CHAPTER-1

INTRODUCTION

1.1 Need for Security

The present era of data explosion involves the necessity of high efficiency in terms of data capacity and data security [1]. As the data collection and processing capacity of the world as a whole is increasing at an exponential pace, there is a huge demand for attaining very high standards in terms of data throughput, processing capabilities and security, all at the same time [2, 3]. There is always a trade-off between security and capacity. However, if data capacity increases, high security is required to protect the data which imposes a Herculean demand on the signal processing, storage and communication equipment [4]. The most prominent beneficiaries in this era of Big Data are social networking sites and internet enabled services such as banking where privacy and security in data transmission is most crucial. Even though there are good protection measures currently, the data storage and transmission capacity offered by such systems is dismal in Big Data terms [5].

The typical examples that show there is a great need for security of big data are shown below:

- Gmail account hacking on September 11, 2014. Google, a very big multinational company with high security protection measures for data privacy, has succumbed to the hacking menace.
- The Snappening —a leak of approximately 90,000 photos and 9,000 videos stolen off the mobile app Snapchat. A UK Local Newspaper on October 13, 2014 confirmed that despite rumors of a hoax, the leak is genuine, and most of the affected users hail from Europe, which makes up 32 percent of its overall audience, according to Snapchat.

- Cars had been breached by a different group of scientists in 2011 through Bluetooth, GPRS and even car's media player CD is burned with a mischievous audio file. Self-driving cars will ascertain a tantalizing target for hackers if introduced in market, according to a top security executive.

1.2 Literature Survey of Secure Communication

Secure communication is a way of sharing information between two entities without any third party listening in. The key types of security are as follows [3]:

1. **The nature of a communication is hidden**

   Here we do not know the information content.

   Ex: Code, Encryption, Steganography etc.

2. **The parties to a communication are hidden – precluding identification, promoting anonymity**

   Here we don't know exactly the information content and the entities involved. For Example

   (i) "Crowds" and related anonymous groups – it is difficult to identify the source of information from a crowd.

   (ii) Unknown communication devices – fake cellphones, Internet centers.

   (iii) Unknown proxies.

3. **The fact that a communication takes place is hidden**.

   Here we do not know whether the communication has taken place or not. For Example

   (i) "Security by vagueness" – alike to needle in a haystack.

   (ii) Random traffic – forming indiscriminate data flow to make the occurrence of modest communication harder to identify and traffic investigation less reliable.

1.3 Role of Chaos in Secure Communication

The main reason to choose chaos in secure communications is because of its extreme sensitivity property [6]. The importance of chaos in secure communications has been explained by Chua and other people in the past and they have been successful in implementing it [6, 7].

The two aspects of chaos that are used as follows

1. Synchronization
2. Pseudorandom number

1.4 Objective

In this work we are trying to propose a novel kind of digital chaos which can be used in secure communications based on the above mentioned two aspects explained as follows [6, 7, 8].

1. **Synchronization**

   Most of the synchronized protocols are digital. In this work we used a simple circuitry to generate "Digital Chaos" from two square waves to be used as a futuristic clock/carrier. Following this we generate and characterize the digital chaos over various platforms like Application Specific Integrated Circuits (ASIC), Field Programmable Gate Arrays (FPGA), Microwind and MATLAB.

2. **Pseudorandom Number**

   We use a bit stream coming out of chaos which can be further used to generate a sequence of pseudorandom numbers. This sequence of pseudorandom numbers is validated using various statistical tests like Maurer's Universal Statistical Test, Discrete Fourier Transform Test, Non-Overlapping Template Matching Test, Runs Test, Rank Test and Monobit Test.

1.5 Project Flow

1. The basic principles of chaos and relavant analysis tools and techniques namely iterative map, cobweb plot, Largest Lyapunov Exponents (LLE), Fractal Dimension (D2), Kolmogorov entropy (K2), Phase Portrait and Recurrence Plot are used as standard parameters for characterization.
2. An iterative map representing digital chaos through the usage of square wave signals and XOR gate is formulated.
3. This iterative map is characterized using cobweb plots for different "r" values.
4. An ASIC based implementation of digital chaos generation is performed and standard characterization is done.
5. A layout level implementation of digital chaos is generated using 90nm CMOS technology is performed and standard characterization is done. The effect of wiring relating parasitics and associated delays on the nature of chaos generated is investigated.
6. In order to evaluate the tunability of the proposed digital chaos, an FPGA based implementation is carried out using Altera – DE1- Cyclone II FPGA.
7. Taking into account the recent development and prominence of software defined radio techniques, a purely software based implementation of digital chaos generation is done using MATLAB. The generated chaos and the nature of repetitions (or their absence) in the sequence is carried out using recurrence plots.
8. The MATLAB implementation of the chaos generator is used as the basis for Pseudo Random bit stream generation. The generated bit stream is tested for randomness using standard statistical tests like Histogram, Maurer's Universal Statistical Test, Non Overlapping Template Matching Test, Rank Test, Discrete Fourier Transform Test, Monobit Test and Runs Test.

CHAPTER-5

APPLICATION OF DIGITAL CHAOS- PSEUDORANDOM BIT GENERATION

5.1 Introduction

In the previous chapters we have generated and characterized the digital chaos. Now the immediate thought we got in our minds is what to do with that? As we already mentioned that we are motivated from the usage of chaos in communications, we wanted to make an application with Digital chaos that can be used in communications. In this modern era of life we know how communication is needed for mankind. The two important things that any communication system must have are capacity and security. Since the digital chaos inherently exhibits both properties, it can be used as the clock in communication system. Another application of digital chaos is pseudorandom bit generator (PBG) which can be used in cryptographic communications [26]. Cryptography is the art of converting the data into secure code for transmission over a public network. Pseudorandom number generation is essential to cryptography where we need an unpredictable sequence to convert the message data into encrypted code [26]. Digital chaos can be used to generate pseudorandom number [27].

5.2 Pseudo Random Bit Generator (PRBG) Using Digital Chaos

A pseudorandom bit generator is a deterministic algorithm which gives an apparently random binary sequence. PRBG's can be designed based on hardware or software. We have designed PRBG based on digital chaos generated in hardware. The hardware digital chaos is sampled such that each state of the waveform will represent a bit (either '0' or '1') depending on the state high or low respectively. The obtained bit stream is exported to MATLAB for further testing. The bit stream obtained from the digital chaos is shown as follows

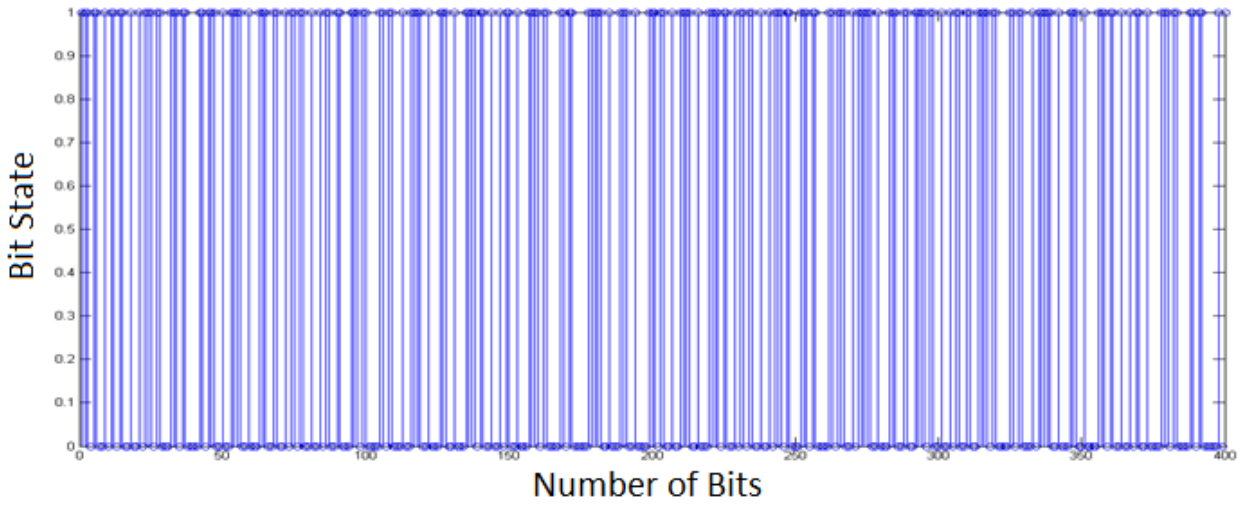

Fig. 5.1 Pseudorandom Bit Stream

5.3 Testing

The PRBG's bit stream is tested with necessary statistical test suite to confirm that it is secure [28].

The following assumptions are made with respect to random binary sequences to be tested [29, 30]:

1. **Uniformity**: At any point in the generation of a sequence of random or pseudorandom bits, the occurrence of a zero or one is equally likely, i.e., the probability of each is exactly 1/2. The expected number of zeros (or ones) is $n/2$, where n = the sequence length.

2. **Scalability**: Any test applicable to a sequence can also be applied to subsequences extracted at random. If a sequence is random, then any such extracted subsequence should also be random. Hence, any extracted subsequence should pass any test for randomness.

3. **Consistency**: The behavior of a generator must be consistent across starting values (seeds). It is inadequate to test a PRBG based on the output from a single seed, or an RBG on the basis of an output produced from a single physical output.

The various random bit generator statistical tests are

- Histogram Analysis for Multiple Bases
- Monobit Test
- Runs Test
- Binary Matrix Rank Test
- Discrete Fourier Transform Test
- Non- Overlapping Template Matching test
- Maurer's Universal Statistical Test

In order to test the PRBG sequence with tests mentioned above, we have considered n=1024 samples. The statistical tests are framed to test a particular Null Hypothesis (Sequence is not random). In all the test cases except in Histogram analysis, a probability parameter (P) is obtained after the algorithm calculation. If $P > 0.01$, then the selected sequence is said to be random which means Null hypothesis is failed [29, 30].

5.3.1 Histogram Analysis for Multiple Bases

The PRBG sequence consists of '0's and '1's. The aim of this test is to examine the distribution of states in different bases after appropriate conversions.

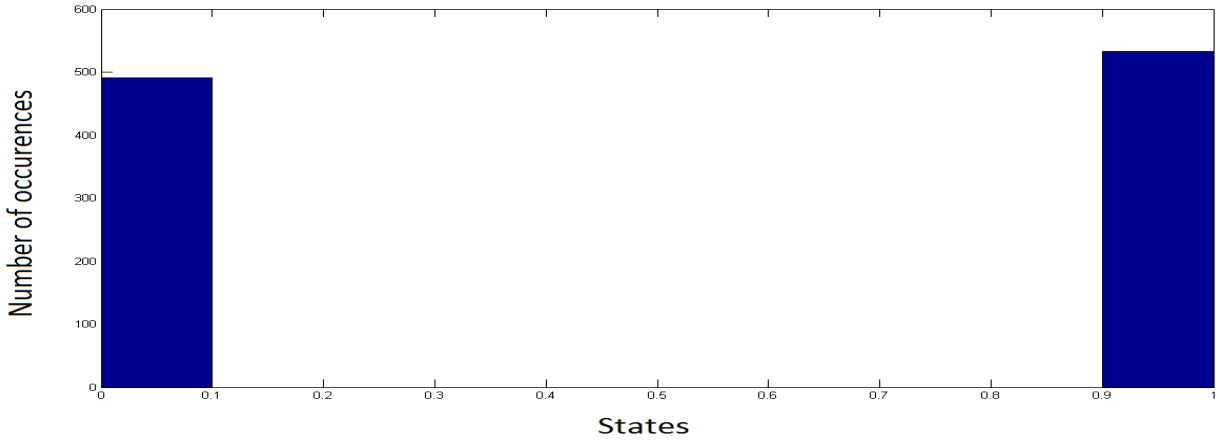

Fig. 5.2 Histogram of PRBG on Binary Base

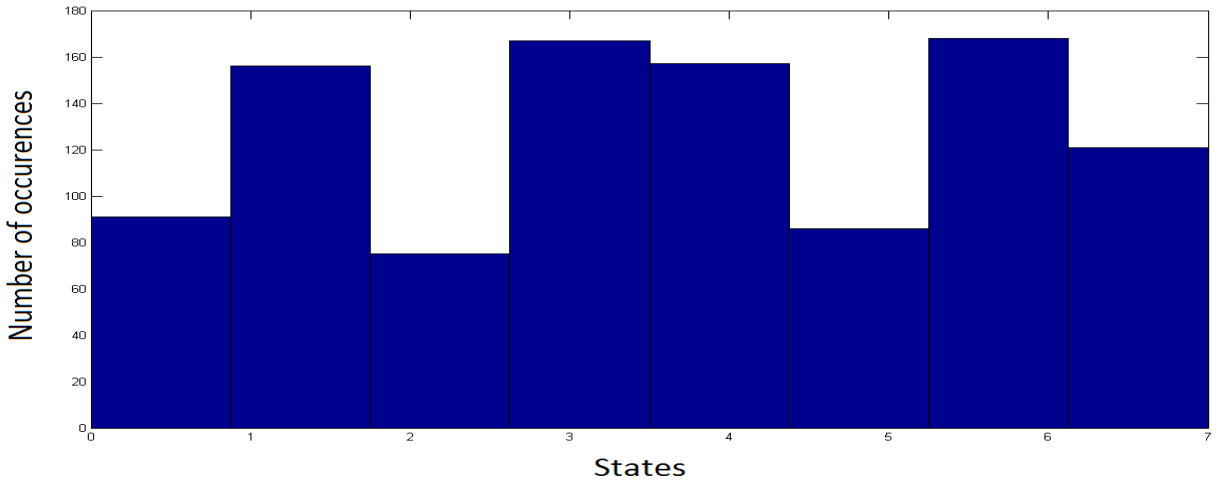

Fig. 5.3 Histogram of PRBG on Octal Base

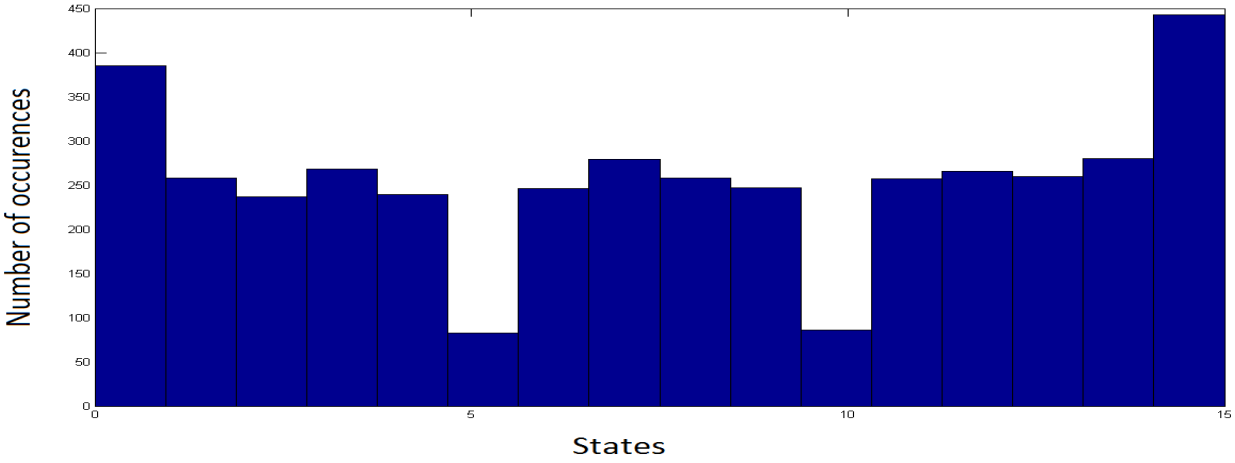

Fig. 5.4 Histogram of PRBG on Hexadecimal Base

The histograms tell how different states are distributed on different bases. In all the cases it is clear that the generated bit stream shows presence in all the states on any base.

5.3.2 Monobit Test

The purpose of this test is to determine whether the number of ones and zeros in a sequence are approximately the same as would be expected for a truly random sequence [29, 30].

5.3.3 Runs Test

The focus of this test is the total number of runs in the sequence, where a run is an uninterrupted sequence of identical bits. A run of length *k* consists of exactly *k* identical bits and is bounded before and after with a bit of the opposite value. The purpose of the runs test is to determine whether the number of runs of ones and zeros of various lengths is as expected for a random sequence [29, 30].

5.3.4 Binary Matrix Rank Test

The focus of the test is the rank of disjoint sub-matrices of the entire sequence. The purpose of this test is to check for linear dependence among fixed length substrings of the original sequence [29, 30].

5.3.5 Discrete Fourier Transform Test

The purpose of this test is to detect periodic features (i.e., repetitive patterns that are near each other) in the tested sequence that would indicate a deviation from the assumption of randomness. The intention is to detect whether the number of peaks exceeding the 95 % threshold is significantly different than 5 % [29, 30].

5.3.6 Non- Overlapping Template Matching test

The focus of this test is the number of occurrences of pre-specified target strings. The purpose of this test is to detect generators that produce too many occurrences of a given non-periodic

(aperiodic) pattern. For this test, an *m*-bit window is used to search for a specific m-bit pattern. If the pattern is not found, the window slides one bit position. If the pattern is found, the window is reset to the bit after the found pattern, and the search resumes [29, 30].

5.3.7 Maurer's Universal Statistical Test

The focus of this test is the number of bits between matching patterns (a measure that is related to the length of a compressed sequence). The purpose of the test is to detect whether or not the sequence can be significantly compressed without loss of information. A significantly compressible sequence is considered to be non-random [29, 30].

The below table shows the P value and the conclusion for each test.

TEST NAME	**P- VALUE**	**CONCLUSION**
Monobit Test	0.1894	Success
Runs Test	0.1227	Success
Binary Matrix Rank Test	0.2048	Success
Discrete Fourier Transform Test	0.8634	Success
Non- Overlapping Template Matching test	0.0352	Success
Maurer's Universal Statistical Test	0.3973	Success

Table. 5.1 Statistical Test Suite Results

5.4 Conclusion

In this chapter we have modeled Pseudorandom Bit Generator with the help of Digital Chaos generated in hardware. To confirm that various statistical tests have been done on the bit stream. All the tests proved that the sequence is pseudorandom.

REFERENCES

[1] R. Broadhurst, P. Grabosky, M. Alazab, S. Chon, Organizations and Cybercrime: *An Analysis of the Nature of Groups engaged in Cyber Crime*, Int. J. Cybercriminology, **8** (2014).

[2] M. Hilbert, *How much of the global information and communication explosion is driven by more, and how much by better technology?*, Wiley Journal of the Association for Information Science and Technology, **65**, 856-861 (2014).

[3] K. E. Himma, *Internet Security: Hacking, Counterhacking, and Society*, (Jones and Bartlett, UK, 2007).

[4] T. H. Lan, M. F. Mansour, A. H. Tewfik, *Robust high capacity data embedding*, Image Processing 2000, **1**, 581-584 (2000).

[5] X.Wu, X.Zhu, G.Q.Wu and W.Ding, *Data mining with big data*, IEEE Trans. on Knowledge and Data Engineering, **26**, 97-107 (2014).

[6] M. Lakshmanan and K. Murali, *Synchronized Chaotic Systems and Secure Communication*, Chaos in Nonlinear Oscillators: Controlling and Synchronization, **13**, 235-283 (1996).

[7] Young-Sik Kim, Jong-Hwan Kim, Sang-Hyo Kim, *A Secure Information Transmission Scheme With a Secret Key Based on Polar Coding*, IEEE Communications Letters, **18**, 937-940 (2014).

[8] K. E. Barner and G. R. Arce, *Nonlinear Signal and Image Processing: Theory, Methods and Applications*, (CRC Press, U.S, 2003).

[9] E.Bilotta and P.Pantano, *A gallery of Chua attractors*, (World Scientific, Singapore, 2008).

[10] S. H. Strogatz, *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*, (Westview Press, Cambridge, 2008).

[11] M. Ausloos, M. Dirickx, *The Logistic Map and the Route to Chaos: From the Beginnings to Modern Applications*, (Springer, US, [2006]).

[12] M. H. Jensen, P. Bak, T. Bohr, *Complete Devil's Staircase, Fractal Dimension, and Universality of Mode-Locking Structure in the Circle Map*, Phys. Rev. Lett., **50**, 1637 (1983).

[13] Xiaowu Wang, Ronnie Mainieri, J. H. Lowenstein, *Circle-map scaling in a two-dimensional setting*, Phys. Rev. A., **40**, 5382 (1989).

[14] R. G. James, K. Burke, J. P. Crutchfield, *Chaos forgets and remembers: Measuring information creation, destruction and storage*, Int. J Bifurcation Chaos, **378**, 2124 (2014).

[15] M. T. Rosenstein, J. J. Collins, C. J. De Luca, *A practical method for calculating largest Lyapunov exponents from small data sets*, Physica D, **65**, 117 (1993).

[16] P. Maragos, F. K. Sun, *Measuring the fractal dimension of signals: morphological covers and iterative optimization*, IEEE Trans. Signal Processing, **41**, 108-121 (1993).

[17] M. F. Barnsley, *Fractals Everywhere*, (Courier, Dover, 2008).

[18] G. B. Giannakis, F. Bach, R. Cendrillon, M. Mahoney, J. Neville, *Signal Processing for Big Data*, IEEE Signal Processing Magazine, **31**, 15-16 (2014).

[19] P. Grassberger and I. Procaccia, *Estimation of the Kolmogorov entropy from a chaotic signal*, Phys. Rev.A, **28**, 2591-2593 (1983).

[20] N. Marwan, M. C. Romano, M. Thiel and J. Kurths, *Recurrence Plots for the Analysis of Complex Systems*, Physics Reports, **438**, 237–329 (2007).

[21] D. Roy Choudhury and Shail B. Jain, *Linear Integrated Circuits*, (New Age International, INDIA, 2003).

[22] B. Razavi, *RF Microelectronics*, (Prentice Hall, US, 2011).

[23] J. P. Uyemura, *Chip Design for Submicron VLSI: CMOS Layout and Simulation*, (Thomson/Nelson, USA, 2006).

[24] V. A. Pedroni, *Circuit Design with VHDL*, (MIT, USA, 2004).

[25] A. Gilat, *MATLAB: An Introduction with Applications*, (Wiley, USA, 2014).

[26] L. Shujun, M. Xuanqin and C. Yuanlong, *Pseudo-random Bit Generator Based on Couple Chaotic Systems and Its Applications in Stream-Cipher Cryptography*, Springer, **2247**, 316-329 (2001).

[27] A.Uchida1, K. Amano, M. Inoue, K. Hirano, S. Naito, H. Someya, I. Oowada, T. Kurashige, M. Shiki, S. Yoshimori, K. Yoshimura, and P. Davis, *Fast physical random bit generation with chaotic semiconductor lasers*, Nature Photonics, **2**, 728-732 (2008).

[28] V. Patidar , K. K. Sud and N. K. Pareek, *A Pseudo Random Bit Generator Based on Chaotic Logistic Map and its Statistical Testing*, Informatica, **33**, 441-452 (2009).

[29] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray and S. Vo, *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*, (NIST Publications, USA , 2010).

[30] K. D. Wagner, C. K. Chin, and E. J. McCluskey, *Pseudorandom Testing,* IEEE Transactions on Computers, **C-36** (1987).