. With over 18 years of professional experience, he has published approximately 300 papers in journals and conference proceedings, including 12 books, with over 10,000 citations. His research interests include quantum communications and cryptography, cybersecurity, artificial intelligence of things (AIoT), AI-based image processing, information hiding, and applications of dynamical systems (discrete-time models: chaotic systems and quantum walks) in cybersecurity. He has received several awards, including the State Encouragement Award in Engineering Sciences from the Arab Republic of Egypt, in 2016; the Best Ph.D. Student Award from the Harbin Institute of Technology, China, in 2013; and the Young Scientist Award from Menoufia University, in 2014. He actively participates in the scientific community, serving as the Chair/Co-Chair for several Scopus/EI conferences. He also holds editorial positions, including the Editor-in-Chief of the *International Journal of Information Security and Privacy*, a Series Editor of *Quantum Information Processing and Computing* and *Advances in Cybersecurity Management*. Additionally, he serves as an academic editor or an associate editor for many indexed journals in the Web of Science (WoS) and Scopus, covering various quartiles.