A Novel Low-Power Encryption Scheme Based on Chaotic Dynamic Triple Pendulum System for Wide Range of Applications

Bikram Paul, Souradip Pal, Abhishek Agrawal, and Gaurav Trivedi

Abstract—

Recent advancements in the domain of quantum computing are posing a security threat to the classical cryptography algorithms. Popular symmetric and asymmetric cryptosystems including RSA, ECC, DES, Diffie-Hellman etc. can be broken by a quantum computer executing Shors and Grovers algorithms. This motivated scientific community to design newer encryption schemes to address security vulnerabilities. Hash, Code, Lattice, Multivariate Polynomial based cryptography algorithms, known as post-quantum cryptography algorithms (PQC), exhibit resistance against classical as well as quantum crypto-attacks. Apart from these PQC algorithms, a relatively new method of constructing cryptosystems utilizing the unpredictability property of discrete chaotic dynamic systems has become noteworthy from the practical perspective. In this paper, we present a novel approach to design an encryption scheme based on the chaotic dynamic physical system, which is derived from a mechanical model depicting nonlinear dynamics and exhibits resistance against various attacks. The effectiveness of the proposed cryptography scheme is validated against various standard tests, such as Lyapunov exponents test, bifurcation diagrams, sensitivity to parametric and to initial values, ergodicity, collision test, NIST, diehard randomness test etc. This algorithm is also verified through an FPGA implementation to assess its usage in low power high throughput applications as well. The power consumption and resource utilization of the proposed design are 56% and 72.6%, respectively, as compared to other known methods while operating at 628.14 MHz. It is observed that the proposed design can work efficiently with various wide range of applications. It is observed that the proposed design can work efficiently with various wide range of applications. The average power and area of its ASIC implementation at 180 nm technology are 61.8836 mW and 0.20374 mm² at 250 MHz, respectively.

Index Terms—Cryptography, Chaos, Non-linear dynamic systems, FPGA, Discrete system, Differential state-space model.

1 INTRODUCTION

THE enhancement of computing power and the rise of quantum computing have made many widely adopted conventional encryption schemes vulnerable. Due to this scientific community is compelled to find a suitable solution to address this issue. Chaotic dynamic system based algorithms categorized as PQC algorithms can act as viable solution to this threat. There is no known successful quantum computer based attacks against chaos based encryption schemes [1]. The main advantages of deterministic chaotic systems are higher degree of unpredictability, ergodicity, high sensitivity to initial conditions and control parameters, close resemblance with true randomness and unstable long aperiodic orbits. These chaotic properties can be employed in synchronization and control, communication, cryptography, neural network, etc. [2], [3], [4]. These systems are distinguished by their high sensitivity to initial conditions, statistical similarity to random signals and a continuous

broadband power spectrum. All this has garnered interest from cryptanalysts to propose various chaos based cryptographic systems [5], [6]. The chaos based cryptosystems can be subdivided into two classes. In the first class, a chaotic system is numerically iterated over a period by employing a message as an initial data [7], whereas, in the second class, a message is scrambled with a chaotic dynamic function. The relevance and usefulness of chaos have been demonstrated through comparative studies employing characteristics of a chaotic system and the requirement of a strong cipher [6], [7]. Cryptographically similar domains of chaotic systems can be utilized to generate a strong cipher for a strong cryptosystem.

Designing cryptographic schemes based on chaotic dynamics can be divided into two steps. In the first step, an encryption algorithm is modeled based on analog chaotic dynamic systems. Thereafter, discretization with finite-precision is introduced during hardware implementation. It is to mention that discretization is the main trade-off between security and resource utilization. As we know, the method of synchronization of multiple chaotic systems is applied in some of the earliest cryptography applications. Among the chaotic maps, the *piece-wise linear chaotic map* (PLCM) [8], [9], [10] is the simplest one-dimensional system and can be implemented easily, but it is prone to security vulnerabilities due to smaller keyspace. Various modifications and variants of 1-D PLCM, such as piece-wise

- Bikram Paul is PhD research scholar with the Department of Electronics and Electrical Engineering, Indian Institute of Technology, Guwahati, Assam, 781039 India. E-mail: bikram@iitg.ac.in.
- Souradip Pal and Abhishek Agarwal are bachelor's students with the Department of Electronics and Electrical Engineering, Indian Institute of Technology, Guwahati, Assam, 781039 India, E-mails: souradip.iitg@gmail.com, abhi.agrawal1412@gmail.com.
- Gaurav Trivedi is Professor at Indian Institute of Technology, Guwahati, E-mail: trivedi@iitg.ac.in.

Manuscript received June –, 2021; revised — –, —.

nonlinear chaotic maps, algorithms based on logistic maps [11], [12], polynomial chaotic maps are reported in literature. Now-a-days, many real world chaotic phenomenon are being modeled to get ideal chaotic characteristics; therefore, physical chaotic dynamical systems have become an imperative area of research in chaos based cryptography [13] [14].

For strengthening the security and to increase the speed of chaotic cryptosystems, a combination of multiple one dimensional algorithms are used. For example, in [15], a combination of polynomial chaotic map and the piecewise nonlinear chaotic map is employed to enhance the speed and security of a cryptosystem. It is to mention that various chaotic block cipher encryption schemes are also developed. A chaotic tent map based single iterative encryption scheme is proposed in [16], which is further generalized in [17], and multiple iterative based dynamical chaotic system models are presented in [18]. A novel symmetric block cipher algorithm based on invertible two dimensional chaotic maps on a torus is presented in [19], whereas, schemes formulated by employing iterative logistic equation and synchronized chaotic systems are discussed in [20] and [21], respectively. In another approach, different chaotic attractors for a random chunk of binary plain-text data is presented [22] in order to implement chaotic based encryption scheme.

As per our knowledge, only a few chaotic map-based image and video encryption schemes have been developed in the past. Some of these methods are mentioned as follows. In [13], a new hyperchaotic temperature fluctuation model based on the circuit analysis is proposed for an image encryption. Simple first order differential equations ($\dot{x} = z$; $\dot{y} = -z(ay + by^2 + xz)$; $\dot{z} = x^2 + y^2 - |x| - 1$) based a new 3D chaotic system employing a peanut-shaped symmetric equilibrium curve is reported in [23]. A non-linear continuous-time dynamical chaotic system based on two circles of equilibrium points and its multistability is discussed in [24]. As mentioned above, all these chaotic systems are utilized for image and video encryption only and there is no general-purpose chaotic dynamic system based cryptosystem exist. With the advent of smart systems incorporating Internet-of-Things (IoT) devices, data (or information) security has become of utmost importance. Internet of Everything (IoE) is the future and it demands a system to have very high security, energy efficiency and reliability. Therefore, it has become imperative to design a general purpose low power cryptosystem, which can be used in various applications ranging from IoT based smart systems to high throughput low latency systems. In this paper, we proposed a novel and generic chaos-based low power cryptosystem to address above mentioned issues, which is based on a chaotic triple pendulum system. This novel algorithm is also implemented on FPGA as well as on ASIC and is compared with other state of the art methods reported in the literature. Details of our proposed algorithm is presented in the subsequent sections.

The organization of the manuscript is as follows. Section II illustrates a step-wise approach and methodology to design a triple pendulum based cryptography algorithm. In Section III, formulations of chaotic double and triple pendulum dynamic systems are presented with case studies. Section IV presents Lyapunov exponents test and bifurcation diagrams to test chaotic behavior of the proposed triple

pendulum based chaotic dynamic system. Later, encryption and decryption algorithms are also depicted in the same section. The details of hardware implementation are exhibited in section V. Thereafter, detailed discussion on the results obtained on FPGA and ASIC to validate effectiveness of the proposed method, and various tests, such as sensitivity to parametric and to initial values, ergodicity, collision test, NIST, diehard randomness test conducted to assure chaotic nature of the proposed method are described in section VI. Finally, a conclusion is drawn in Section VII.

2 PROPOSED APPROACH

The proposed research focuses on implementing a new chaos-based encryption-decryption technique on Field Programmable Gate Array (FPGA) and as an Application Specific Integrated Circuit (ASIC). Unlike traditional cryptographic methods, we aim to design a system in such a way that it should not only be less resource-intensive and computationally efficient but also provide a reasonable degree of security. For achieving this goal, necessary steps to be performed along with design and implementation are given below.

- *Verification of Encryption-Decryption Algorithm:* Validity of chaotic nature and quantum-safe properties of the proposed scheme
- *Hardware-level optimization of the Algorithm:* Efficient pipelining of the proposed algorithm for fast response in practical scenarios
- *Key Generation & Management:* Storage and maintenance of valid keys, and efficient key exchange strategies
- *Security Analysis of the Cryptosystem:* Analysis of randomness and complexity of the proposed system

Figure 1 depicts design and implementation steps of the proposed cryptosystem.

Fig. 1: Design steps of the proposed method

The motivation behind employing the proposed algorithms for encryption and decryption is based on Baptista-type encryption-decryption scheme. It is described using an interval-partitioning of chaotic orbits of a 1D logistic map, which enables construction of very fast and secure

encryption-decryption schemes, because of its less complex structure. The main objective of our scheme is to first map text characters to numerical values and, then algorithms are applied to encrypt the message. Decryption is performed by iterating a chaotic map and, then corresponding symbols for numerical values are obtained by inverting the process. After referring literature, we have decided to employ a compound triple-pendulum model for creating desired chaotic map in our cryptosystem. Since, input to the system is a plain message, the system parameter(s) and additionally the initial conditions of a dynamical system are assumed to be the part of the secret key. After analyzing a triple-pendulum system, a valid set of keys for encryption and decryption is generated. The entire mechanical model is implemented on an FPGA along with a custom ALU and a control unit adhering to the specifications of our proposed system. After successful validation of the design on an FPGA, it is realized using Semi Conductor Laboratory (SCL) 180nm Bulk CMOS technology.

3 MATHEMATICAL MODELING OF MULTI PENDULUM SYSTEM

The formulation of a nonlinear triple pendulum chaotic system is similar to a double-pendulum system. For simplicity and to limit usage of the variables in mathematical formulation, the construction of differential equations of a double pendulum system is demonstrated below. Figure 2 exhibits a basic double pendulum pictorially.

Fig. 2: Basic double pendulum setup

The basic position relations from the above figure are

$$\begin{aligned} x_1 &= L_1 \sin \theta_1 \\ x_2 &= L_1 \sin \theta_1 + L_2 \sin \theta_2 \\ y_1 &= -L_1 \cos \theta_1 \\ y_2 &= -L_1 \cos \theta_1 - L_2 \cos \theta_2 \end{aligned}$$

The expressions of potential (P) and kinetic (K) energies, as mentioned below, are obtained by analyzing position relations.

$$\begin{aligned} P &= -m_1 g L_1 \cos \theta_1 - m_2 g (L_1 \cos \theta_1 + L_2 \cos \theta_2) \\ K &= \frac{1}{2} m_1 \dot{\theta}_1^2 L_1^2 + \frac{1}{2} m_2 (\dot{\theta}_1^2 L_1^2 + \dot{\theta}_2^2 L_2^2 + 2 \dot{\theta}_1 L_1 \dot{\theta}_2 L_2 \cos(\theta_1 - \theta_2)) \end{aligned}$$

Later, Euler-Lagrange condition $\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta} = 0$ is applied, where $L = K - P$ and

$$L = \frac{1}{2} (m_1 + m_2) L_1^2 \dot{\theta}_1^2 + \frac{1}{2} m_2 L_2^2 \dot{\theta}_2^2 + m_2 L_1 L_2 \dot{\theta}_1 \dot{\theta}_2 \cos(\theta_1 + \theta_2) + (m_1 + m_2) g L_1 \cos \theta_1 + m_2 L_2 g \cos \theta_2.$$

Thereafter, substituting L into above mentioned Euler-Lagrange equation and solving the resultant equation for θ_1 and θ_2 , two second order differential equations are obtained.

$$\begin{aligned} \ddot{\theta}_1 &= [-m_2 L_1 \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) \cos(\theta_1 - \theta_2) + g m_2 \sin \theta_2 \cos(\theta_1 - \theta_2) - m_2 L_2 \dot{\theta}_2^2 \sin(\theta_1 - \theta_2) - (m_1 + m_2) g \sin \theta_1] / [L_1 (m_1 + m_2) - m_2 L_1 \cos^2(\theta_1 - \theta_2)], \text{ and} \\ \ddot{\theta}_2 &= [m_2 L_2 \dot{\theta}_2^2 \sin(\theta_1 - \theta_2) \cos(\theta_1 - \theta_2) + g \sin \theta_1 \cos(\theta_1 - \theta_2) (m_1 + m_2) + L_1 \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) (m_1 + m_2) - g \sin \theta_2 (m_1 + m_2)] / [L_2 (m_1 + m_2) - m_2 L_2 \cos^2(\theta_1 - \theta_2)] \end{aligned}$$

Formulation of Compound Triple Pendulum

As stated above, the steps of mathematical formulation of a triple pendulum system is similar to the double pendulum system. In this section, an abstract representation of mathematical formulation of triple pendulum system is presented. The bars of a triple-pendulum have significant mass, so that it can be modeled as a compound pendulum. Damping factors are also included in the model to introduce higher degree of chaos and nonlinearity in the system. Each bar i is defined by a set of four parameters mentioned below.

- m_i , the mass of the bar,
- l_i , the length of the bar,
- I_i , the moment of inertia of the bar and
- k_i , the damping coefficient of the bar rotating about its upper joint.

The position and velocity of the bars are defined by six system state variables, $\theta_1, \theta_2, \theta_3, \dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3$. The relevant governing equations to deduce these variable are given below.

$$y_1 = \frac{l_1}{2} \cos \theta_1 \quad (1)$$

$$y_2 = l_1 \cos \theta_1 + \frac{l_2}{2} \cos \theta_2 \quad (2)$$

$$y_3 = l_1 \cos \theta_1 + l_2 \cos \theta_2 + \frac{l_3}{2} \cos \theta_3 \quad (3)$$

$$x_1 = \frac{l_1}{2} \sin \theta_1 \quad (4)$$

$$x_2 = l_1 \sin \theta_1 + \frac{l_2}{2} \sin \theta_2 \quad (5)$$

$$x_3 = l_1 \sin \theta_1 + l_2 \sin \theta_2 + \frac{l_3}{2} \sin \theta_3 \quad (6)$$

The magnitude of the velocity of each bar can be expressed as

$$v_i = \sqrt{\dot{x}_i^2 + \dot{y}_i^2} \quad (7)$$

Fig. 3: Circular auto-correlation and periodicity test for two trajectories of θ_3

The translational and rotational kinetic energies, TKE_i and RKE_i , respectively, are depicted by

$$TKE_i = \frac{1}{2} m_i v_i^2 \quad (8)$$

$$RKE_i = \frac{1}{2} I_i \dot{\theta}_i^2 \quad (9)$$

The gravitational potential energy, GPE_i , can be presented as

$$GPE_i = m_i g y_i \quad (10)$$

Lagrangian of the system is expressed below.

$$\mathcal{L} = T - V = \sum_{i=1}^3 TKE_i + RKE_i - GPE_i \quad (11)$$

Lagrange's equation can be defined as

$$\frac{d}{dt} \left(\frac{d\mathcal{L}}{dq_i} \right) = \frac{d\mathcal{L}}{dq_i} - \frac{dD}{dq_i} \quad (12)$$

Rayleigh Dissipation Function, D , is described below.

$$D = \frac{1}{2} \left(\sum_{i=1}^3 k_i \dot{\theta}_i^2 \right) \quad (13)$$

Based on the above mentioned equations, a compound triple-pendulum model is numerically analyzed using MATLAB [25] employing ordinary differential equations (ODEs) used to describe random motions. The parameters and initial conditions of these ODEs are given in Table 1. For the analysis, numerical methods are utilized to solve a state-space model of differential equations, which are given in Appendix A. The numerical values corresponding to angular positions of the bars are obtained within a certain duration as per the predefined precision. This generates a chaotic map for an encryption module. The equations of $\ddot{\theta}_1$, $\ddot{\theta}_2$ and $\ddot{\theta}_3$ stated in Appendix A can be simplified as follows. $\ddot{\theta}_1$, $\ddot{\theta}_2$ and $\ddot{\theta}_3$ are evaluated using differential equations by iterating from $t = 0$ to $t = N$, where $N = \frac{T}{\Delta t}$.

$$\begin{aligned} \ddot{\theta}_1^{(t+1)} &= f_1(\theta_1^{(t)}, \theta_2^{(t)}, \theta_3^{(t)}, \dot{\theta}_1^{(t)}, \dot{\theta}_2^{(t)}, \dot{\theta}_3^{(t)}) \\ \ddot{\theta}_2^{(t+1)} &= f_2(\theta_1^{(t)}, \theta_2^{(t)}, \theta_3^{(t)}, \dot{\theta}_1^{(t)}, \dot{\theta}_2^{(t)}, \dot{\theta}_3^{(t)}) \\ \ddot{\theta}_3^{(t+1)} &= f_3(\theta_1^{(t)}, \theta_2^{(t)}, \theta_3^{(t)}, \dot{\theta}_1^{(t)}, \dot{\theta}_2^{(t)}, \dot{\theta}_3^{(t)}) \\ \dot{\theta}_1^{(t+1)} &\leftarrow \dot{\theta}_1^{(t)} + \ddot{\theta}_1^{(t+1)} \Delta t \\ \dot{\theta}_2^{(t+1)} &\leftarrow \dot{\theta}_2^{(t)} + \ddot{\theta}_2^{(t+1)} \Delta t \\ \dot{\theta}_3^{(t+1)} &\leftarrow \dot{\theta}_3^{(t)} + \ddot{\theta}_3^{(t+1)} \Delta t \\ \theta_1^{(t+1)} &\leftarrow \theta_1^{(t)} + \dot{\theta}_1^{(t+1)} \Delta t \\ \theta_2^{(t+1)} &\leftarrow \theta_2^{(t)} + \dot{\theta}_2^{(t+1)} \Delta t \\ \theta_3^{(t+1)} &\leftarrow \theta_3^{(t)} + \dot{\theta}_3^{(t+1)} \Delta t \end{aligned} \quad (14)$$

Later, θ are stored in a vector y , such that,

$$y^i = \{\theta_i^{(0)}, \theta_i^{(1)}, \theta_i^{(2)}, \dots, \theta_i^{(N-1)}\}, \text{ for } i = 1, 2, 3 \text{ and } N = \frac{T}{\Delta t}$$

Subsequently, y is phase wrapped in the range $[0, 2\pi]$ (or $[-\pi, \pi]$) to achieve,

$$\hat{y}^i = \{\hat{\theta}_i^{(0)}, \hat{\theta}_i^{(1)}, \hat{\theta}_i^{(2)}, \dots, \hat{\theta}_i^{(N-1)}\} \text{ for } i = 1, 2, 3$$

$$\text{where, } \hat{\theta}_i^{(t)} = \theta_i^{(t)} - \lfloor \frac{\theta_i^{(t)}}{2\pi} \rfloor \times 2\pi$$

It is to mention that state variables $\ddot{\theta}_1$, $\ddot{\theta}_2$ and $\ddot{\theta}_3$ are used to generate a one-to-one function F , such that,

$$\hat{y} = F(\hat{y}^1, \hat{y}^2, \hat{y}^3)$$

The Figure 3 presents a circular auto-correlation, estimated power of θ_3 with respect to its periodicity for periodic and nonperiodic trajectories. Periodic nature of circular auto-correlation signifies overall periodic nature of the trajectories and vice-versa. It can be observed from Figure 3(a) and Figure 3(b) that the power is concentrated around a few frequencies for periodic signals, whereas, power is distributed in the whole domain for aperiodic signals as shown in Figure 3(c) and Figure 3(d), respectively.

Fisher's G-statistic tests are also performed to further visualize similar properties of periodic and non-periodic trajectories, which are depicted in Figure 4. These tests, which are illustrated in Figure 4, are periodogram, bob's path traced and the time variation of angles. Periodogram is the power spectral density at each frequency of a bob. Bob's

Fig. 4: Fisher's G-statistic test with a periodic trajectory (a,b,c) and with a nonperiodic trajectory (d,e,f)

path traced during chaotic oscillations presents an actual physical path traced in 2D plane. The time variation of angles exhibits variations in $(\theta_1, \theta_2, \theta_3)$ with respect to time. The numerical values obtained from periodicity test, circular auto-correlation and other analyses clearly differentiates parameters and initial values, which leads to periodic and aperiodic nature of the motion. Thus, by iterating through all possible parameters emphasizing nonperiodicity in the selection, a set of keys are generated. In the following section, an implementation of our proposed triple pendulum system is presented to validate its correctness along with a study on its chaotic behavior.

4 KEY GENERATION AND ENCRYPTION-DECRYPTION METHODOLOGY

Fig. 5: Dynamics of Lyapunov exponents time $t = 0$ to 8 sec.

In this section, main constituents of our proposed cryptosystem, i.e. key generation and encryption-decryption methodology, are described in detail. The proposed triple pendulum system is implemented using MATLAB in order to validate correctness of the mathematical model presented in section 2. Employing initial conditions given in the Table 1, the phase wrapped angular displacement profiles of θ_1 , θ_2 and θ_3 are generated. It can be observed that the proposed system is chaotic in nature because of aperiodicity of angular displacements. Lyapunov exponents and bifurcation tests for all the key parameters are presented in Figures 5 and 6 to verify chaotic nature of our proposed system. The encryption scheme based on chaotic dynamic system presented in section 2 is validated by attractor profiling obtained using Lyapunov exponents and bifurcation diagrams. The chaotic nature of a dynamic system is estimated by the presence of positive exponents in the Lyapunov test, which implies exponential divergence of the system. It is to mention that chaotic systems do not synchronize with any other physical system in reality, although their synchronization conditions can be formulated mathematically iff all the initial conditions and parameters resemble exactly with the actual physical systems, which is only possible in an ideal scenario. Thus, two similar chaotic systems produce significantly different outputs, when initial conditions and parameters are marginally different. For estimating Lyapunov exponents [26] for an oscillating chaotic system, a 3D Lorenz equation 15 is formulated and is compared with different parameters of our proposed dynamic system. It can be observed from Figure 5 that our proposed triple pendulum based dynamic system fulfills Lyapunov criteria and can be considered as a chaotic system.

$$\dot{x} = \sigma(y - x); \dot{y} = x(\rho - z) - y; \dot{z} = xy - \beta z \quad (15)$$

Thereafter, a bifurcation diagrams are generated for all the independent parameters $m_1, m_2, m_3, l_1, l_2, l_3, k_2$ and k_3 for $\theta_1, \theta_1, \theta_2, \theta_2, \theta_3$ and $\dot{\theta}_3$ each. It is to mention that generally chaotic nature of systems depends on a single parameter and its bifurcation diagram is produced by changing the value of that particular parameter. Our proposed chaotic dynamic system has total 12 parameters constituting an entire encryption space, and eight parameters among them exhibit highly chaotic behavior. All the bifurcation diagrams illustrated in Figure 6 are generated by solving $\dot{\theta}$ using initial conditions of Table 1. Using Figure 6, range of each key values employed for encryption and decryption can be determined. It is to mention that in Figure 6, θ_1 and $\dot{\theta}_1, \theta_2$ and $\dot{\theta}_2$, and θ_3 and $\dot{\theta}_3$ are indicated with green, red and blue colors, respectively.

TABLE 1: Key parameters value and initial condition

Parameter/ Condition	Value	Unit
m_1, m_2, m_3	0.2944, 0.1756, 0.0947	kg
l_1, l_2, l_3	0.508, 0.254, 0.127	m
I_1, I_2, I_3	$(9.526, 1.625, 1.848) \times 10^{-3}$	kg.m ²
k_1, k_2, k_3	$(5, 0, 0.8) \times 10^{-3}$	N.m.s/rad
$\theta_1, \theta_2, \theta_3$	-0.4603, -1.2051, -1.5165	rad
$\dot{\theta}_n$	0	rad/s

4.1 Key Generation

The key, K , of proposed symmetric cryptosystem includes parameters, such as initial conditions, time duration, time step, the minimum value of $\hat{y}$, and ϵ . The ϵ is a mapping interval and can be expressed as $\epsilon = \frac{(\hat{y}_{max} - \hat{y}_{min})}{N_c}$. There are total 21 variables constituting basic key structure shown below.

$$K = \{\theta_1^{(0)}, \theta_2^{(0)}, \theta_3^{(0)}, \dot{\theta}_1^{(0)}, \dot{\theta}_2^{(0)}, \dot{\theta}_3^{(0)}, m_1, m_2, m_3, l_1, l_2, l_3, I_1, I_2, I_3, k_1, k_2, k_3, g, \hat{y}_{min}, \hat{y}_{min}/\epsilon\}$$

TABLE 2: Size of the key for various PQC algorithms [27]

Algorithm	Symmetric Key size(bits)	Asymmetric Key size(bits)	Key Memory
Lattice Multivariate	14000	6595	<1KB
NTRU	6743	6130	<1KB
Rainbow	740000	991000	125KB
Hash Signature	36000	36000	5KB
Goppa McEliece	92027	8373911	1MB
Quasi-cyclic MDPC McEliece	4384	65542	8KB
SIDH	6144	6144	<1KB
Proposed (single precision)*	672	-	<1KB
Proposed (double precision)*	1344	-	<1KB

* implement for symmetric key

Table 2 presents a comparison of key size and key memory for various PQC algorithms. It can be observed that the key size generated by our proposed method is the least amongst all other PQCs. It can be seen that for certain specific parameters or initial conditions, motion of the bobs of a triple-pendulum exhibits periodic nature after a certain

span of time. Hence, it becomes imperative to eliminate those parameters or initial conditions for which motion is periodic to preserve chaotic behavior of the system. Therefore, *Fisher's g-statistic test* [28], is employed to extract prominent period of the signal using statistical analysis on the spectrum of a signal. This test exploits periodic components of the signal derived from its periodogram, which is used to identify hidden periods (or frequencies) of a time series. This is helpful in identifying dominant periodic behavior in a series. The periodogram indicates relative occurrence of possible frequencies oscillating in between a given interval repeatedly. It estimates *Power Spectral Density* of a signal $x_L(n)$ of length L , which is defined below. Here, F_s is the sampling frequency.

$$P_{xx}(f) = \frac{1}{LF_s} \left| \sum_{n=0}^{L-1} x_L(n) e^{-j2\pi f n / F_s} \right|^2 \quad (16)$$

The computation of $P_{xx}(f)$ can be performed only at a finite number of discrete frequency points given by

$$f_k = \frac{kF_s}{N}, k = 0, 1, \dots, N-1 \quad (17)$$

Fisher's g -statistic is defined as the ratio of the maximum periodogram value ($P_{xx}(f_k)$ or $I(\omega_k)$) to the sum of all periodogram values of a signal.

$$g = \frac{\max_k I(\omega_k)}{\sum_{k=1}^{\lfloor N/2 \rfloor} I(\omega_k)} \quad (18)$$

where, $I(\omega_k)$ are periodogram values. Using the values of g , dominant periods of a signal are calculated. Thus, after removing the range of the parameters, in which a signal exhibits possible periodic behavior, remaining values of the parameters can be utilized to span key space. Further, based on the bifurcation diagrams, the strongest key may be chosen for encryption and decryption steps. Since bifurcation is a resource intensive process, there is a tradeoff between power and choosing a stronger key for an application. In the following subsection encryption and decryption methods are explained in detail.

4.2 Encryption - Decryption

Algorithm 1: Proposed encryption algorithm

Input: $\hat{y}, P, N_c, \eta$

Output: C

- 1 Initialize $D = \{\}$
- 2 Find $\hat{y}_{max} = \max_{1 \leq i \leq N}(\hat{y}_i)$, $\hat{y}_{min} = \min_{1 \leq i \leq N}(\hat{y}_i)$
- 3 Calculate $\epsilon = \frac{(\hat{y}_{max} - \hat{y}_{min})}{N_c}$
- 4 For each $\hat{y}_i$ in $\hat{y}$
 - Calculate $k = \lfloor \frac{\hat{y}_i}{\epsilon} \rfloor$
 - $D(\pi(k)) \leftarrow D(\pi(k)) \cup \{i\}$
- 5 For $m = 1$ to M
 - Initialize $j = 1, p = P(m)$
 - Calculate $\hat{p} = ASCII(p)$, $l = size(D(\pi(\hat{p})))$
 - Generate random number r
 - While $r \leq \eta$, do $j = (j + 1) \% l$ else
 - $C(m) = D(\pi(\hat{p}), j)$
- 6 Return C

Fig. 6: Bifurcation diagrams of important independent key parameters $\{m_1, m_2, m_3, l_1, l_2, l_3, k_2 \text{ and } k_3\}$ for $\{\theta_1, \dot{\theta}_1$ in green $\}$, $\{\theta_2, \dot{\theta}_2$ in red $\}$ and $\{\theta_3, \dot{\theta}_3$ in blue $\}$ [row-wise left to right sequence]

Algorithm 2: Proposed decryption algorithm

- Input:** $\hat{y}, \tilde{C}, N_c$
Output: $\tilde{P}$
- 1 Find $\hat{y}_{max} = \max_{1 \leq i \leq N}(\hat{y}_i)$, $\hat{y}_{min} = \min_{1 \leq i \leq N}(\hat{y}_i)$
 - 2 Calculate $\epsilon = \frac{(\hat{y}_{max} - \hat{y}_{min})}{N_c}$
 - 3 For $m = 1$ to M
 Initialize $c = C(m)$
 Calculate $\hat{k} = \lfloor \frac{\hat{y}_c}{\epsilon} \rfloor$
 Compute $k = \pi^{-1}(\hat{k})$
 $\tilde{P}(m) \leftarrow CHAR(k)$ Skip the step for image input
 - 4 Return $\tilde{P}$
-

The approach discussed in this paper is to convert plain text of M characters into an ASCII format and map it to the intervals formulated using state variables of the proposed triple-pendulum system. Figure 7 exhibits flow of this ap-

Fig. 7: Encryption-Decryption Strategy

proach. The initial conditions and parameters of different equations constitute a part of the private key. The differential equations are numerically integrated to get state variable $\theta_1, \theta_2, \theta_3$ followed by phase wrapping in the range of $[0, 2\pi]$ or $[-\pi, \pi]$. Later, employing Baptista-type method, an entire range of chaotic function $[\hat{y}_{min}, \hat{y}_{max}]$ is partitioned into a number of intervals equal to the number of characters (N_c). Each character in the plain-text P is first mapped to a specific interval (k) and is stamped with a time point (i) randomly selected from this interval. A permutation map $\pi(k)$ is employed to determine interval identifiers for each character's ASCII value. Proposed encryption scheme is

TABLE 3: Encryption decryption example with three different input image for a particular key value

Original image	Original histogram	Encrypted image	Encrypted histogram	Decrypted image	Decrypted histogram

depicted in Algorithm 1. Here, η is a constant and C is cipher text.

In the decryption module, an interval, in which an encrypted value lies, is computed using the same symmetric key. By applying inverse permutation map $(\pi^{-1}(k))$, corresponding indices are converted back to clear-text ($\tilde{P}$). The message can be decoded by converting $\tilde{P}$ into characters. Proposed decryption scheme is presented in Algorithm 2.

In addition to the plain text encryption, the proposed method can also be applied to images with RGB values ranging from 0 to 255. The parameter N_c in Algorithm 1 can be set to 256 and a serialized image is fed as an input for image encryption. This algorithm can be extended for video encryption as well by treating each frame as an independent image. Our proposed encryption and decryption algorithms are demonstrated on three set of images. The original, encrypted and decrypted images are illustrated with their corresponding histograms presented in Table 3. The parameters of a key used for encryption and decryption are $m_1 = 0.2944$, $m_2 = 0.8756$, $m_3 = 0.0947$, $l_1 = 0.508$, $l_2 = 0.254$, $l_3 = 0.127$, $k_1 = 0.005$, $k_2 = 0$, $k_3 = 0.0008$, $I_1 = 9.526 \times 10^{-3}$, $I_2 = 1.625 \times 10^{-3}$, $I_3 = 1.848 \times 10^{-4}$, $g = 9.81$. Selection of these parameters for a key is accomplished using bifurcation diagram shown in Figure 6. Although, histograms of input images are vastly different, but histograms of encrypted images are similar in nature. This makes proposed scheme less vulnerable as compared to other methods, because it is hard to determine input just by using an encrypted histogram without knowing key. It is to mention that the proposed scheme has wide range of applicability.

Complexity Analysis

The computational complexity of the proposed scheme is described below.

- The computation of state variables (θ) for a duration t and time step ΔT requires N iterations, where $N = \frac{t}{\Delta T}$. In each iteration, only six equations are analyzed to find six state variables ($\theta_1, \theta_2, \theta_3, \dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3$) in $O(1)$ time.

Thus, the time required for generating an asymptotical map is $O(N)$.

- The space complexity for storing state variables is $O(N)$.
- Baptista type partitioning takes constant time, if range of the variables is already computed in the previous step. For assigning variables to corresponding intervals, $O(N)$ time is required for each variable. The $K_c = 21$ variables constituting symmetric key in our proposed method, $K_c O(N)$ time is required.
- The encryption of a plain-text having M characters utilizes M iterations. In each iteration, an encrypted value is selected randomly in an interval in a constant time. Therefore, time complexities for setting up of an encryption module and processing each plaintext through are $O(N)$ and $O(M)$, respectively.
- Space complexity of an encryption module is $O(N)$.
- The time complexities of setting up decryption module and its execution are $O(N)$ and $O(M)$, respectively. The information rate of the proposed cryptosystem is defined as the ratio of the size of plain-text, SZ_{pt} , to the size of cipher-text, SZ_{ct} , which is defined below.

$$R = \frac{SZ_{pt}}{SZ_{ct}} = \frac{8 * M}{\lceil \log_2 N \rceil * M} = \frac{8}{\lceil \log_2 N \rceil} \quad (19)$$

The overall time complexity of the complete encryption and decryption methods is the sum of individual complexities mentioned above, which add upto $O(N)$. Therefore, it can be utilized in many applications efficiently to make them secure. Since, both time and space complexities of the proposed method are linear, it offers a very good scope for its parallelization to boost performance at hardware level as well. Due to its immunity to quantum computing based attacks, scalability and being computationally simple, our proposed method becomes a prominent candidate to provide much needed security to various wide range of applications, such as IoT based smart systems, high throughput low latency systems etc.

5 HARDWARE IMPLEMENTATION

The proposed design has been implemented using System Verilog HDL on Digilent Zed-Board having Xilinx Zynq-7 FPGA. Its synthesized netlist is generated by Synopsis Design Compiler (DC) employing SCL's 180nm design libraries. For simplicity and to validate the proposed algorithms, permutation map ($\pi(k)$) is fixed as an identity map and maximum three state variables are stored in a particular interval for each character during an encryption phase. As a result, it is required to search over total number of characters, N_c , for an interval during decryption cycle. For evaluating equation 14, its RHS expression is converted into a postfix format and the resultant postfix expression is stored in a ROM on an FPGA board. Later, this expression is evaluated using a stack-based postfix evaluation technique. The angle combinations of sinusoidal terms present in the expressions are cached in an LUT and are appropriately decoded while evaluating these expression. The overall block diagram of this scheme is presented in Figure 8 and its finite state machine is illustrated in Figure 9.

Fig. 8: Overall block diagram of proposed scheme

Fig. 9: Overview of FPGA Implementation

The hardware design can be segregated into three main parts, viz. LUT and Memory, ALU and Control Unit. A brief description of these units are elaborated in the following subsections.

5.1 Look Up Table (LUT) and Memory

In the FPGA implementation, five LUTs and a stack memory are utilized for solving differential equations by evaluating postfix expressions, which depends on angle combinations presented in Table 4. These five LUTs are used to store *Parameters*, *Angle Combinations*, *State Variables*, *Numeric Constants* and *Postfix Evaluations*. Parameters represented in Table 1 are employed for the above mentioned purpose.

TABLE 4: LUT Details

Angle Combinations	$t = 0$	Angle Combinations	$t = 0$
$2\theta_1 - 2\theta_3$	2.1124	θ_2	5.0781
$2\theta_1 - \theta_2$	1.4896	$\theta_2 - \theta_3$	0.3114
$\theta_1 - \theta_2$	0.7448	$2\theta_1 - \theta_2 - \theta_3$	1.801
$\theta_1 + \theta_2 - 2\theta_3$	1.3676	$2\theta_1 - \theta_2$	6.5677
$\theta_1 - \theta_3$	1.0562	$\theta_2 - 2\theta_3$	-4.4553
$\theta_1 - 2\theta_2 + \theta_3$	0.4334	$2\theta_1 + \theta_2 - 2\theta_3$	7.1905
θ_1	5.8229	θ_3	4.7667
$\theta_1 - 2\theta_3$	-3.7105	$2\theta_1 - \theta_3$	6.8791
$2\theta_2 - 2\theta_3$	0.6228	$2\theta_2 - \theta_3$	5.3895
$\theta_1 - 2\theta_2$	-4.3333	$2\theta_1 - 2\theta_2 + \theta_3$	6.2563

As it is mentioned earlier, in the implementation of postfix evaluations, RHS expressions in equation 14 are converted into postfix format. The resultant postfix expression constitutes numeric constants, state variables or parameters, operations and sinusoidal terms. Each of these terms is encoded to a 8-bit code and an entire postfix expression is stored in an LUT in the coded format. During postfix evaluation, these codes are decoded and an appropriate data value is fetched from a corresponding address. The 8-bit encoding format can be expressed as shown below.

0	1	2	3	4	5	6	7
Type	T/A	Address/Op					

The mathematical model of triple pendulum system presented in the Appendix of this manuscript is parsed and evaluated using information presented in Table 5. Thereafter, stack memory is configured to execute encryption and decryption modules.

TABLE 5: Encoding for mathematical operations using 8-bit postfix format

Type	T/A	Address/Operation	Remarks
00	X	XXXXX	Constant terms
01	0	0x00 to 0x1F (addr.)	Parameters from Table I & II
	1	0x00 to 0x1F (addr.)	State variables
10	0	0x00 to 0x1F (addr.)	Sine values from Table IV
	1	0x00 to 0x1F (addr.)	Cosine values from Table IV
01	X	XX000 (op.)	Exponentiation
		XX001 (op.)	Multiplication
		XX010 (op.)	Division
		XX011 (op.)	Addition
		XX100 (op.)	Subtraction
11	1	11111 (op.)	End of expressions

X = don't care; op. = operation; addr. = memory location of LUT

Stack: A stack consisting of two immediate registers and a memory is implemented for the evaluation of postfix expressions. The immediate registers store top two values of the stack. An operation is performed in an expression

Fig. 10: FSM for various control unit blocks

employing these two immediate registers and its result is stored in the first register. Another register is again filled with a value popped from top of the stack for next operation. It continues until all the operations related with a mathematical expression are completed.

When a complex operation requires more resources than two immediate registers, then this operation is performed in two or more than two steps. In this operation, value stored in the second immediate register is pushed onto the stack and, corresponding memory location and program counter are updated. This process repeats until a complex operation is executed producing a correct output. This approach reduces resource utilization while performing a complex operation. Another advantage of using stack-based push-pop architecture is that it can be used for solving any differential equation expressed in the form of state-space equation. For this operation, only angle combinations and mnemonics stored in the stack need to be changed keeping hardware architecture identical for all the operations. The brief discussion on Arithmetic and Logic Unit (ALU) is presented in the following section.

5.2 Arithmetic and Logic Unit (ALU)

In the proposed design, an arithmetic and logical unit (ALU) consists of two addition, one multiplication, one exponentiation and one division modules. All these modules perform floating point operations in the standard IEEE-754 format. The datapath in the addition module is split into four stages. The clock frequency of this operation is fixed at 100 MHz to meet timing constraints of the FPGA employed in validation of the proposed design. The datapath in multiplication module is divided into five stages to achieve above mentioned clock frequency. Moreover, if required, the multiplication module can be configured to function in a pipelined mode. Exponentiation module evaluates value of a floating point input raised to an integer as an exponent. It instantiates multiplication module internally for the evaluation of an exponent. Datapath of division module is sliced into multiple stages to meet the target clock frequency of 100 MHz. However, the division operation requires significant clock cycles as compared to other operations, therefore, the design of proposed algorithms is optimized to keep the number of division operations as less as possible.

5.3 Control Unit

The control unit of the proposed design is essentially a finite state machine (FSM), which performs required tasks in a

well-defined sequence. In order to implement it efficiently, the complete FSM is factored into smaller state machines hierarchically. The top level FSM serves as a controller for all the smaller FSMs. Detailed descriptions for these state machines are given below.

- (i) **Top Level State Machine:** Figure 10(a) exhibits a simplified FSM of the top module. Description of each state is given below.
 - **DEFAULT:** It is an initial state of FSM after reset.
 - **RECEIVE_KEY:** It receives encryption or decryption keys through USB or Ethernet interfaces.
 - **EVALUATE_EXPRESSION:** It evaluates state-space expression using given initial conditions for a chaotic dynamic system.
 - **MODIFY_STATE_VARIABLES:** In this state, state variables are updated based on the previous state. It involves multiply and addition modules to perform updated operations.
 - **OBTAIN_CHAOTIC_MAP:** It evaluates a chaotic map through a linear combination of all state variables and calculates a linear combination of θ , $\dot{\theta}$ and $\ddot{\theta}$. It is to mention that θ corresponds to the chaotic map. Further, it calculates intervals, in which θ lies and adds it to corresponding address in LUT.
 - **GENERATE_CIPHERTEXT:** It generates a ciphertext for a particular plaintext string using a generated chaotic map and utilizes LUT to convert plaintext to ciphertext.
- (ii) **Evaluate State-Space Expression:** Figure 10(b) exhibits an FSM of a state space solver module and its detailed description is presented below.
 - **DEFAULT:** This is an initial state of FSM after reset.
 - **FETCH_INITIAL_CONDITIONS:** It fetches required initial conditions, θ and $\dot{\theta}$, from Block-RAM (BRAM).
 - **ANGLE_COMBINATION_CALC:** This state generates linear combinations of state variables.
 - **ANGLE_NORMALIZATION:** In this state, trigonometric inputs are normalized in the range of $[-\pi, \pi]$.
 - **POSTFIX_TERM_ACCUMULATE:** It evaluates a state-space expression stored in the postfix form.
 - **STORE_OUTPUT_DATA:** This state stores state-space expression obtained in the previous state in the memory.
- (iii) **Postfix Expression Evaluation:** Figure 10(c) presents an FSM of postfix expression evaluation, which outputs

TABLE 6: FPGA implementation details of proposed chaos based cryptosystem (NS means not specified)

Parameters		Adrián et al. [29]	Pérez-Resa et al. [30]	Richard ¹ et al. [31]	Richard ² et al. [31]	Proposed Scheme
Resources	LUT	10943	16978	7426	3587	2955
	LUTRAM	NS	NS	NS	NS	87
	Flip-Flop	3629	11127	11285	5592	3519
	Slices	3190	5636	3268	1596	3237
	BRAM	0	77	343	170	5
	DSP-block	144	0	0	0	8
BUFG		NS	NS	NS	NS	2
Total Resource Required		17906	33818	22322	10945	9813
Comparative Resource Utilization		1.825X	3.446X	2.275X	1.115X	X
Power	Dynamic + Static	332 mW	775 mW	NS	NS	186 mW
Delay	Logic + Network	192 nS	8 nS	NS	NS	1.5920 nS
Power-Delay Product		63.744 nJ	6.2 nJ	NS	NS	0.296 nJ
Throughput Achieved (MBps)		1000 Mbps	1000 Mbps	41.1 Mbps	109.6 Mbps	2396.164 MBps
Operating Frequency (MHz)		125 MHz	125 MHz	279.6 MHz	283.5 MHz	628.14 MHz

encrypted/decrypted values. A description of each state is described below.

- **DEFAULT:** This is an initial state of FSM after reset.
- **READ_POSTFIX_CODE:** In this state, mnemonic of the next term of a postfix expression is fetched from the stack.
- **DATA OR OPERATOR:** This state checks whether a fetched code corresponds to data or an operator.
- **OPERATE_ON_TOP_TWO_STACK_VALUES:** It performs an operation based on top two data values stored in a stack.
- **PUSH_DATA_TO_STACK:** It pushes data in a postfix expression or result of an ALU operation onto the stack.

It is to mention that the differentiating factor between encryption and decryption operations in *Postfix Expression Evaluation* FSM depends on postfix expressions, which are stored in postfix evaluation LUTs. Employing the above mentioned FSMs, encryption and decryption algorithms proposed in this paper are implemented and validated against various benchmarks. In the next section, implementation results of these methods on various platforms are compared and discussed in detail.

6 RESULTS AND ANALYSIS

In this section, we present hardware implementation results and relevant tests for the randomness quality of a chaotic map generated by the proposed approach.

6.1 FPGA and ASIC Implementation Results

The detailed analysis of the proposed encryption and decryption algorithms is presented in section 4 to validate their correctness and efficacy. Later, these methods are implemented on hardware using System Verilog and is synthesized using Xilinx Vivado [32]. It is to mention that the results obtained using hardware implementation of these algorithms match with their MATLAB based implementations accurately.

Table 6 exhibits a detailed comparison and resource utilization of our hardware implementation with other prior art work, such as Adrián et al. [29], Pérez-Resa et al. [30] and two variants of Richard et al. [31] etc., reported in the literature. It can be observed that the proposed design

utilizes 45.20% and 71% less resources as compared to chaos based symmetric streaming encryption methods [29] and [30], respectively. It also uses 56% and 10.35% less resources than the above mentioned two variants of format preserving encryption algorithm presented in Richard et al. [31], respectively. It is to mention that our proposed method exploits maximum hardware resources, memory and LUTs, in storing state variables only. Although it enables proposed design to be memory optimized, but most of the static power is consumed to store content in memory. Since FPGA is the target device, a few DSP slices (DSP48) are employed during synthesis of the proposed design, which enable efficient implementation of trigonometric functions.

It can be observed further that the power consumption of our proposed design is 186 mW. Its throughput and theoretical operating frequency are 2396.164 MBps and 628.14 MHz, respectively. It has the least power-delay product when compared with other designs, which indicates that our proposed design is energy efficient as well. Thus, the high throughput of our design along with a very high energy efficiency and very high strength against attacks make it a suitable candidate to be used in the wide range of electronics systems ranging from smart IoT devices to low latency high performance systems. Further, the proposed

Fig. 11: Chip layout at 180nm Bulk CMOS technology node design is realized employing 180nm Bulk CMOS technology available with Semi-Conductor Lab, India. Table 7 presents actual physical area, power and resource utilization esti-

Fig. 12: Power profile of the proposed design at various operating frequencies (1KHz to 500MHz)

mated using above mentioned technology node at 250MHz clock. The chip layout of the proposed design illustrated in Figure 11 is developed by using Electronic Design Automation (EDA) tool, IC Compiler, provided by Synopsys Inc. Similarly, the synthesized design netlist is generated by employing Design Compiler provided by Synopsys Inc. It is to mention that all the design parameters exhibited in Table 7 are extracted through IC Compiler. In order to validate

TABLE 7: Actual resource utilization for proposed cryptosystem based on SCL's 180nm technology node at 250MHz

Parameters	Value
Total std. cells	6229
Combinational cells	5086
Sequential cells	1105
Number of buf/inv	1075
Number of references	38
Combinational area	113800.501382 μm^2
Buf/Inv area	9217.829951 μm^2
Non combinational area	83402.028854 μm^2
Net Interconnect area	6533.662340 μm^2
Total cell area	197202.530237 μm^2
Total area	203736.192577 μm^2
Internal Power	51.1217 mW
Switching Power	10.7619 mW
Leakage Power	1.0772 μW
Total Power	61.8836 mW

usage of our proposed design in various applications, its power profile is analyzed by varying operating frequency from 1KHz to 500 MHz, which is presented in Figure 12. The total power measured at a particular frequency is the sum of static, dynamic and leakage power. It can be also be observed in Figure 12 that the static power contributes significantly to the total power. Below 100 KHz, leakage power becomes comparable to the dynamic power, which is not recommended for reliable circuit performance at 180nm technology. Above 100 KHz, leakage power is not significant as compared to static and dynamic power. It is to further mention that the leakage power does not rise substantially at 500 MHz as well. This implies that our proposed design can be used reliably at more than 100 KHz to high operating frequencies at 180nm technology node. With the smaller technology nodes, the operating range is anticipated to be different.

6.2 Chaos Analysis for Security Estimation

In order to test randomness of a chaotic map generated in the proposed method, various statistical tests are performed. These statistical tests are generally employed to validate randomness in pseudo random-number generators. As we know, these chaotic maps are used to generate keys for encryption and decryption schemes. Therefore, the statistical tests can ensure whether the keys generated in our proposed scheme are ideally indistinguishable from any true random sequence or not. The statistical tests, which are conducted to support our claims, are described below.

TABLE 8: Comparison of proposed with other existing schemes PRNGs Based On NIST SP800 – 22 Test

NIST Tests	p-value ($0.01 \leq p\text{-value} < 1$)		
	Hua et al [33]	Srisubha et al [34]	Proposed
Frequency	0.364146	0.49	0.394808
Block Frequency	0.427274	0.33	0.783942
Cumulative Sums	0.670408	0.59	0.197764
Runs	0.213309	0.45	0.335584
Longest Run of 1's	0.204076	0.53	0.284067
Rank	0.819544	0.39	0.197764
DFT/Spectral	0.407091	0.41	0.769057
Non-periodic T.M.	0.472933	0.49	0.278074
Overlapping T.M.	0.602458	0.61	0.554566
Universal Statistical	0.772760	0.29	0.477911
Approx. Entropy	0.287306	0.60	0.365177
Random Excursions	0.442347	0.38	0.513954
Random Excursions Variant	0.356540	0.35	0.242148
Serial	0.175321	0.57	0.156848
Linear Complexity	0.264458	0.58	0.129930

TABLE 9: Diehard Test

test_name	ntup	tsamples	psamples	p-value
diehard_birthdays	0	100	100	0.98409924
diehard_operm5	0	1000000	100	0.02570841
diehard_rank_32x32	0	40000	100	0.30314396
diehard_bitstream	0	2097152	100	0.83264505
diehard_opso	0	2097152	100	0.93701062
diehard_oqso	0	2097152	100	0.63759752
diehard_dna	0	2097152	100	0.73795350
diehard_count_1s_str	0	256000	100	0.77268562
diehard_count_1s_byt	0	256000	100	0.55542008
diehard_parking_lot	0	12000	100	0.70977074
diehard_2dsphere	2	8000	100	0.58428028
diehard_3dsphere	3	4000	100	0.86446205
diehard_squeeze	0	100000	100	0.19422339
diehard_sums	0	100	100	0.33263813
diehard_runs	0	100000	100	0.50609448
diehard_craps	0	200000	100	0.36377661
marsaglia_tsang_gcd	0	10000000	100	0.66121444
sts_monobit	1	100000	100	0.31074688
sts_serial	16	100000	100	0.83380616
rgb_bitdist	12	100000	100	0.64865649
rgb_minimum_distance	2	10000	1000	0.41755433
rgb_permutations	2	100000	100	0.79654377
rgb_lagged_sum	16	1000000	100	0.78219023
rgb_kstest_test	0	10000	1000	0.58005862
dab_bytedistrib	0	51200000	1	0.01089458
dab_dct	256	50000	1	0.06122408
dab_filltree	32	15000000	1	0.12524761
dab_monobit2	12	65000000	1	0.27940682

6.2.1 Test for Randomness

NIST suite [35] and Diehard random number generator [36] test suites are employed to test randomness properties of a chaotic map. The information about tests performed on

Fig. 13: Sensitivity to parameter value m_1 and θ_1 with $\Delta m = 10^{-6}$ and $\Delta \theta_1 = 10^{-6}$

the chaotic map generated during analysis and the corresponding results obtained are presented in Table 8 and Table 9, respectively. In the NIST test, our proposed chaotic generator is compared with two recently published methods described in [33] and [34] to test its randomness. In [33], a sine chaotification model is employed to generate 1-D chaotic map, whereas, [34] presents physically unclonable functions (PUF) based chaotic maps. The [33] and [34] are used so that our proposed method is compared fairly with widely used chaotic schemes.

For Diehard test, only self-test is performed due to the unavailability of similar tests in the literature to the best of our knowledge. From Tables 8 and 9, it can be concluded that the proposed chaotic map generator performs well among all the tests. This is largely due to nonlinear dynamics of chaotic map. The outcome of these tests indicate that the proposed chaotic map is indistinguishable with respect to any other random sequence generators depicted in [33] and [34].

6.2.2 Test for Chaos

There are a few essential conditions, which need to be met by any chaos based cryptosystem. These conditions include ergodicity, sensitivity to parametric values and sensitivity to initial condition to be as unpredictable as possible. For completeness, these essential conditions are described below.

- (i) **Sensitivity to Parametric values:** A small perturbation in one of the system parameters, for example m_1 , is enough to create two exponential diverging trajectories starting at the same initial point. This is reflected in Figure 13(a) and Figure 13(b) for the parameters given in Table 1.
- (ii) **Sensitivity to Initial Condition:** Two trajectories starting at two different but arbitrarily close initial points denoted by θ_1 diverge from each other at an exponential rate. This is illustrated in Figure 13(a) and Figure 13(c) for the initial conditions given in Table 1.
- (iii) **Ergodicity:** Ergodicity is the phase space exploration either uniformly or randomly by a chaotic system. Here, phase space is defined as a relationship between angular velocity, ω , and the angle of pendulum's bob, ϕ . The triple pendulum system exhibits ergodic nature because phase space is sparse and is not cluttered in any specific region. The phase space of the bottom bob

of the proposed triple pendulum system illustrated in Figure 14 under conditions given in Table 1 exhibits a trajectory, which satisfies ergodicity criteria. This leads to a conclusion that a system under specific conditions can be considered ergodic.

Fig. 14: Phase space plot for ergodicity test for θ_3

The proposed triple pendulum based cryptosystem holds above mentioned properties, which are clearly evident in Figure 14 and Figure 13. It is to mention that the map generated using proposed compound triple pendulum model possesses essential chaotic properties to be employed in a general cryptosystem.

6.2.3 Collision Test

Collision resistance is the property of cryptographic algorithms, which makes it difficult to find two inputs that give the same encrypted output. It is well known that finding collisions is computationally very hard. In this paper, it is demonstrated that the proposed algorithm is fully collision resistant. As it is discussed earlier, an entire range of $\hat{y}$ are partitioned according to number of characters. For example, $N_c = 256$, i.e. $[\hat{y}_{min} + k\epsilon, \hat{y}_{min} + (k+1)\epsilon)$ for $k = 0, 1, \dots, 255$, which are mapped to different intervals. In the proposed method, two different inputs having different characters are mapped to different intervals always. Thus, it can be considered as collision resistant.

Based on the experimental and analytical results, it can be emphasized that the proposed scheme is better as compared to any other chaos based PQC algorithms. Because of smaller key size, optimal area, high throughput and low power consumption, it can also be used efficiently to enhance security of IoT devices as well as any low latency high performance secure application.

7 CONCLUSION

The work presented in this paper discusses a novel symmetric key cryptography scheme based on a nonlinear triple pendulum chaotic map. The detailed analysis and standard tests of the proposed method classify it to be considered as a potential post quantum cryptography (PQC) algorithm. The proposed encryption scheme is compared with various other chaos based schemes over standard security parameters to prove its effectiveness. The quantum-safe property of this method is due to its innate chaotic nature. Also, the highly nonlinear and interdependent unpredictable nature of our proposed scheme makes it difficult even for quantum computing based attacks to breach the security without complete knowledge of its key. It is also unbreakable practically with the partial knowledge of its key because of high sensitivity of chaotic maps to key values. An FPGA implementation of the proposed cryptosystem exhibit 45.64% less resource utilization, 33.60% less power consumption on an average and more than double throughput as compared to similar cryptosystem. The proposed cryptosystem is further implemented using 180nm Bulk CMOS technology occupying 0.20374 mm² die area and consuming 61.88 mW power at 250MHz at 180nm. It is shown that this system can operate from low (at least 100KHz) to high operating frequencies efficiently without showing any performance deterioration. The efficient area utilization of the proposed post quantum cryptosystem with high throughput at a very low power enables us to utilize it not only in the IoT based systems but also in the high performance applications to provide them desired strength against any malicious attack.

APPENDIX A

DIFFERENTIAL EQUATION FOR θ_1

$$\ddot{\theta}_1 = -(2((l_1^2 m_3^2 \sin(2\theta_1 - 2\theta_3)(4I_2 - l_2^2 m_2) + l_2^2 \sin(2\theta_1 - 2\theta_2)(m_2 + 2m_3)(m_2 m_3 l_3^2 + 4I_3(m_2 + 2m_3)))l_1^2 \dot{\theta}_1^2 + (l_2(\sin(\theta_1 - \theta_2)((m_2 m_3(m_2 + 3m_3)l_3^2 + 4I_3(m_2^2 + 6m_2 m_3 + 8m_3^2))l_2^2 + 4I_2(m_3(m_2 + m_3)l_3^2 + 4I_3(m_2 + 2m_3))) + l_2^2 m_3^2 \sin(\theta_1 + \theta_2 - 2\theta_3)(4I_2 - l_2^2 m_2)) \dots \theta_2^2 - 4k_2 l_2 (\cos(\theta_1 - \theta_2)(m_3(m_2 + m_3)l_3^2 + 4I_3(m_2 + 2m_3)) - l_2^2 m_3^2 \cos(\theta_1 + \theta_2 - 2\theta_3))\theta_2 + l_3 m_3 (\sin(\theta_1 - \theta_3)(8I_3 m_3 l_3^2 + 4I_3 m_3 l_3^2 + 16I_2 I_3) + l_2^2 \sin(\theta_1 - 2\theta_2 + \theta_3)(m_2 m_3 l_3^2 + 4I_3(m_2 + 2m_3)))\dot{\theta}_3^2 - 4k_3 l_3 m_3 (\cos(\theta_1 - \theta_3)(2m_3 l_2^2 + 4I_2) - l_2^2 \cos(\theta_1 - 2\theta_2 + \theta_3)(m_2 + 2m_3))\theta_3 - g(\sin(\theta_1)((m_3(m_1 m_2 + 2m_1 m_3 + 3m_2 m_3 + m_2^2)l_3^2 + 4I_3(m_2^2 + 6m_2 m_3 + m_1 m_2 + 4m_3^2 + 4m_1 m_3))l_2^2 + 4I_2(m_3(m_1 + 2m_2 + m_3)l_3^2 + 4I_3(m_1 + 2m_2 + 2m_3))) + l_2^2 m_3^2 (\sin(\theta_1 - 2\theta_3)(4I_2 l_2^2 m_2 - 2l_2^2 \cos(2\theta_2 - 2\theta_3) \sin(\theta_1)(m_1 + m_2)) + l_2^2 \sin(\theta_1 - 2\theta_2)(m_2 + 2m_3)(m_2 m_3 l_3^2 + 4I_3(m_2 + 2m_3)))l_1 + 2k_1(4I_2(m_3 l_3^2 + 4I_3) + l_2^2(m_3(m_2 + 2m_3)l_3^2 + 4I_3(m_2 + 4m_3)) - 2l_2^2 l_3^2 m_3^2 \cos(2\theta_2 - 2\theta_3))\theta_1)) / (64I_1 I_2 I_3 + 8I_3 l_1^2 l_2^2 m_2^2 + 8I_1 l_2^2 l_3^2 m_3^2 + 8I_2 l_1^2 l_3^2 m_3^2 + 32I_3 l_1^2 l_2^2 m_3^2 + 16I_2 I_3 l_1^2 m_1 + 16I_1 I_3 l_2^2 m_2 + 4I_1 l_2^2 l_3^2 m_2 m_3 +$$

$$16I_2 l_1^2 l_3^2 m_2 m_3 + 48I_3 l_1^2 l_2^2 m_2 m_3 - 8I_1 l_2^2 l_3^2 m_3^2 \cos(2\theta_2 - 2\theta_3) - 2l_1^2 l_2^2 \cos(2\theta_1 - 2\theta_2)(m_2 + 2m_3)(m_2 m_3 l_3^2 + 4I_3(m_2 + 2m_3)) - 2l_1^2 l_3^2 m_3^2 \cos(2\theta_1 - 2\theta_3)(-m_2 l_2^2 + 4I_2) + 2l_1^2 l_2^2 l_3^2 m_1 m_3^2 + 6l_1^2 l_2^2 l_3^2 m_2 m_3^2 + 2l_1^2 l_2^2 l_3^2 m_3^2 m_3 + l_1^2 l_2^2 l_3^2 m_1 m_2 m_3 - 2l_1^2 l_2^2 l_3^2 m_1 m_3^2 \cos(2\theta_2 - 2\theta_3) - 4l_1^2 l_2^2 l_3^2 m_2 m_3^2 \cos(2\theta_2 - 2\theta_3))$$

DIFFERENTIAL EQUATION FOR θ_2

$$\ddot{\theta}_2 = (2((l_1^2 \sin(2\theta_1 - 2\theta_2)(m_2 + 2m_3)(m_2 m_3 l_3^2 + 4I_3(m_2 + 2m_3)) - l_2^2 m_3^2 \sin(2\theta_2 - 2\theta_3)((m_1 + 2m_2)l_1^2 + 4I_1))l_2^2 \dot{\theta}_2^2 + l_1(\sin(\theta_1 - \theta_2)((m_3(m_1(m_2 + m_3) + 2m_2(2m_2 + 3m_3))l_3^2 + 4I_3(m_2 + 2m_3)(m_1 + 4m_2 + 4m_3))l_1^2 + 4I_1(m_3(m_2 + m_3)l_3^2 + 4I_3(m_2 + 2m_3))) - l_2^2 m_3^2 \sin(\theta_1 + \theta_2 - 2\theta_3)((m_1 + 2m_2)l_1^2 + 4I_1))l_2 \dot{\theta}_1^2 + 4k_1 l_1 (\cos(\theta_1 - \theta_2)(m_3(m_2 + m_3)l_3^2 + 4I_3(m_2 + 2m_3)) - l_2^2 m_3^2 \cos(\theta_1 + \theta_2 - 2\theta_3))l_2 \dot{\theta}_1 + (-l_3 m_3 (\sin(\theta_2 - \theta_3)((m_3(m_1 + 3m_2)l_3^2 + 4I_3(m_1 + 3m_2 + 2m_3))l_1^2 + 4I_1(m_3 l_3^2 + 4I_3)) - l_1^2 \sin(2\theta_1 - \theta_2 - \theta_3)(m_2 m_3 l_3^2 + 4I_3(m_2 + 2m_3)))\theta_3 + 4k_3 l_3 m_3 (\cos(\theta_2 - \theta_3)((m_1 + 3m_2 + 2m_3)l_1^2 + 4I_1) - l_1^2 \cos(2\theta_1 - \theta_2 - \theta_3)(m_2 + 2m_3))\theta_3 + g(\sin(\theta_2)((m_2 m_3(2m_2 + 3m_3)l_3^2 + 8I_3(m_2^2 + 3m_2 m_3 + 2m_3^2))l_1^2 + 4I_1(m_3(m_2 + m_3)l_3^2 + 4I_3(m_2 + 2m_3))) - l_1^2 \sin(2\theta_1 - \theta_2)(m_3(m_1(m_2 + m_3) + m_2(2m_2 + 3m_3))l_3^2 + 4I_3(m_2 + 2m_3)(m_1 + 2m_2 + 2m_3)) + l_2^2 m_3^2 (\sin(\theta_2 - 2\theta_3)(m_2 l_1^2 + 4I_1) + l_1^2 \sin(2\theta_1 + \theta_2 - 2\theta_3)(m_1 + m_2)))l_2 - 2k_2(4I_1(m_3 l_3^2 + 4I_3) + l_1^2(m_3(m_1 + 4m_2 + 2m_3)l_3^2 + 4I_3(m_1 + 4m_2 + 4m_3)) - 2l_1^2 l_3^2 m_3^2 \cos(2\theta_1 - 2\theta_3))\theta_2) / (64I_1 I_2 I_3 + 8I_3 l_1^2 l_2^2 m_2^2 + 8I_1 l_2^2 l_3^2 m_3^2 + 32I_3 l_1^2 l_2^2 m_3^2 + 16I_2 I_3 l_1^2 m_1 + 16I_1 I_3 l_2^2 m_2 + 4I_1 l_2^2 l_3^2 m_2 m_3 + 16I_2 l_1^2 l_3^2 m_2 m_3 + 48I_3 l_1^2 l_2^2 m_2 m_3 - 8I_1 l_2^2 l_3^2 m_3^2 \cos(2\theta_2 - 2\theta_3) - 3l_1^2 l_2^2 \cos(2\theta_1 - 2\theta_2)(m_2 + 2m_3)(m_2 m_3 l_3^2 + 4I_3(m_2 + 2m_3)) - 2l_1^2 l_2^2 m_3^2 \cos(2\theta_1 - 2\theta_3)(-m_2 l_2^2 + 4I_2) + 2l_1^2 l_2^2 l_3^2 m_1 m_3^2 + 6l_1^2 l_2^2 l_3^2 m_2 m_3^2 + 2l_1^2 l_2^2 l_3^2 m_2^2 m_3 + l_1^2 l_2^2 l_3^2 m_1 m_2 m_3 - 2l_1^2 l_2^2 l_3^2 - 2l_1^2 l_2^2 l_3^2 m_1 m_3^2 \cos(2\theta_2 - 2\theta_3) - 4l_1^2 l_2^2 l_3^2 m_2 m_3^2 \cos(2\theta_2 - 2\theta_3))$$

DIFFERENTIAL EQUATION FOR θ_3

$$\ddot{\theta}_3 = -(2(32I_1 I_2 k_3 \dot{\theta}_3 - l_2 l_3 m_3 \dot{\theta}_2^2 (\sin(\theta_1 - \theta_3)((l_2^2(m_1 m_2 + 4m_1 m_3 + 6m_2 m_3 + m_2^2) + 4I_2(m_1 + 3m_2 + 2m_3))l_1^2 + 4I_1(4I_2 + l_2^2(m_2 + 4m_3))) + l_1^2 \sin(2\theta_1 - \theta_2 - \theta_3)(m_2 + 2m_3)(4I_2 - m_2 l_2^2)) - l_1 l_3 m_3 \dot{\theta}_1^2 (\sin(\theta_1 - \theta_3)(8I_1(m_3 l_2^2 + 2I_2) + 2l_1^2((m_1 m_3 - m_2^2)l_2^2 + 2I_2(m_1 + 4m_2 + 4m_3))) - l_2^2 \sin(\theta_1 - 2\theta_2 + \theta_3)(m_2 + 2m_3)((m_1 + 2m_2)l_1^2 + 4I_1)) + 4k_3 l_1^2 l_2^2 m_2^2 \theta_3 + 16l_3 l_1^2 l_2^2 m_3^2 \theta_3 + 8I_2 k_3 l_1^2 m_1 \dot{\theta}_3 + 8I_1 k_3 l_2^2 m_2 \dot{\theta}_3 + 32I_2 k_3 l_1^2 m_2 \dot{\theta}_3 + 32I_1 k_3 l_2^2 m_3 \dot{\theta}_3 + 32I_2 k_3 l_1^2 m_3 \dot{\theta}_3 - 4k_1 l_1 k_3 m_3 \theta_1 (\cos(\theta_1 - \theta_3)(2m_3 l_2^2 + 4I_2) - l_2^2 \cos(\theta_2 - 2\theta_2 + \theta_3)(m_2 + 2m_3)) - 16I_1 I_2 g l_3 m_3 \sin(\theta_3) - 4I_2 l_1^2 l_3^2 m_3^2 \sin(2\theta_1 - 2\theta_3) - 4I_1 l_2^2 l_3^2 m_3^2 \sin(2\theta_2 - 2\theta_3) + 8I_2 g l_1^2 l_3 m_3^2 \sin(2\theta_1 - \theta_3) + 8I_1 g l_2^2 l_3 m_3^2 \sin(2\theta_2 - \theta_3) + 2k_3 l_1^2 l_2^2 m_1 m_2 \theta_3 + 8k_3 l_1^2 l_2^2 m_1 m_3 \theta_3 + 24k_3 l_1^2 l_2^2 m_2 m_3 \theta_3 - 4k_2 l_2 l_3 m_3 \dot{\theta}_2 (\cos(\theta_2 - \theta_3)((m_1 + 3m_2 + 2m_3)l_1^2 + 4I_1) - l_1^2 \cos(2\theta_1 - \theta_2 - \theta_3)(m_2 + 2m_3)) - 8I_1 g l_2^2 l_3 m_3^2 \sin(\theta_3) - 8I_2 g l_1^2 l_3 m_3^2 \sin(\theta_3) - 4k_3 l_1^2 l_2^2 m_2^2 \theta_3 \cos(2\theta_1 - 2\theta_2) - 16k_3 l_1^2 l_2^2 m_3^2 \theta_3 \cos(2\theta_1 - 2\theta_2) - 8I_2 g l_1^2 l_3 m_2 m_3 \sin(\theta_3) - 16k_3 l_1^2 l_2^2 m_2 m_3 \theta_3 \cos(2\theta - 2\theta_2) - l_1^2 l_2^2 l_3^2 m_1 m_3^2 \dot{\theta}_3^2 \sin(2\theta_2 - 2\theta_3) + l_1^2 l_2^2 l_3^2 m_2 m_3^2 \dot{\theta}_3^2 \sin(2\theta_1 - 2\theta_3) - 2l_1^2 l_2^2 l_3^2 m_2 m_3^2 \dot{\theta}_3^2 \sin(2\theta_2 2\theta_3) + 2g l_1^2 l_2^2 l_3 m_1 m_3^2 \sin(2\theta_2 - \theta_3) - g l_1^2 l_2^2 l_3 m_2 m_3^2 \sin(2\theta_1 - \theta_3) + 2g l_1^2 l_2^2 l_3 m_2 m_3^2 \sin(2\theta_2 - \theta_3) + g l_1^2 l_2^2 l_3 m_2 m_3 \sin(2\theta_2 - \theta_3) + 4I_2 g l_1^2 l_3 m_1 m_3 \sin(2\theta_1 - \theta_3) + 8I_2 g l_1^2 l_3 m_2 m_3 \sin(2\theta_1 - \theta_3) + 4I_1 g l_2^2 l_3 m_2 m_3 \sin(2\theta_2 - \theta_3) - 2g l_1^2 l_2^2 l_3 m_1 m_3^2 \sin(2\theta_1 - \theta_2 + \theta_3) - 2g l_1^2 l_2^2 l_3 m_2 m_3^2 \sin(2\theta_1 - 2\theta_2 + \theta_3) - g l_1^2 l_2^2 l_3 m_2 m_3 \sin(2\theta_1 -$$

$$\begin{aligned}
& 2\theta_2 + \theta_3) + g_1^2 l_2^2 l_3 m_2^2 m_3 \sin(\theta_3) - g_1^2 l_2^2 l_3 m_1 m_2 m_3 \sin(2\theta_1 - \\
& 2\theta_2 + \theta_3)) / (64 I_1 I_2 I_3 + 8 I_3 l_1^2 l_2^2 m_2^2 + 8 I_1 l_2^2 l_3^2 m_3^2 + 8 I_2 l_1^2 l_3^2 m_3^2 + \\
& 32 I_3 l_1^2 l_2^2 m_3^2 + 16 I_2 I_3 l_1^2 m_1 + 16 I_1 I_3 l_2^2 m_2 + 64 I_2 I_3 l_1^2 m_2 + \\
& 16 I_1 I_2 l_3^2 m_3 + 64 I_1 I_3 l_2^2 m_3 + 64 I_2 I_3 l_1^2 m_3 + 4 I_3 l_1^2 l_2^2 m_1 m_2 + \\
& 4 I_2 l_1^2 l_3^2 m_1 m_3 + 16 I_3 l_1^2 l_2^2 m_1 m_3 + 4 I_1 l_2^2 l_3^2 m_2 m_3 + \\
& 16 I_2 l_1^2 l_3^2 m_2 m_3 + 48 I_3 l_1^2 l_2^2 m_2 m_3 - 8 I_1 l_2^2 l_3^2 m_3^2 \cos(2\theta_2 - \\
& 2\theta_3) - 2 l_1^2 l_2^2 \cos(2\theta_1 - 2\theta_2)(m_2 + 2m_3)(m_2 m_3 l_3^2 + 4 I_3(m_2 + \\
& 2m_3)) - 2 l_1^2 l_3^2 m_3^2 \cos(2\theta_1 - 2\theta_3)(4 I_2 - m_2 l_2^2) + 2 l_1^2 l_2^2 l_3^2 m_1 m_3^2 + \\
& 6 l_1^2 l_2^2 l_3^2 m_2 m_3^2 + 2 l_1^2 l_2^2 l_3^2 m_2 m_3 + l_1^2 l_2^2 l_3^2 m_1 m_2 m_3 - \\
& 2 l_1^2 l_2^2 l_3^2 m_1 m_3 \cos(2\theta_2 - 2\theta_3) - 4 l_1^2 l_2^2 l_3^2 m_2 m_3 \cos(2\theta_2 - 2\theta_3))
\end{aligned}$$

REFERENCES

- [1] M. C. Kasapbai, "A new chaotic image steganography technique based on huffman compression of turkish texts and fractal encryption with post-quantum security," *IEEE Access*, vol. 7, pp. 148 495–148 510, 2019.
- [2] X. Wu, J. Li, and G. Chen, "Chaos in the fractional order unified system and its synchronization," *Journal of the Franklin Institute*, vol. 345, no. 4, pp. 392–401, 2008.
- [3] E. E. May, M. A. Vouk, D. L. Bitzer, and D. I. Rosnick, "An error-correcting code framework for genetic sequence analysis," *Journal of the Franklin Institute*, vol. 341, no. 1-2, pp. 89–109, 2004.
- [4] R. Schmitz, "Use of chaotic dynamical systems in cryptography," *Journal of the Franklin Institute*, vol. 338, no. 4, pp. 429–441, 2001.
- [5] T. Yang, "A survey of chaotic secure communication systems," *International journal of computational cognition*, vol. 2, no. 2, pp. 81–130, 2004.
- [6] L. Kocarev, "Chaos-based cryptography: a brief overview," *IEEE Circuits and Systems Magazine*, vol. 1, no. 3, pp. 6–21, 2001.
- [7] T. Habutsu, Y. Nishio, I. Sasase, and S. Mori, "A secret key cryptosystem by iterating a chaotic map," in *Workshop on the Theory and Application of Cryptographic Techniques*. Springer, 1991, pp. 127–140.
- [8] N. Masuda and K. Aihara, "Cryptosystems with discretized chaotic maps," *Ieee transactions on circuits and systems i: fundamental theory and applications*, vol. 49, no. 1, pp. 28–40, 2002.
- [9] H. Zhou and X.-T. Ling, "Problems with the chaotic inverse system encryption approach," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 44, no. 3, pp. 268–271, 1997.
- [10] F. Huang and Z.-H. Guan, "A modified method of a class of recently presented cryptosystems," *Chaos, Solitons & Fractals*, vol. 23, no. 5, pp. 1893–1899, 2005.
- [11] N. K. Pareek, V. Patidar, and K. Sud, "Discrete chaotic cryptography using external key," *Physics Letters A*, vol. 309, no. 1-2, pp. 75–82, 2003.
- [12] L. Kocarev and G. Jakimoski, "Logistic map as a block encryption algorithm," *Physics Letters A*, vol. 289, no. 4-5, pp. 199–206, 2001.
- [13] S. Vaidyanathan, A. T. Azar, K. Rajagopal, A. Sambas, S. Kacar, and U. Cavusoglu, "A new hyperchaotic temperature fluctuations model, its circuit simulation, fpga implementation and an application to image encryption," *International Journal of Simulation and Process Modelling*, vol. 13, no. 3, pp. 281–296, 2018.
- [14] S. S. Askar, A. A. Karawia, and A. Alshamrani, "Image encryption algorithm based on chaotic economic model," *Mathematical Problems in Engineering*, vol. 2015, p. 341729, Jan 2015. [Online]. Available: <https://doi.org/10.1155/2015/341729>
- [15] A. Akhavan, H. Mahmodi, and A. Akhshani, "A new image encryption algorithm based on one-dimensional polynomial chaotic maps," in *International Symposium on Computer and Information Sciences*. Springer, 2006, pp. 963–971.
- [16] T. Habutsu, Y. Nishio, I. Sasase, and S. Mori, "A secret key cryptosystem by iterating a chaotic map," in *Workshop on the Theory and Application of Cryptographic Techniques*. Springer, 1991, pp. 127–140.
- [17] Z. Kotulski and J. Szczepański, "Discrete chaotic cryptography," *Annalen der Physik*, vol. 509, no. 5, pp. 381–394, 1997.
- [18] Z. Kotulski, J. Szczepański, K. Górski, A. Paszkiewicz, and A. Zugaaj, "Application of discrete chaotic dynamical systems in cryptographydccc method," *International Journal of Bifurcation and Chaos*, vol. 9, no. 06, pp. 1121–1135, 1999.
- [19] J. Fridrich, "Symmetric ciphers based on two-dimensional chaotic maps," *International Journal of Bifurcation and chaos*, vol. 8, no. 06, pp. 1259–1284, 1998.
- [20] L. M. Ward and R. L. West, "Modeling human chaotic behavior: Nonlinear forecasting analysis of logistic iteration," *Nonlinear Dynamics, Psychology, and Life Sciences*, vol. 2, no. 4, pp. 261–282, 1998.
- [21] Y.-H. Chu and S. Chang, "Dynamical cryptography based on synchronized chaotic systems," *Electronics Letters*, vol. 35, no. 12, pp. 974–975, 1999.
- [22] E. Alvarez, A. Fernandez, P. Garcia, J. Jiménez, and A. Marcano, "New approach to chaotic encryption," *Physics Letters A*, vol. 263, no. 4-6, pp. 373–375, 1999.
- [23] A. Sambas, S. Vaidyanathan, E. Tlelo-Cuautle, B. Abd-El-Atty, A. A. A. El-Latif, O. Guillen-Fernandez, Sukono, Y. Hidayat, and G. Gundara, "A 3-d multi-stable system with a peanut-shaped equilibrium curve: Circuit design, fpga realization, and an application to image encryption," *IEEE Access*, vol. 8, pp. 137 116–137 132, 2020.
- [24] A. Sambas, S. Vaidyanathan, E. Tlelo-Cuautle, S. Zhang, O. Guillen-Fernandez, Sukono, Y. Hidayat, and G. Gundara, "A novel chaotic system with two circles of equilibrium points: Multistability, electronic circuit and fpga realization," *Electronics*, vol. 8, no. 11, 2019. [Online]. Available: <https://www.mdpi.com/2079-9292/8/11/1211>
- [25] "Matlab scientific programming tool," 2019. [Online]. Available: <https://in.mathworks.com/products/matlab.html>
- [26] K. Ramasubramanian and M. Sriram, "A comparative study of computation of lyapunov spectra with different algorithms," *Physica D: Nonlinear Phenomena*, vol. 139, no. 1-2, pp. 72–86, 2000.
- [27] K. Roy, "A survey on post-quantum cryptography for constrained devices," *International Journal of Applied Engineering Research*, vol. 14, pp. 2608–2615, 05 2019.
- [28] D. Bernstein, J. Buchmann, and E. Dahmen, "Post-quantum cryptography springer-verlag," *Berlin Heidelberg*, 2009.
- [29] A. Pérez-Res, M. Garcia-Bosque, C. Sánchez-Azqueta, and S. Celma, "Chaotic encryption applied to optical ethernet in industrial control systems," *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 12, pp. 4876–4886, 2019.
- [30] —, "Physical layer encryption for industrial ethernet in gigabit optical links," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 4, pp. 3287–3295, 2019.
- [31] R. Agbeyibor, J. Butts, M. Grimaila, and R. Mills, "Evaluation of format-preserving encryption algorithms for critical infrastructure protection," in *International Conference on Critical Infrastructure Protection*. Springer, 2014, pp. 245–261.
- [32] "Xilinx vivado eda tool," 2012. [Online]. Available: <https://bit.ly/2MpTxeP>
- [33] Z. Hua, B. Zhou, and Y. Zhou, "Sine chaotification model for enhancing chaos and its hardware implementation," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 2, pp. 1273–1284, 2019.
- [34] S. Kalanadhabhatta, D. Kumar, K. K. Anumandla, S. A. Reddy, and A. Acharyya, "Puf-based secure chaotic random number generator design methodology," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 7, pp. 1740–1744, 2020.
- [35] "A statistical test suite for random and pseudorandom number generators for cryptographic applications," April 2010. [Online]. Available: <https://doi.org/10.6028/NIST.SP.800-22r1a>
- [36] "The marsaglia random number cdrom including the diehard battery of tests of randomness," 1995. [Online]. Available: <http://stat.fsu.edu/pub/diehard/>