# Design & Implementation of Quantum Computing Immune Cryptography Processor

A thesis submitted in partial fulfillment of
the requirements for the degree of

**Bachelor of Technology**

by

**Abhishek Agrawal**
**(150102002)**
**Souradip Pal**
**(150102076)**

Under the guidance of
**Dr. Gaurav Trivedi**


Department of Electronics & Electrical Engineering
INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI
April 2019

# Abstract

Recent advancements in the domain of Quantum Computing are posing a security threat to most of the classical cryptographic methods and algorithms. Most popular symmetric and asymmetric cryptosystems including RSA, ECC, DES, Diffie-Hellman etc. can be efficiently broken by a quantum computer running Shor's Algorithm. There exist two methods to counter this threat: firstly using **Quantum Cryptography** which involves setting up a quantum channel, quantum computers and would require a complete overhaul of the data communication infrastructure. Second alternative called **Post-Quantum Cryptography**, can however be implemented on the existing infrastructure and is equally resistant to classical and quantum crypto-attacks. Post-Quantum Cryptographic algorithms are more immune to these attacks because they are designed on the weakness of quantum circuits and algorithms. Hash, Code, Lattice based and Multivariate Polynomial based cryptographic algorithms are the most popular class of such algorithms. *In this report, we present the design and hardware implementation of a chaos-based post-quantum cryptographic system derived from a mechanical model showing non-linear dynamics and show its resistance against quantum computing attacks.* We also provide in this report the details of the encryption-decryption algorithms and experimentations performed on the implementation of the system in *Artix FPGA* hardware platform.

# Contents

# List of Figures

# Nomenclature

| | |
|---|---|
| FPGA | Field Programmable Gate Array |
| DSA | Digital Signature Algorithm |
| RSA | Rivest-Shamir-Adleman |
| DES | Data Encryption Standard |
| PQC | Post-Quantum Cryptography |
| ECC | Elliptic Curve Cryptography |
| NTRU | Nth-Degree Truncated Polynomial Ring |
| CLB | Configurable Logic Block |
| ALU | Arithmetic Logic Unit |
| LUT | Look-Up Table |
| USB | Universal Serial Bus |

# Chapter 1

# Introduction

## 1.1 Motivation

In this modern era, while transferring data from one computer to another, we almost everytime assume a secure connection. However, in a situation where this security is broken, the effects can be devastating and almost all the businesses and services relying on internet including banking and cloud services would be rendered useless. Such a situation may not be too distant in future. Advancements in quantum computing have posed a threat to the existing cryptographic methods on which the complete data communication infrastructure relies on, including the most popular public-key cryptography.

In 1994, a mathematician named Peter Shor, developed an algorithm which is able to break the security of key exchanges and digital signatures. Using this algorithm, a quantum computer would be able to crack the most sophisticated encryption in a matter of minutes. Quantum computers operate differently from traditional computers i.e. they work at atomic level. The essential units of a quantum computer is a qubit in contrast to bits used in traditional computers. A qubit is able to represent 0 and 1 simultaneously. Due to this property, few qubits can speed up certain types of computation by an enormous amount and hence quantum computers are more suitable for brute-force exhaustive searches. Although, this new technology is ideal for solving complex problems in astrophysics, pharmaceuticals and weather forecasting, it can also break the encryption and endanger our privacy and security.

## 1.2 Problem Definition

The objective of this project is to design and implement a chaos-based cryptographic system on hardware that is secure against Shor's Algorithm running on an ideal quantum computer.

This problem focuses on going beyond traditional cryptographic methods and implementing a new chaos based encryption-decryption technique in dedicated hardware like Field Programmable Gate Array (FPGA). Our aim is to design the system in such a way that it is simple enough to be implemented in practice, it is computationally efficient and it provides a reasonable degree of security. The system must also possess a number of fundamental features which are important for any cryptosystem in general. In order to achieve this goal some necessary steps to be performed along with the design and implementation are:

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

# Chapter 2

# Preliminaries

## 2.1 Present-day Cryptography

Cryptography is the practice and study of techniques for secure data transfer over insecure channels in presence of unauthorized users, usually called adversaries. Prior to modern age, cryptography was most exclusively referred to as encryption, i.e. the process of converting data from readable state to apparent noise.

Most of the present-day encryption algorithms are based on overlapping theory of mathematics and computer science. These are largely designed around a 'trap-door functions', i.e. problems with sufficient computational hardness. These problems can be theoretically solved, but this is not possible to do within reasonable time and with the resources that usually at disposal.

Advances in mathematics such as improved algorithms on the integer factorization problem and discrete logarithm problem and availability of more computational power require these methods to adapt with time. Most of the algorithms can, however, be made secure against these advancements just by increasing the key-length.

## 2.2 Shor's Algorithm

Published in 1995 and named after it's formulator, Shor's algorithm, is a quantum algorithm for finding the prime factors of any given integer N.

The importance of Shor's algorithm lies in it's ability to find the prime factors of an integer in polynomial time, in comparison to the most efficient classical algorithms known, such as general number field sieve, which works in sub-exponential time. This is possible to due to efficiency of quantum Fourier transform and modular exponentiation by repeated squaring.

## 2.3 Shor's Algorithm & Present Encryption Methods

All of the popular cryptographic algorithms rely on one of the three hard problems - integer-factorization problem, the discrete logarithm problem or the elliptic-curve discrete logarithm problem. These problems were viewed as important because of the difficulty of factoring large numbers is relied upon for many cryptography systems. If an efficient method of factoring large numbers is implemented, most of the encryption schemes would be next to worthless to protect their data.

With Shor's algorithm, these can be be solved in reasonable polynomial time. This implies that, with a quantum computer having sufficient number of qubits, Shor's algorithm can be used to break the public-key cryptography schemes including RSA.However, the experimental quantum computers available today succumb to noise and decoherence problems, hence, cannot be used to break current encryption schemes.

## 2.4 Post-Quantum Cryptography

Post-Quantum Cryptography refers to the study of classical encryption schemes that are considered to be secure against an attack by quantum computers. As discussed above, most of the popular encryption methods can be efficiently broken using a sufficiently powerful hypothetical quantum computer. Thus, scientists are preparing methods to secure data against attacks when quantum computing becomes much more powerful.

Daniel J. Bernstein[11] lists the following important classes of cryptographic methods beyond RSA, DSA and ECRSA -

**Hash-based -** This includes Merkle's hash-tree public-key signature system, building upon the one-message idea.

**Code-based -** An example is McEliece's hidden-Goppa-code public-key encryption system.

**Lattice-based -** The most popular example that garnered most interest is Hoffstein-Pipher-Silverman NTRU public-key-encryption system.

**Multivariate quadratic equation based-** One of the examples is Patarin's *HFEv* public-key-signature system originally proposed by Matsumoto and Imai.

**Secret-key based -** The leading example is the DaemenRijmen *Rijndael* cipher which was renamed AES(Advanced Encryption Standard).

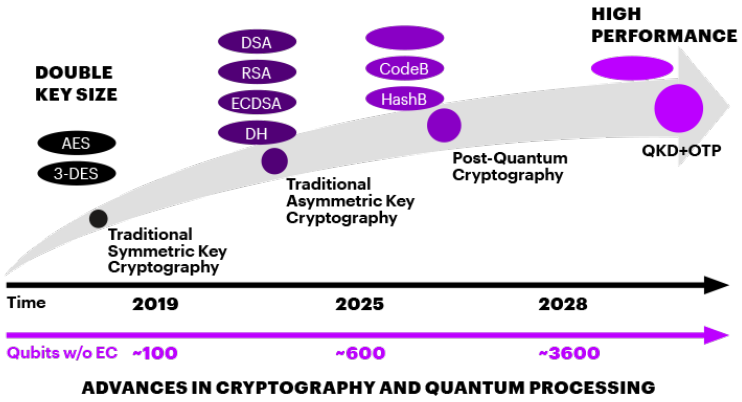
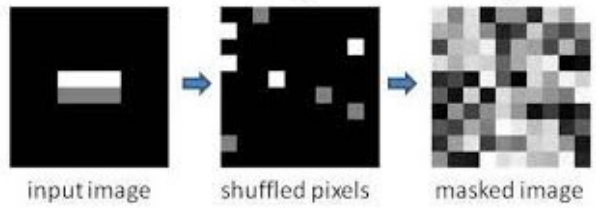



Figure 2.1: Timeline for Quantum Cryptography

## 2.5 Chaos based Cryptography

Apart from the popular post-quantum cryptographic methods, a new method of constructing cryptosystems utilising the nonpredictability property of discrete chaotic systems has become somewhat noteworthy from practical perspectives. This type of systems are based on the characteristics of chaos, which are sensitivity of parameters, sensitivity of initial points, and randomness of sequences obtained by iterating a chaotic map. A ciphertext is obtained by the iteration of a inverse chaotic map from an initial point, which denotes a plaintext.

If the times of the iteration is large enough, the randomness of the encryption and the decryption functions are so large that attackers are unable to break this cryptosystem by statistical characteristics. Hence, nonlinear dynamical systems with chaos seem to be good candidates for constructing such encryption-decryption algorithms. Most of these methods have been used in the development of symmetric ciphers for encryption of 2D images.




(a) Image Encryption using 1D Chaotic Map


(b) Chaos based Image Encryption

Figure 2.2: Chaos based Cryptography

## 2.6 Field Programmable Gate Array

A field programmable gate array is a set (array) of reconfigurable gates that can implement both sequential and combinational logic including multi-level logic functions.

FPGAs are built in form of an array of configurable logic blocks (CLBs). Each of these CLBs can be programmed to perform a logic function and then connected to each other through a hierarchy of interconnects (routing channels) to form a complex logic. FPGA also has a set of I/O elements for interfacing with external devices such as flash memories and network ports.

The structure of CLBs varies with model and manufacturers, but all of them share a lot of similarity. Figure 2.1 shows the structure of Cyclone IV CLB, consisting of a 4-input LUT and a flip-flop. By loading appropriate values, the LUT can be programmed to perform any 4-input function. Further, the choice of multiplexer select signals determines how the data is routed through LUT to the neighbouring LEs and IOEs. The LUT output either goes directly to the LE output for combinational logic, or it can be routed through the register for sequential logic.

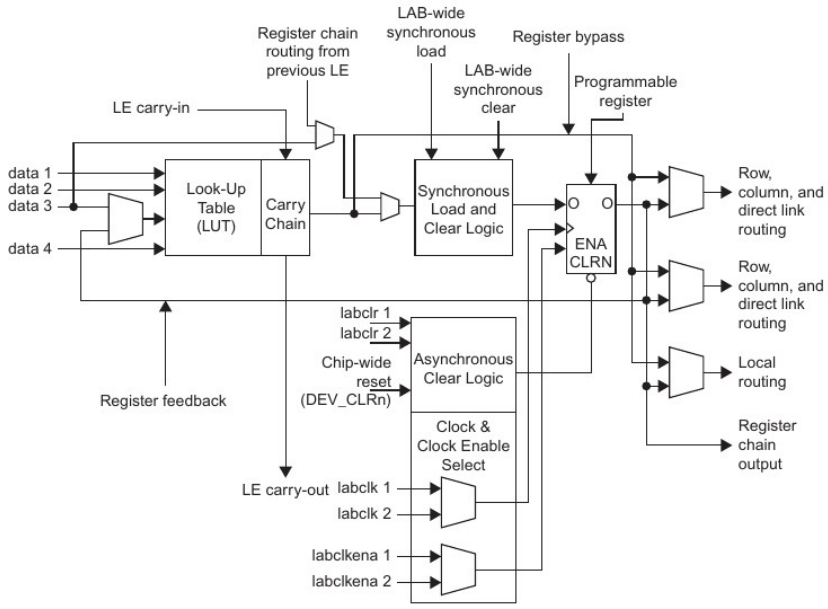



Figure 2.3: FPGA CLB Structure

An engineer programs the logic using Hardware Description Language (HDLs) such as VHDL or Verilog. A synthesis is then used to generate a gate-level netlist. An implementation tool then determines how the CLBs and routing channels should be configured to perform the specified function and generates a bitstream that can be loaded into a FPGA.

# Chapter 3

# Literature Survey

## 3.1 Chaos and Cryptography

Over last few years, there has been a great interest in understanding the working of chaotic systems. They are distinguished by their high sensitivity to initial conditions, statistically similar to random signals and a continuous broad-band power spectrum. This has garnered interest from cryptanalysts and there have been several publications proposing various chaos-based cryptographic systems such as in [2], [3].

The chaos-based cryptosystems can be sub-divided into two classes. First one involves numerically computing a large number of iterations over time of a chaotic system, using message as the initial data. (see [5], [6]). The second class amounts to scrambling a message with a chaotic dynamic. This includes additive masking, chaotic switching, message embedding, etc.

The relevance and usefulness of chaos in these systems have been demonstrated through comparative studies between characteristics of chaotic systems and requirements of a strong cipher (see [1], [4]). Several properties of chaotic maps are similar to those of cryptographic maps : extreme sensitivity to initial conditions and parameters, unstable periodic orbits with large time-periods. Further, iterations of chaotic map spread the initial region over entire map, introducing diffusion which is an important requirement for a strong cipher.

## 3.2 Discrete Chaos

It must be noted that when chaotic systems are simulated on computers with limited precision, the sequences $x_k$ generated are not exactly chaotic. Since, the cardinality of this set is finite, such sequences will always be a part of a loop of finite period. It can be expected that this period wouldn't be too short and will be greatly chaotic in nature.

Claiming such properties, however, requires some consideration [7]. Contributions made in this regard and discussion about discrete chaos can be found in [8]. However, some noteworthy takeaways are listed below -

- Through numerical experiments, it has been shown that mean cycle L of such a system is $O(2^{P/2})$, where P is the amount of precision in terms of number of bits. This serves as good reference while working with chaotic systems. However, it must be verified as there are no mathematical proofs to support it.
- The rounding error in computer systems poses another problem. The errors made in each iteration will culminate at a very fast rate due to high sensitivity of the system on the initial conditions. Thus, the actual trajectory and the calculated trajectory will be considerably different after a few iterations. However, "Shadowing Lemma" in [10], guarantees that one can always find an actual trajectory that is arbitrarily near the calculated trajectory.

## 3.3 Chaos in Non-Linear Dynamic Systems

Many real world phenomena can be mathematically modelled as non-linear dynamic systems. Out of these phenomena, some exhibit significant degree of chaos. The unpredictability of these non-linear phenomena is due to the fact that the system passes through a series of *unstable states*. Also, these non-linear systems generally display very sensitive dependence on initial conditions which is the main reason for generating chaotic maps using non-linear dynamic systems. It must, however, be noted that not every complicated dynamic behavior can be considered chaotic. Chaotic systems differ from *noisy motion* in that their randomness is due to interaction of few simple laws. The quantitive description, however, lies in the concept of Lyapunov exponents which measures the exponential divergence of trajectories of the chaotic maps.

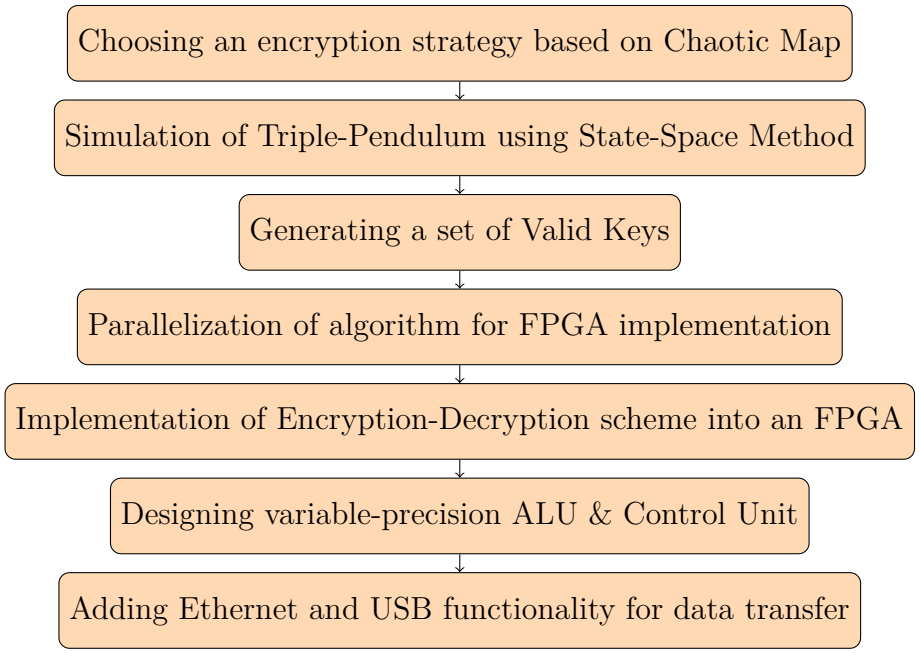
## 3.4 Baptista-type Cryptosystem

Proposed by M.S. Baptista, this is a chaotic-cryptosystem based on the interval partitioning of chaotic orbits of the logistic map. This uses the ergodic property of chaos, which enables the construction of fast and secure chaotic encryption-decryption schemes due to its simplicity and less complex structure. The main idea of this scheme is to first map the text characters to real values and then algorithms are applied to encrypt the message. Decryption is performed by iterating the chaotic map and then corresponding symbol for the real values are obtained.

# Chapter 4

# Proposed Approach

The proposed algorithm of encryption and decryption is based on multiple iterations of a certain dynamical chaotic system and using a Baptista-type encryption-decryption scheme from the chaotic map generated. After referring extensive literature and keeping in mind the resources at our disposal, we have decided to take the compound triple-pendulum model for creating the chaotic map in our cryptosystem. We assume that input to the system is a plain message. The system parameter(s) and additionally the initial conditions of the dynamical system are assumed to be the part of the secret key. After simulating the triple-pendulum, our next step would be to generate a valid set of keys for encryption and decryption. The entire mechanical model would then be implemented on *Artix FPGA* including the design and implementation of ALU and Control Unit, adhering to the performance specification of the problem. Additionally, an extra data transfer module can be added to the hardware for key exchanges.
The entire approach can be summarized using the following roadmap :-



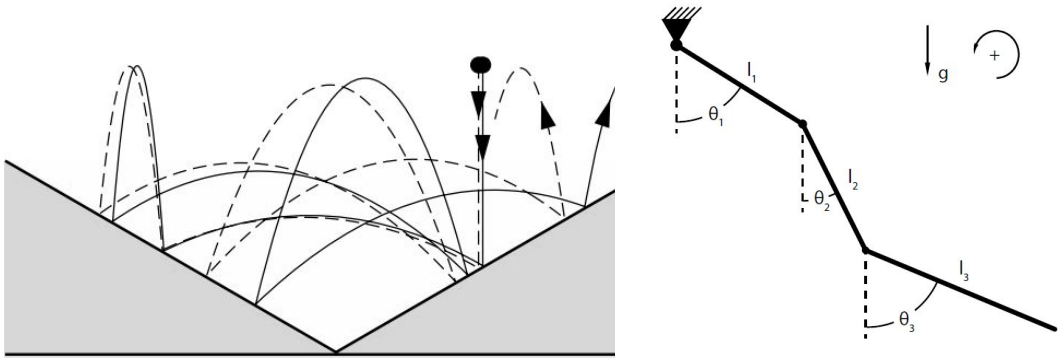
# Chapter 5

# Work Progress

## 5.1 Verifying the Algorithm in MATLAB

### 5.1.1 Model of Non-Linear Dynamic System

A schematic diagram of the compound triple-pendulum system is shown in Figure 5.1.The bars of the pendulum have significant mass so that it can be modeled as a compound pendulum. The model has been parameterized according to the physical characteristics of the system including mass of the bars, their inertia etc. The position and velocity of the bars are defined by the six system state variables: $\theta_1$, $\theta_2$, $\theta_3$, $\dot{\theta_1}$, $\dot{\theta_2}$, $\dot{\theta_3}$




(a) Chaotic motion of a bouncing ball

(b) Schematic Diagram of Triple Pendulum

Figure 5.1: Examples of Non-Linear Dynamic systems

### 5.1.2 Simulation of Compound Triple Pendulum

This compound triple-pendulum model has been simulated using MATLAB using approximate differential equations describing the random motions. The parameters and initial conditions of the ODEs are given in Table 1 and Table 2 respectively. For the simulation, simple numerical methods were used to solve the differential equations and the values

| Parameter | Value | Unit |
|---|---|---|
| $m_1$ | 0.2944 | kg |
| $m_2$ | 0.1756 | kg |
| $m_3$ | 0.0947 | kg |
| $l_1$ | 0.508 | m |
| $l_2$ | 0.254 | m |
| $l_3$ | 0.127 | m |
| $I_1$ | 9.526e-3 | kg· m^2 |
| $I_2$ | 1.625e-3 | kg· m^2 |
| $I_3$ | 1.848e-4 | kg· m^2 |
| $k_1$ | 5e-3 | N·m·s/rad |
| $k_2$ | 0 | N·m·s/rad |
| $k_3$ | 8e-4 | N·m·s/rad |

(a) Table 1

| Condition | Value | Unit |
|---|---|---|
| $\theta_1$ | -0.4603 | rad |
| $\theta_2$ | -1.2051 | rad |
| $\theta_3$ | -1.5165 | rad |
| $\dot{theta}_n$ | 0 | rad/s |

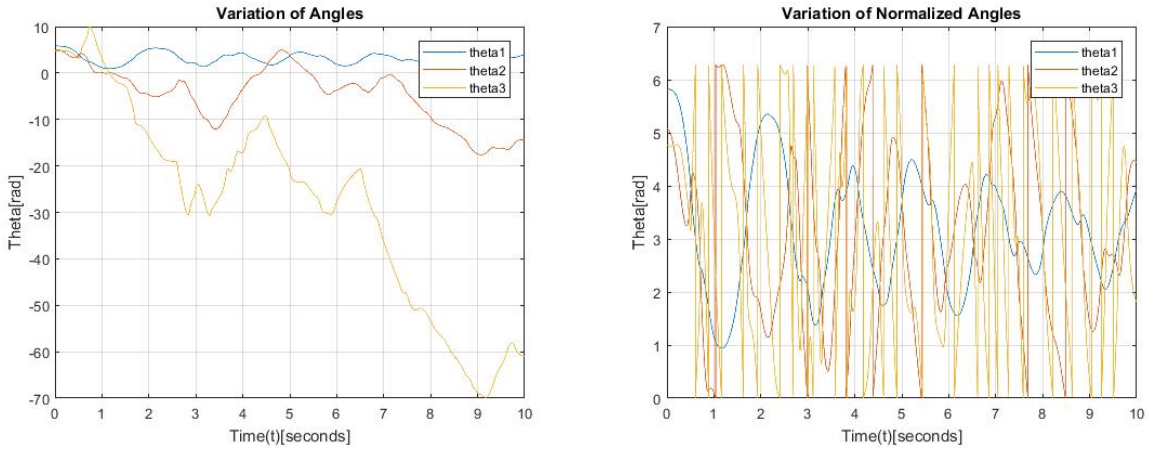
(b) Table 2

Figure 5.2: Parameters & Initial Conditions for the Initial Value Problem

corresponding to the angular position of the bars were obtained within a certain duration of time with a predefined precision. This generates the mapping values for the encryption module.

**Simulation Results**

These are some of the observations from the simulation of the triple-pendulum model for a duration of t = 0 to t = 10 seconds with $\Delta$t = 0.001 :
(i) Initial conditions are same as given in Table 1 & 2:




(a) Plot of $\theta$ vs Time

(b) Plot of Normalized $\theta$ vs Time

Figure 5.3: Motion of Triple-Pendulum for t = 0 to t = 10 sec

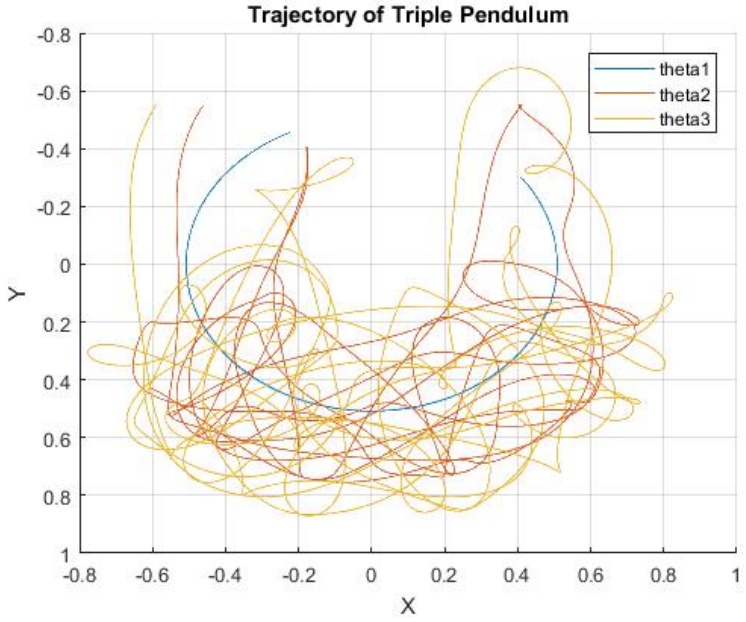



Figure 5.4: Plot of Motion of Triple-Pendulum

### 5.1.3 Encryption - Decryption

Our approach is to convert the plain-text into ascii format and map the value to the partitioned intervals of the triple-pendulum motion simulated within a specific duration of time for a particular set of parameters and initial conditions. The initial conditions and parameters of the different equation forms a part of the private key. Following the Baptista-type method, the entire range of the chaotic function was partitioned into a number of intervals equal to the number of characters. Each character in the plain-text is then mapped to a specific interval and then to a time point randomly selecting from that interval.

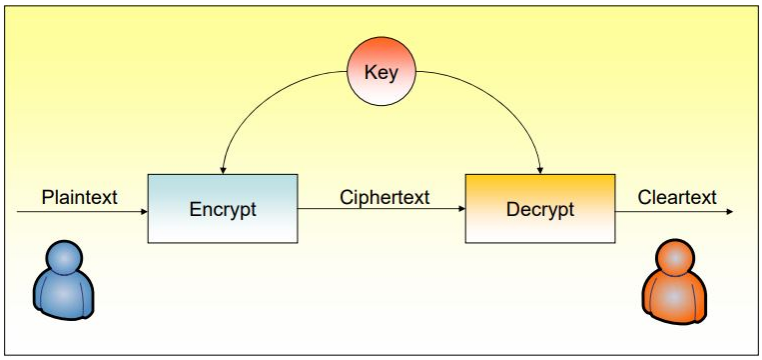



Figure 5.5: Working Principle of a Symmetric Cryptosystem

On the decryption module, the interval in which the encrypted value lies is computed from the generated motion of the triple-pendulum for the same key and the corresponding

index would then refer to the ascii converted clear-text. Converting them into characters, the message can be decoded.

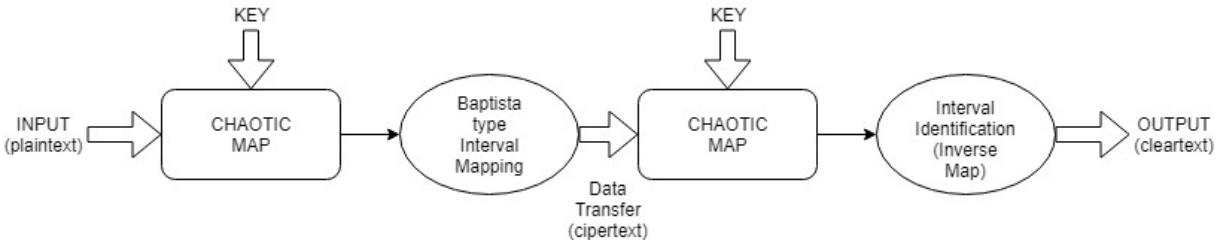



Figure 5.6: Encryption-Decryption Strategy

## 5.2 Key Generation

It is observed that for certain specific parameters or initial conditions, the motion of the bars of the triple-pendulum shows periodic nature after a certain span of time. Hence there is a need to eliminate those parameters or initial conditions for which the motion is periodic as the periodic nature breaks the chaotic behavior of the system. For that a test for periodicity was employed to extract the prominent period of the signal using statistical analysis on the spectrum of the signal. The test used is known as ***Fisher's g-statistic test***(see [11]). This method is based on the test of significance of the periodic components of the signal derived from its periodogram.

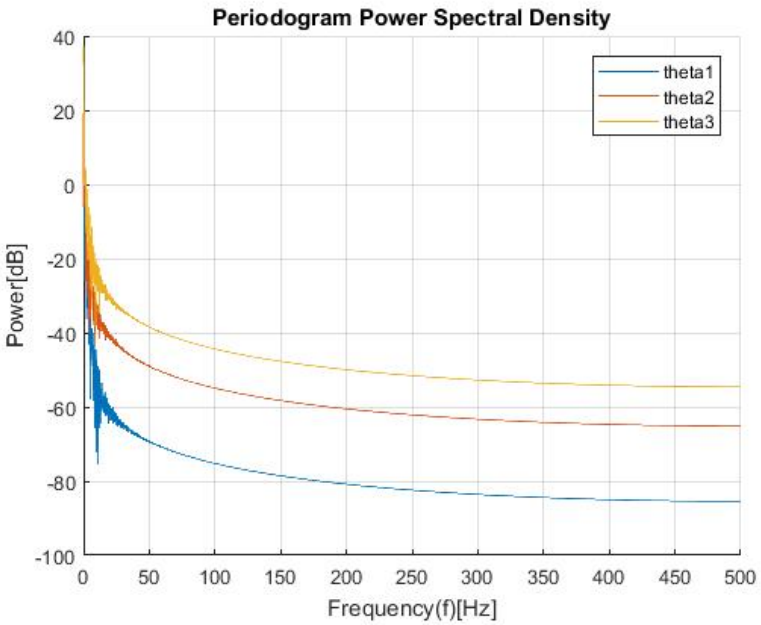
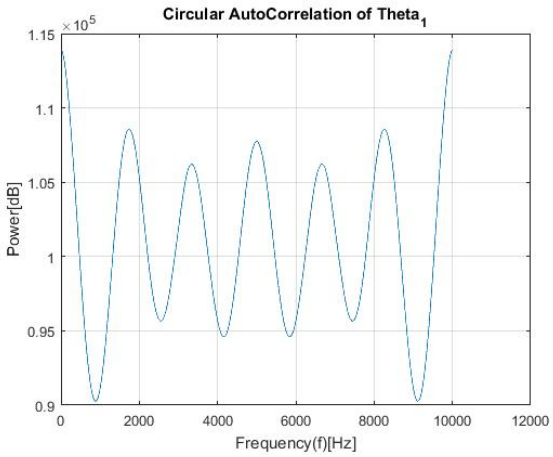
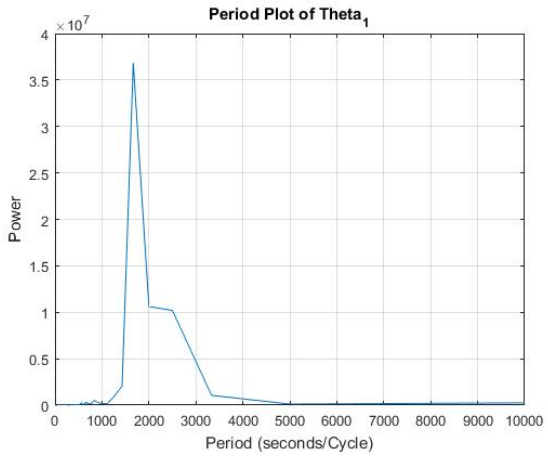



Figure 5.7: Periodogram Plot for $\theta$

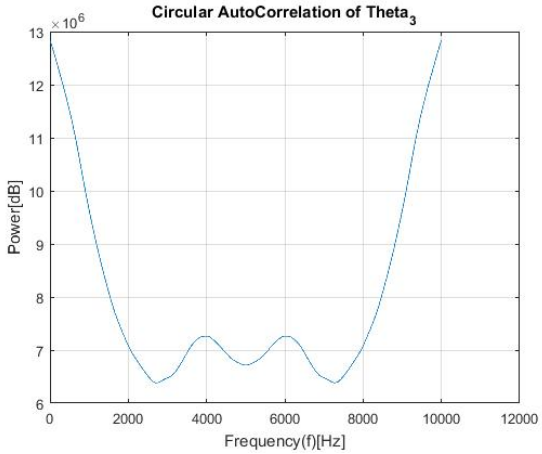



(a) Plot of Circular Auto-Correlation for $\theta_1$

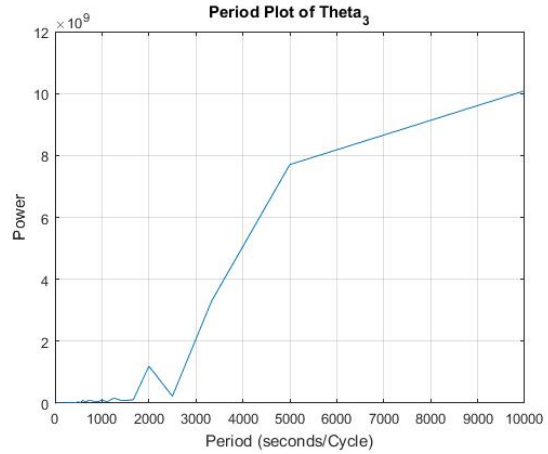



(b) Plot of Periodicity for $\theta_1$

Figure 5.8: Periodic Properties of $\theta_1$




(a) Plot of Circular Auto-Correlation for $\theta_3$




(b) Plot of Periodicity for $\theta_3$

Figure 5.9: Periodic Properties of $\theta_3$

From the figures 5.8 & 5.9 , it is observed that there is peak at 1700 (seconds/Cycle) for $\theta_1$ whereas no such peak occurs for $\theta_3$. Thus $\theta_1$ is periodic in nature. The values obtained from the periodicity test and plots of circular auto-correlation clearly differentiates the parameters and initial values which leads to periodic nature of the motion and those which lead to non-periodic nature of the motion. Thus by iterating through all possibles values of the parameters and checking likewise for non-periodic nature, a set of keys was generated and stored.

## 5.3 FPGA Implementation

The complete design is being implemented at Register-Transfer Level (RTL) in **Verilog HDL** and the target device chosen is Digilent Nexys Board with **Xilinx Artix-7 FPGA**.

The following diagram shows the implementation plan for the design :

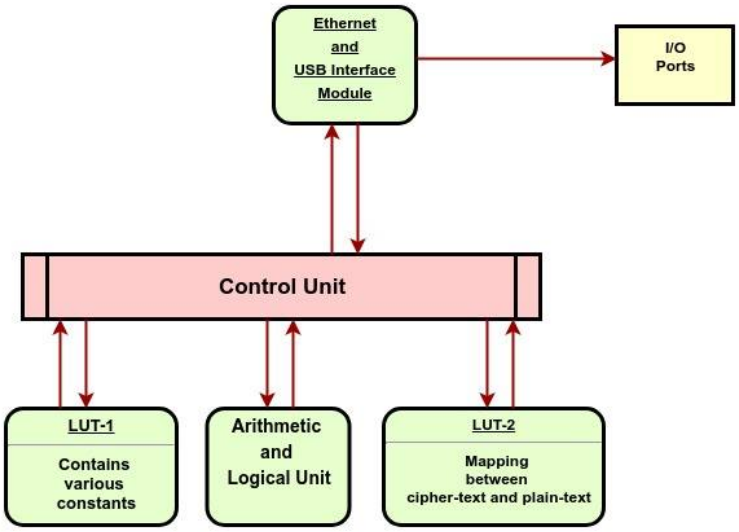



Figure 5.10: Block Diagram of FPGA Implementation

1. **Arithmetic and Logical Unit (ALU)** - The ALU uses floating-point arithmetic whose precision can be configured at the time of synthesis. This has been done to enable the study of how security strength varies with arithmetic precision.

2. **Look-Up Table (LUT)** - A fast Look-Up table is to be implemented for storing various constants and the mapping between clear-text and cipher-text.

3. **Ethernet and USB Module** - These are required to enable the device to encrypt or decrypt both a stream of data as well as complete file.

4. **Control Unit** - Essentially a state machine that co-ordinates the operation of all other modules including evaluation of state-variables and transfer of data through USB/Ethernet port.

## 5.4 Analysis

### 5.4.1 Test for Chaos

#### Collison Test

### 5.4.2 Test for Randomness

### 5.4.3 Complexity

# Chapter 6

# Future Work

This report so far discussed the methods for encryption-decryption based on non-linear chaotic map and utility in post-quantum cryptography.Through a number of simulations, we can conclude that this cryptosystem can provide the desired level of chaos and security. Moreover, this cryptosystem also fulfills the basic requirements defined by Shannon including diffusion and confusion.

Our next step would be generate a set of keys which can be employed for the purpose of encryption-decryption in practical scenarios. Although the implementation of basic hardware modules in FPGA are completed, various optimization of the algorithm is still in progress. We plan to measure the performance of the algorithm under various precision constraints and estimate the error rate. Further, we are required to perform necessary security analysis for the validation of this chaos-based cryptosystem.

# Bibliography


[1] A Connection Between Chaotic and Conventional Cryptography

[2] T. Yang, "A survey of chaotic secure communication systems", *Int. J. Comput. Cogn.*, 2004.

[3] M. Hasler, "Synchronization of chaotic systems and transmission of information", *Int. J. Bifurc. Chaos*, vol. 8, no. 4, Apr. 1998.

[4] Chaos-Based Cryptography: A Brief Overview by Ljupco Kocarev

[5] R. Schmitz, "Symmetric ciphers based on two-dimensional chaotic maps", *J. Franklin Inst.*, vol. 338, pp. 429-441, 2001

[6] D. E. Knuth, "The Art of Computer Programming. Reading", *MA: Addison-Wesley*, 1998, vol. 2.

[7] L. Kocarev, J. Szczepanski, J. M. Amigo, and I. Tomosvski, "L. Kocarev, J. Szczepanski, J. M. Amigo, and I. Tomosvski", *IEEE Trans. Circuits Syst. I, Reg. Papes,*, to be published.

[8] R. Matthews, D. Wheeler, "Supercomputer investigations of a chaotic encryption algorithm", *Cryptologia XV*, (1991) 140-152.

[9] E. Coven, I. Kan, J.A. Yorke, "Pseudo-orbit shadowing in the family of tent maps", *Amer. Math. Soc. 308 1*, (1988) 227-241.

[10] Sofia Wichert, Konstantinos Fokianos and Korbinian Strimmer, "Identifying periodically expressed transcripts in microarray time series data", *Bioinformatics. 20 1*, (2004) 5-20.

[11] Daniel J. Bernstein, Johannes Buchmann, Erik Dahmen, "Post-Quantum Cryptography"*Springer, Berlin, 2009.*