On the Application of PSpice for Localised Cloud Security

Paul Tobin
Technological University Dublin, paul.tobin@tudublin.ie

Follow this and additional works at: https://arrow.tudublin.ie/engdoc

Part of the Electrical and Electronics Commons

Recommended Citation
doi:10.21427/n3pp-fn74

This Theses, Ph.D is brought to you for free and open access by the Engineering at ARROW@TU Dublin. It has been accepted for inclusion in Doctoral by an authorized administrator of ARROW@TU Dublin. For more information, please contact yvonne.desmond@tudublin.ie, arrow.admin@tudublin.ie, brian.widdis@tudublin.ie.

This work is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 License
On the Application of PSpice for Localised Cloud Security

Author: Paul Tobin
Supervisor: Dr Michael Mckeever
Prof. Jonathan Blackledge

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy in the
College of Engineering and the Built Environment
School of Electrical and Electronic Engineering

December 9, 2018
Declaration of Authorship

I, Paul Tobin, declare that this thesis titled, “On the Application of PSpice for Localised Cloud Security”, and the work presented in it, are my own. I confirm that: this thesis, which I now submit for examination for the award of Doctor of Philosophy, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work.

This thesis was prepared according to the regulations for postgraduate study by research of the Dublin Institute of Technology and has not been submitted in whole or in part for another award in any Institute.

The work reported on in this thesis conforms to the principles and requirements of the Institute’s guidelines for ethics in research.

The Institute has permission to keep, lend or copy, this thesis in whole or in part, on condition that any such use of the material of the thesis be duly acknowledged.

Paul Tobin

December 9, 2018
Abstract

The work reported in this thesis commenced with a review of methods for creating random binary sequences for encoding data locally by the client before storing in the Cloud. The first method reviewed investigated evolutionary computing software which generated noise-producing functions from natural noise, a highly-speculative novel idea since noise is stochastic. Nevertheless, a function was created which generated noise to seed chaos oscillators which produced random binary sequences and this research led to a circuit-based one-time pad key chaos encoder for encrypting data.

Circuit-based delay chaos oscillators, initialised with sampled electronic noise, were simulated in a linear circuit simulator called PSpice. Many simulation problems were encountered because of the nonlinear nature of chaos but were solved by creating new simulation parts, tools and simulation paradigms. Simulation data from a range of chaos sources was exported and analysed using Lyapunov analysis and identified two sources which produced one-time pad sequences with maximum entropy. This led to an encoding system which generated unlimited, infinitely-long period, unique random one-time pad encryption keys for plaintext data length matching. The keys were studied for maximum entropy and passed a suite of stringent internationally-accepted statistical tests for randomness.

A prototype containing two delay chaos sources initialised by electronic noise, was produced on a double-sided printed circuit board and produced more than 200 Mbits of OTPs. According to Vladimir Kotelnikov in 1941 and Claude Shannon in 1945, one-time pad sequences are theoretically-perfect and unbreakable, provided specific rules are adhered to. Two other techniques for generating random binary sequences were researched; a new circuit element, memristance was incorporated in a Chua chaos oscillator, and a fractional-order Lorenz chaos system with order less than three. Quantum computing will present many problems to cryptographic system security when existing systems are upgraded in the near future. The only existing encoding system that will resist cryptanalysis by this system is the unconditionally-secure one-time pad encryption.
Acknowledgements

This PhD Thesis would not have been possible without the generous contribution of time, expertise and assistance, in varied forms, from friends, family, colleagues and mentors. There were so many people involved and I would like to thank them all. Professor Gerald Farrell, Director and Dean, College of Engineering and Built Environment DIT, a colleague and friend, was always very supportive and full of encouragement. Special thanks to Dr Marek Rebow who was very helpful throughout my PhD, and Professor Michael Conlon, Professor Max Ammann, Professor Mark Davis provided good advice during annual reviews. Dr Bob Duncan, Aberdeen University, with whom I shared a paper, invested precious time and energy. He is a notable scholar and I appreciate and value his friendship. Special thanks to my friend and colleague, Tony Kelly, for listening to my chaos ideas over the years, which was both fun and very stimulating.

In 2013, Stokes Professor Jonathan Blackledge accepted me as a PhD student and introduced concepts and ideas which broadened my academic horizons. After his departure to South Africa, he continued as assistant supervisor and friend. I shall be forever grateful to him for sharing his knowledge, kindness and mentorship. The onerous task of finding a replacement supervisor, who had knowledge of chaos theory, cryptography, and electronic simulation, was filled most graciously by Dr Michael Mc Keever. Mick is a friend and long-time colleague who stepped up without hesitation. He did Trojan work editing my thesis and my sincerest thanks go to him for being a patient and knowledgeable supervisor. I would like to thank a dear friend and fellow musician, Deb Munks, who was unfailing in her precise and diligent editing, and her persistent encouragement. Also, I would like to thank Kishore Karnane, product management director, PCB Group, Cadence for a full PSpice licence.

Last, but not least, my wonderful long-suffering family. I give my devoted thanks and love to my wife Marie, who put up with me being errant for the last four years, with my head buried constantly in my laptop. I had the enormous pleasure of sharing a few conference papers with my son Lee during this time. His technical and academic input was invaluable. As for my sons Roy, Scott and Keith; I look forward to interacting with them again in many ways besides lap-topping!
## Examiners in alphabetical order:

<table>
<thead>
<tr>
<th>External Examiner 1</th>
<th>External Examiner 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Professor Kevin Curran</td>
<td>Professor Michael Rycroft</td>
</tr>
<tr>
<td>School of Computing, Engineering and Intelligence Systems</td>
<td>International Space University</td>
</tr>
<tr>
<td>Computer Science Research Institute</td>
<td>Boul. Gonthier d’Andernach</td>
</tr>
<tr>
<td><a href="mailto:kj.curran@ulster.ac.uk">kj.curran@ulster.ac.uk</a></td>
<td>67400 Illkirch, France</td>
</tr>
<tr>
<td></td>
<td><a href="mailto:michael.j.rycroft@ukgateway.net">michael.j.rycroft@ukgateway.net</a></td>
</tr>
</tbody>
</table>
# Contents

Declaration of Authorship ............................................................... i

Contents ............................................................................................. v

List of Figures ....................................................................................... xi

List of Tables ........................................................................................ xvi

List of Abbreviations ........................................................................... xviii

Physical Constants ............................................................................... xx

List of Symbols ..................................................................................... xxi

1 Introduction to the Thesis .................................................................. 1

1.1 Thesis Overview ............................................................................. 1

1.1.1 Chapter Overview ...................................................................... 3

1.2 The Research Question .................................................................. 3

1.2.1 Research Methodology, Aims and Objectives .............................. 3

1.2.2 PSpice Research Aims ................................................................. 4

1.2.3 Eureqa Research Aims ................................................................. 4

1.2.4 Encoder System Research Aims .................................................. 4

1.2.5 The Research Objectives .............................................................. 5

1.3 Random and Pseudo-Random Bit Generation ................................. 5

1.3.1 Circuit-Generated Random Sequences ........................................ 6

1.3.2 Computer-Generated Random Sequences ................................... 7

1.4 OTP Entropy and Randomness ....................................................... 7

1.4.1 Entropy and Information ............................................................... 8

1.4.2 Shannon Entropy ........................................................................ 8

1.4.3 The Kolmogorov-Sinai Entropy .................................................... 9

1.4.4 The von Neumann Deskewing Algorithm ................................... 10

1.5 OTP Applications in the Cloud ....................................................... 10

1.6 Original Thesis Contributions ....................................................... 11

1.6.1 A New Cryptographic Encoder ................................................ 12

1.6.2 Novel Eureqa Contributions ....................................................... 13
1.6.3 Novel PSpice Parts and Tools ........................................ 13
1.7 Thesis Organisation ...................................................... 14
1.8 Chapter Conclusion ....................................................... 16
1.9 Publication List: Journals and Peer-reviewed conference papers ... 17

2 Cloud Cryptography ....................................................... 19
2.1 Chapter Overview ......................................................... 19
2.2 Cloud Cryptography ....................................................... 19
  2.2.1 Pre and Post-Snowden Eras in Cryptography .................... 20
2.3 The Need for Localised Encryption .................................... 21
2.4 General Data Protection Regulations .................................. 22
2.5 One-Time Pad Encryption ................................................. 23
2.6 History of One-Time Pad Encoding .................................... 24
  2.6.1 Lessons Learned from OTP History ................................. 24
  2.6.2 OTP rules for Encoding Data ...................................... 25
2.7 Modern One-to-Cloud OTP Applications .............................. 26
  2.7.1 Advantages of OTC Applications ................................. 27
  2.7.2 Encrypting DiCOM Medical Image using OTP Encoding ... 27
  2.7.3 The Paperless Courtroom ........................................... 28
2.8 Chapter Conclusion ....................................................... 30

3 Chaos-based Cipher Generation ......................................... 31
3.1 Chapter Overview ......................................................... 31
  3.1.1 Origin of Chaos Theory ............................................. 31
  3.1.2 Modern Chaos ......................................................... 32
  3.1.3 Classification of Chaos Systems ................................... 33
3.2 The Importance of Phase Space ........................................ 33
  3.2.1 Strange Attractors: Hopf-Landau vs. Ruelle-Takens ............ 34
3.3 Chaos Cryptography ....................................................... 34
  3.3.1 Comparing chaotic and cryptographic properties ............... 35
3.4 Evaluating and Selecting Chaos Systems ............................ 36
  3.4.1 Takens’s Embedding Method ....................................... 37
3.5 The Lorenz Weather Model .............................................. 39
  3.5.1 Modelling the Lorenz System in PSpice .......................... 40
  3.5.2 Sensitivity to Initial Conditions ................................ 41
3.6 Stability of the Lorenz Strange Attractor ........................... 42
  3.6.1 Fixed Points at the Attractor Lobe Centres ...................... 43
3.7 Three-Dimensional Lorenz Attractor .................................. 44
9.3 Encoding Dicom Images ........................................... 140
  9.3.1 Bias in the OTP .............................................. 141
  9.3.2 NIST Suite of Statistical Tests ............................. 141
  9.3.3 Prototype NIST Results .................................... 145
  9.3.4 Testing Prototype Chaos Circuits ......................... 146
  9.3.5 Histogram Testing Long Prototype sequences .......... 147
9.4 Chapter Conclusions ............................................. 148

10 Conclusion and Future work ................................. 149
  10.1 Conclusions and Future Work .............................. 149
  10.2 Future Work ................................................ 150
  10.3 Final Comments ............................................ 151

Bibliography .......................................................... 152

A Padé Delay Test Results ....................................... 167
  A.1 Analogue Delay Tests .......................................... 167

B Additional Chaos Circuit Designs ............................ 171
  B.1 Operational Amplifier Offset Voltages ................... 171
  B.2 The AD633 4-Quadrant Multiplier ......................... 172
     B.2.1 Active Integration ...................................... 173
     B.2.2 Summing Integrator .................................... 174
  B.3 Active Integration using the AD633 ....................... 174
     B.3.1 The High-Frequency Lorenz Chaos Generator ....... 175
  B.4 Gyrator Design ................................................ 176

C Digital Maps ........................................................ 179
  C.1 The Baker Map ................................................. 179
  C.2 The Bernoulli Map ............................................ 180
  C.3 The Lozi Map .................................................. 181
  C.4 The Gaussian Map ............................................ 182
  C.5 The Tinkerbell Map .......................................... 183

D Book List and Arduino Code .................................. 184
  D.1 Arduino Code .................................................. 185
## List of Figures

1.1 A noise-initialised chaos OTP encoding system. ........................................ 2  
1.2 Maxwell’s Demon separating gas particles. ............................................. 8  
1.3 Encoding data with OTP and modulo-two encoding. ................................. 10  
1.4 Modulo-two encryption between data and OTP sequences. ........................ 11  
1.5 The alpha OTP chaos prototype encoder with initialising noise source and Arduino Nano for processing the OTP .................................................. 12  

2.1 Modulo-two encoding and decoding. ...................................................... 23  
2.2 The SIGSALY encoder and the OTP prototype. ....................................... 24  
2.3 OTC OTP encoding. ................................................................................. 26  
2.4 Encoding DICOM medical images. ............................................................. 27  
2.5 Encoding a DICOM image using OTP encoding. ....................................... 28  
2.6 The paperless litigation courtroom. .......................................................... 29  

3.1 The LD Algorithm for choosing the chaos sources. .................................. 36  
3.2 ABM schematic for plotting a Takens strange attractor. .............................. 37  
3.3 (a) Takens attractor W v Udelayed2 (b) U v Udelayed. ............................... 38  
3.4 A 3-D attractor from one series and two delays. ....................................... 38  
3.5 Lorenz ABM chaos oscillator using net aliasing. ....................................... 40  
3.6 (a) Lorenz x, y and z signals (b) Spectrum in dB. ..................................... 41  
3.7 Showing where the trajectory enters or leaves a loop. ................................. 42  
3.8 The trajectory of a 3-D Lorenz attractor from an IC = 1 mV. ...................... 44  
3.9 Scaling the cross-products terms. ............................................................ 45  
3.10 Lorenz Xs, Ys, and Zs signals scaled down by 10. ................................... 45  
3.11 A scaled Lorenz strange attractor. .......................................................... 47  
3.12 The Lorenz analogue chaos oscillator. .................................................... 47  
3.13 Thresholding the Lorenz x signal using a 1-bit ADC. ............................... 48  
3.14 (a) Superimposing threshold levels on the biased x signal (b) Lorenz attractor (z v Xbias), with Reset and Set signal trajectories. ....................... 49  
3.15 A time delay added to the z signal feedback path. .................................. 51  
3.16 All-pass CR filter Padé delay circuit. ..................................................... 52  
3.17 (a) A 0.5 us delay between z and zd,(b) A hyperchaos funnel-type Lorenz strange attractor (x v zd) ................................................................. 52
3.18 (a) Noise and Xbias signal (b) Hyper-attractor with set and reset signals (c) Constant width digital signals and set and reset signals. . . . 54
3.19 The standard Chua chaos oscillator. .......................... 55
3.20 The Chua diode. .................................................. 56
3.21 The Chua diode nonlinear segmented negative resistance characteristic. 57
3.22 (a) Chua chaos time series (b) Chua Spectrum. .................. 57
3.23 The Chua strange attractor. ....................................... 58
3.24 A new ABM Chua system. ........................................ 58
3.25 (a) Chua attractors (b) Cubic and square plots. ................ 59
3.26 A novel delay Chua chaos oscillator. ............................ 59
3.27 (a) The Chua attractor (b) Digital signals on top, biased x signal and y analogue signals below. ................................. 60

4.1 Block diagram of a superheterodyne FM Receiver. .......... 63
4.2 Superheterodyne FM Receiver. .................................... 64
4.3 The 433 MHz data receiver. ...................................... 64
4.4 Sampling the receiver noise using an electronic switch. ...... 65
4.5 (a) Chaos pulses operating the sampling switch (b) Sampled receiver noise (c) Audio noise signal from the receiver. ................. 65
4.6 Evolving noise and chaos functions in Eureqa. .................. 66
4.7 Eureqa evolving a new function from noise. .................... 67
4.8 (a) Simulating the evolved noise function using PSpice ABM parts (b) 1-bit ADC producing PRBS from the noise function. ............... 68
4.9 (a) Eureqa random binary sequence (b) Eureka noise Spectrum (c) Eureqa Noise (d) A noise attractor (xn v xnd). .................. 68

5.1 Relationships between charge, q, and flux \( \phi \). .................. 72
5.2 (a) Doped-undoped regions (b) On/Off regions represented as two potentiometers. ............................................ 73
5.3 An ABM memristance model. ...................................... 75
5.4 Memristance signals from the ABM model. ...................... 75
5.5 (a) Three Window functions (b) Bow-tie for P = 10. .......... 76
5.6 Memristor symbol containing an HP sub-circuit model. ......... 77
5.7 A memristance emulator from electronic parts. .................. 77
5.8 A cubic function with three real roots at y = 0. .................. 77
5.9 Chua chaos oscillator with memristance. ........................ 78
5.10 A Chua chaos circuit with emulated memristance. .............. 78
5.11 The Chua memristor oscillator voltage across C2. .............. 79
5.12 Memristor-Chua oscillator attractor v(C2) vs. vC1). ............. 79
5.13 An ABM FO Lorenz circuit with thresholding. .......... 82
5.14 (a) Output from an FO Lorenz system (b) Biased X signal with set (green) and reset (red) pulse superimposed. .......... 83
5.15 (a) FO Lorenz attractor with set and reset signals. .......... 83
5.16 A Lorenz FO summing integrator. ....................... 84

6.1 (a) The continuous Logistic equation schematic (b) The Logistic map solution (c) A growth factor generator sweep part. .......... 86
6.2 (a) Sigmoidal function and integrator output (b) Continuous Logistic orbit plot for $R = 4$. .......................... 86
6.3 The LIMIT part for solving convergence problems at $R = 3.57$ and delay = 100 us. ......................... 88
6.4 (a) The Logistic map $y(0) = 0.2$ V, $R = 2.8$ (b) 3-D Logistic. ............ 88
6.5 Logistic map bifurcation diagram plotted in Matlab. .......... 89
6.6 (a) Period bifurcation plots (b) Tangential distortion. .......... 91
6.7 Measuring Feigenbaum constants. .......................... 93
6.8 A SH delay. ............................................ 94
6.9 (a) Anti-phase clocks (b) A 10 us delay between delayed and undelayed sampled signals. ...................... 94
6.10 The complete Logistic map implementation. ................. 95
6.11 The new analogue delay Logistic map. ....................... 95
6.12 (a) The New Logistic map signal, $y_n$ (b) FFT of $y_n$ (c) Attractor. .... 96
6.13 The non-uniform Logistic map PSD. ......................... 97
6.14 Small parameter perturbation moves the Poincaré trajectory point towards P1 or P2, where stable and unstable directions are shown by arrows. ......................... 99
6.15 FPs of the Logistic map. ................................ 100
6.16 Superimposing the three UPOs. ............................ 100
6.17 The OGY method for synchronising systems. ................. 101
6.18 Controlling the Logistic equation .......................... 101
6.19 Points close to the at FP = 0.7425 V. ...................... 102
6.20 UPOs near the FP line = 0.743589 V. ...................... 102
6.21 Controlling the Logistic map. ............................. 103
6.22 Controlling the Logistic map at $R = 3.9$. .................. 104
6.23 An ABM schematic of the Hénon map. ..................... 105
6.24 (a) Hénon Spectrum (b) Hénon signal (c) Attractor. ........ 106
6.25 ABM schematic for plotting a Bifurcation diagram. ........ 106
6.26 A Hénon Bifurcation plot. .............................. 107
6.27 (a) A Tent map schematic (b) The Eureqa evolved Tent map schematic. 108
6.28 (a) Tent map signal (b) Tent map attractor (c) Evolved Tent map attractor. 108
6.29 Eureqa evolving a Lorenz Poincaré map. 109

7.1 Comparing delays methods in PSpice. 112
7.2 Peakmeter subcircuit. 113
7.3 Detecting peaks in the Lorenz Z signal (resistance on the x-axis). 114
7.4 (a) VCR Zx part (b) The Zx subcircuit. 115
7.5 (a) Zx part (b) Resistance versus time. 116
7.6 The Lorenz bifurcation schematic. 117
7.7 Sampled Lorenz (Z) vs. (v(x)-v(Rv2))/I(Rsense). 117
7.8 A Chua bifurcation schematic. 118
7.9 Chua bifurcation plot period-three (1.82 kΩ to 1.87 kΩ). 118
7.10 The identity line intersecting the return map at an FP. 119
7.11 A PSpice schematic for plotting a Lorenz return map. 120
7.12 Novel design for producing a Poincaré section. 120
7.13 (a) Sampled Xs and Ys (b) Strange Attractor (c) A Poincaré section. 121
7.14 (a) Staircase generator (b) Sweep characteristic (magnified step is shown in the inset). 122

8.1 (a) The Lyapunov exponential plotted using the average function in Probe (b) Two identical Lorenz chaos systems initialised with two voltages differing by one billionth of a volt, deviate at approximately 350 ms. 125
8.2 An overview of the prototype encoder. 128
8.3 The complete prototype OTP encoder circuit diagram. 129
8.4 A two-sided OTP encoder prototype PCB. 130
8.5 The alpha prototype encoder. 130
8.6 (a) Scaled x and biased x signals with Fixed Points on the x signal (b) Reset signal on the x. 131
8.7 The scaled Lorenz strange attractor. 131
8.8 The prototype Chua strange attractor. 132
8.9 (a) Nand Clock (CD4011BE) (b) 5 V regulator (LM7805E) (c) Noise sampling (HCF4016B) (d) The data receiver antennae switch. 132
8.10 (a) XOR and 7402 NOR gates (b) The Arduino Nano with a USB port. 133
8.11 (a) Tuning capacitor dual-in-line (DIL) holder (b) Chaos isolation switch (c) Connector. 134
8.12 FIPs 140-2 certification for new cryptographic devices. 135
9.1 (a) ACF for bipolar true noise and OTP (b) ACF for a bipolar OTP.
9.2 (a) The Lena image (b) Histogram of unencoded image (c) PSD of unencoded image.
9.3 Averaging the x signal for three delays.
9.4 OTP DICOM image encoding.
9.5 Encoded image using biased OTP.
9.6 Histogram of the p-values for 158 Mbit OTPs divided into 1000 bit sequences.

A.1 Analogue delay range 200 ns to 1.2 us.
A.2 Analogue delay spectrum 200 ns to 1.2 us.
A.3 Takens embedding Attractor $Z_{vcZd}$.
A.4 Taken’s Strange Attractor $Z_d$ vs $X$ for delay range 200 ns to 1.2 us.
A.5 Delay range 1 us to 6 us.
A.6 Takens attractors vs Delay.
A.7 Large delay spectrum.
A.8 Delay range 1 us to 6 us.
A.9 Delay spectra for 1 us to 6 us.
A.10 The average x plot for 0 to 0.8 us delay.
A.11 The average x for 1 us to 1.4 us delays.

B.1 Measuring operational amplifier bias characteristics.
B.2 The operational amplifier voltage transfer function characteristic.
B.3 ABM and inverting operational amplifiers integrators.
B.4 A summing integrator circuit.
B.5 Configuring the AD633 as an integrator to implement the Lorenz chaos model.
B.6 A high-frequency Lorenz chaos oscillator.
B.7 Gyrator circuit for emulating inductive reactance.
B.8 Inductive reactance plots versus frequency.
B.9 Testing the gyrator in a parallel resonant circuit.
B.10 The parallel resonant frequency response with an inductive gyrator.

C.1 The Baker Map schematic.
C.2 The Baker Map attractor.
C.3 The Bernoulli map schematic.
C.4 A Bernoulli attractor.
C.5 The Lozi map schematic.
C.6 The Lozi attractor.
C.7 The Gaussian schematic. ........................................... 182
C.8 A Gaussian bifurcation plot. ........................................ 182
C.9 The Gaussian Orbit plot. ............................................ 182
C.10 The Tinkerbell schematic. ......................................... 183
C.11 Tinkerbell noise. ................................................... 183
# List of Tables

3.1 Comparing chaos and cryptographic properties (Alvarez and Li, 2006)  
3.2 Evaluating chaos using the LD  
3.3 The PSpice Vector1 text file of digital data  
4.1 Pin information for the noise source FM data receiver  
5.1 FO linear transfer functions approximations  
7.1 The Peak meter subcircuit netlist  
7.2 The Zx model subcircuit netlist  
8.1 Evaluating chaos using the LD  
9.1 Hypothesis Testing  
9.2 NIST tests  
9.3 NIST requirements for OTP binary sequences  
9.4 NIST results for short OTP sequences  
9.5 NIST results for large OTP sequences  
9.6 Prototype results for 100-bit sequences  
9.7 165 Mbits divided into 1 kbit sequences  
9.8 Comparing Lorenz and Chua systems
# List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABM</td>
<td>Analogue Behavioural Model</td>
</tr>
<tr>
<td>AC</td>
<td>Algorithmic Complexity</td>
</tr>
<tr>
<td>ACF</td>
<td>AutoCorrelation Function</td>
</tr>
<tr>
<td>ADALINE</td>
<td>ADaptive LINear NEuron</td>
</tr>
<tr>
<td>ADC</td>
<td>Analogue to Digital Converter</td>
</tr>
<tr>
<td>ADSF</td>
<td>analogue device scaling factor</td>
</tr>
<tr>
<td>AES</td>
<td>Advanced Encryption Standard</td>
</tr>
<tr>
<td>ANN</td>
<td>Artificial Neural Network</td>
</tr>
<tr>
<td>BIBO</td>
<td>Bounded input Bounded output</td>
</tr>
<tr>
<td>CC</td>
<td>Cloud Computing</td>
</tr>
<tr>
<td>CIA</td>
<td>Central Intelligence Agency</td>
</tr>
<tr>
<td>CPU</td>
<td>Central Processing Unit</td>
</tr>
<tr>
<td>CSA</td>
<td>Cloud Security Alliance</td>
</tr>
<tr>
<td>CSP</td>
<td>Cloud Service Providers</td>
</tr>
<tr>
<td>DAC</td>
<td>Digital to Analogue Converter</td>
</tr>
<tr>
<td>DFT</td>
<td>Discrete Frequency Transform</td>
</tr>
<tr>
<td>DH</td>
<td>Diffie-Hellman</td>
</tr>
<tr>
<td>DICOM</td>
<td>Digital Imaging and Communications in Medicine</td>
</tr>
<tr>
<td>DIL</td>
<td>Dual In Line</td>
</tr>
<tr>
<td>DSA</td>
<td>Digital Signature Algorithm</td>
</tr>
<tr>
<td>DSP</td>
<td>Digital Signal Processing</td>
</tr>
<tr>
<td>EC</td>
<td>Evolutionary Computing</td>
</tr>
<tr>
<td>ECC</td>
<td>Elliptic Curve Cryptography</td>
</tr>
<tr>
<td>EXOR</td>
<td>Exclusive Or Gate</td>
</tr>
<tr>
<td>FET</td>
<td>Fixed Evolution Time</td>
</tr>
<tr>
<td>FIPS</td>
<td>Federal Information Processing Standard</td>
</tr>
<tr>
<td>FFT</td>
<td>Fast Fourier Transform</td>
</tr>
<tr>
<td>FIT</td>
<td>Fitness</td>
</tr>
<tr>
<td>FM</td>
<td>Frequency Modulation</td>
</tr>
<tr>
<td>FP</td>
<td>Fixed Point</td>
</tr>
<tr>
<td>GDPR</td>
<td>General Data Protection Regulations</td>
</tr>
<tr>
<td>HP</td>
<td>Hewlitt-Packard</td>
</tr>
<tr>
<td>HMAC</td>
<td>Hash Message Authentication Code</td>
</tr>
<tr>
<td>IaaS</td>
<td>Infrastructure as a Service</td>
</tr>
<tr>
<td>IC</td>
<td>Initial Condition</td>
</tr>
<tr>
<td>IF</td>
<td>Intermediate Frequency</td>
</tr>
<tr>
<td>IoD</td>
<td>Institute of Directors</td>
</tr>
<tr>
<td>IoT</td>
<td>Internet of Things</td>
</tr>
<tr>
<td>KC</td>
<td>Kolmogorov complexity</td>
</tr>
<tr>
<td>KDP</td>
<td>Key distribution Problem</td>
</tr>
<tr>
<td>KS</td>
<td>Kolmogorov-Sinai entropy</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>KYD</td>
<td>Kaplan-Yorke dimension</td>
</tr>
<tr>
<td>LD</td>
<td>Lyapunov Dimension</td>
</tr>
<tr>
<td>LE</td>
<td>Lyapunov Exponent</td>
</tr>
<tr>
<td>LT</td>
<td>Lyapunov Time</td>
</tr>
<tr>
<td>MIT</td>
<td>Massachusetts Institute of Technology</td>
</tr>
<tr>
<td>MRI</td>
<td>Magnetic Resonance Intensity</td>
</tr>
<tr>
<td>MSF</td>
<td>Magnitude Scaling Factor</td>
</tr>
<tr>
<td>NAND</td>
<td>Not And gate</td>
</tr>
<tr>
<td>NDS</td>
<td>Nonlinear Dynamic Systems</td>
</tr>
<tr>
<td>NIST</td>
<td>National Institute of Standards and Technology</td>
</tr>
<tr>
<td>NN</td>
<td>Neural Network</td>
</tr>
<tr>
<td>NSA</td>
<td>National Security Agency</td>
</tr>
<tr>
<td>OR</td>
<td>Or Gate</td>
</tr>
<tr>
<td>OTC</td>
<td>One-To-Cloud</td>
</tr>
<tr>
<td>OTP</td>
<td>One-Time-Pad</td>
</tr>
<tr>
<td>OTT</td>
<td>One-Time-Tape</td>
</tr>
<tr>
<td>OGY</td>
<td>Otto, Grebogi and Yorke</td>
</tr>
<tr>
<td>PB</td>
<td>Poincaré-Bendixson</td>
</tr>
<tr>
<td>PCB</td>
<td>Printed Circuit Board</td>
</tr>
<tr>
<td>PDF</td>
<td>Probability Distribution Function</td>
</tr>
<tr>
<td>PLL</td>
<td>Phase Lock Loop</td>
</tr>
<tr>
<td>PNG</td>
<td>Portable Network Graphics</td>
</tr>
<tr>
<td>PRBS</td>
<td>Pseudo-Random Binary Sequences</td>
</tr>
<tr>
<td>PSD</td>
<td>Power Spectral Density</td>
</tr>
<tr>
<td>PSpice</td>
<td>Personal Simulation Program Integrated Circuit Emphasis</td>
</tr>
<tr>
<td>QC</td>
<td>Quantum Computing</td>
</tr>
<tr>
<td>RF</td>
<td>Radio Frequency</td>
</tr>
<tr>
<td>RIP</td>
<td>Regulation of Investigatory Powers</td>
</tr>
<tr>
<td>RMS</td>
<td>Root Mean Square</td>
</tr>
<tr>
<td>RNG</td>
<td>Random Number Generator</td>
</tr>
<tr>
<td>RSA</td>
<td>Rivest Shamir Adleman</td>
</tr>
<tr>
<td>SH</td>
<td>Sample and Hold</td>
</tr>
<tr>
<td>SHA</td>
<td>Secure Hash Algorithm</td>
</tr>
<tr>
<td>SIC</td>
<td>Sensitivity to Initial Conditions</td>
</tr>
<tr>
<td>SLD</td>
<td>Stability Loss Delay</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal to Noise Ratio</td>
</tr>
<tr>
<td>TRBS</td>
<td>True Random Binary Sequences</td>
</tr>
<tr>
<td>UI</td>
<td>User Interface</td>
</tr>
<tr>
<td>USB</td>
<td>Universal Serial Bus</td>
</tr>
<tr>
<td>UPO</td>
<td>Unstable Periodic Oscillations</td>
</tr>
<tr>
<td>USS</td>
<td>Unstable Steady States</td>
</tr>
<tr>
<td>VCR</td>
<td>Voltage-Controlled Resistance</td>
</tr>
<tr>
<td>VN</td>
<td>Von Neumann</td>
</tr>
<tr>
<td>VPWL</td>
<td>Voltage Piece-Wise-Linear</td>
</tr>
<tr>
<td>XOR</td>
<td>Exclusive OR gate</td>
</tr>
</tbody>
</table>
Physical Constants

Base of natural logarithms $e = 2.718.$
Feigenbaum first constant $\alpha = 4.6692\ldots$
Feigenbaum second constant $\delta = 2.54\ldots$
Boltzmann’s constant $k_B = 1.38065156 \times 10^{-23}$ Joules/K
List of Symbols

\( \delta \) Difference between two iterated trajectory values
\( f \) frequency in Hz
\( f(t) \) A function
\( H \) information entropy in bits
\( Ht \) Entropy at time t
\( Hw \) Weighted sum of the entropy of separate phase space compartments
\( H(t) \) Entropy computed over a particular duration of time t
\( i \) A global counter representing the ith bin of the group of bins
\( I \) Information
\( k \) Control parameter
\( K \) Boltzmann’s constant
\( k? \) k value (logistic equation) at which chaos begins
\( kn \) k value at time t or observation n
\( \lambda \) Lyapunov exponent (global, not local)
\( L \) Length or distance
\( m \) A lag, offset, displacement
\( n \) Number (position) of an iteration, or period within a sequence
\( Pi \) Probability
\( Ps \) Sequence probability
\( R \) Resistance \( \Omega \)
\( rac \) AC Resistance \( \Omega \)
\( rdc \) DC Resistance \( \Omega \)
\( Rm \) Autocorrelation at lag m
\( s2 \) Variance (same as power)
\( t \) Time seconds
\( T \) integrator time constant
\( w \) A variable
\( X \) Reactance \( \Omega \)
\( x(t) \) Value of the variable x at time or observation t
\( x* \) Attractor (a value of x)
\( Xc \) Capacitive reactance \( \Omega \)
\( xi \) (a) The ith value of x, (b) all values of x as a group
\( XL \) Inductive reactance \( \Omega \)
\( xn \) Value of the variable x at the nth observation.
\( y \) Dependent variable or its associated value
\( Z \) Impedance \( \Omega \)
\( \Sigma \) summation symbol
\( \theta \) phase angle
Dedicated to the best parents, Edward and Ann Tobin
Introduction to the Thesis

“We ought to regard the present state of the universe as the effect of its antecedent state and as the cause of the state that is to follow. An intelligence knowing all the forces acting in nature at a given instant, as well as the momentary positions of all things in the universe, would be able to comprehend in one single formula the motions of the largest bodies as well as the lightest atoms in the world, provided that its intellect were sufficiently powerful to subject all data to analysis; to it nothing would be uncertain, the future, as well as the past, would be present to its eyes”. Pierre-Simon Laplace.

1.1 Thesis Overview

Sensitivity to initial conditions (SIC) in chaos systems means future predictions are impossible due to its non-deterministic nature, contrary to the above quote. The future of the universe cannot be determined even if unlimited computing power with infinite-precision is available. Randomness rather than determinism is the universal rule and lies at the core of the research outlined in this thesis [Stone, 1989], [Laplace, 2009]. An important metric for measuring randomness is entropy and methods for quantifying it are examined in this chapter. The relationship between entropy and information was established by looking back in time to see how it developed from a solution of Maxwell’s Demon. The thesis investigates the production and application of true random binary sequences (TRBS) called one-time pads (OTP) for encoding sensitive data locally before storing in the Cloud.

A range of analogue chaos oscillators and digital chaos maps were simulated and the data exported and subjected to Lyapunov analysis in Chapter 3 and Chapter 6. From this analysis, two analogue chaos sources were selected which were initialised by sampled audio noise from a data receiver. Another initialising noise source was also investigated using evolutionary computing (EC) software called Eureqa which produced noise-generating functions. Two additional methods for generating TRBS were investigated at a later stage in the research: A new circuit element, memristance, and fractional-order (FO) chaos oscillators. One of the research aims was to design, simulate and build a prototype encoding system which makes stored cloud data unreadable to any adversary who gains access to a server.
Chapter 1. Introduction to the Thesis

The cloud client protects stored data by adding a layer of security locally which complements public encryption provided by the cloud service providers (CSP) and cloud computing (CC). The block diagram in Figure 1.1 presents the thesis in four distinct parts outlining how the research generates OTP sequences to encrypt sensitive data before storing in the Cloud.

**Figure 1.1:** A noise-initialised chaos OTP encoding system.

Block one is the mechanism for initialising chaos sources from a true source of randomly-sampled noise and ensures the chaos start from a different state each time it generates an OTP sequence [Tobin et al., 2018]. Two techniques for initialising chaos sources noise are discussed in Chapter 4: Sampling noise from a data receiver, and generating noise-producing functions from Eureqa software.

The second block represents the two novel analogue delay chaos circuits for generating OTP and is discussed in Chapter 3. Block three symbolises the threshold mechanism for producing binary sequences from each analogue chaos signals using equilibrium fixed points in the chaos system. In essence, thresholding is a 1-bit analogue-to-digital converter (ADC) discussed in Chapter 3 and produces constant-width random binary pulse sequences. The binary sequences from each chaos source were combined in parallel and applied to a deskewing algorithm which removed residual bias.

The last block represents the internationally-accepted National Institute of Standards and Technology (NIST) suite of tests (hereafter called NIST tests) for testing the randomness of the Prototype OTP sequences and is discussed in Chapter 9. These tests ensure the prototype encoder satisfies the requirements of the Federal Information Processing Standards Publications (FIPS) 140-2 validation for certification of security devices. The encoder prototype design process was aided by simulating all chaos oscillators and maps in the industrial circuit simulation standard, Cadence OrCAD PSpice software (hereafter called PSpice) to validate the research approach and methodology. Lyapunov software in Chapter 3 analysed simulation data from thirty chaos sources and selected two chaos sources for the prototype encoder. Chapter 2 shows how the client encodes data locally with OTP keys in one-to cloud (OTC) applications.

---

1PSpice is an acronym for Personal Simulation Program with Integrated Circuit Emphasis
Chapter 1. Introduction to the Thesis

1.1.1 Chapter Overview

This chapter introduces the thesis and details the research aims and objectives for creating a prototype encoder which protects data stored in the Cloud. Localised encoding could reduce the fines for late breach discovery now part of the EU legislation introduced in May 2018 [Duncan and Whittington, 2016b]. The discussion centres on randomness and illustrates how only analogue circuitry can generate keys which have no repeat cycle, unlike the finite-precision software approach. The research aims, objectives and methodology are outlined, along with the thesis research question, and is discussed at the end of this chapter. Novel thesis aspects are itemised in three areas to show how the thesis objectives are fulfilled.

1.2 The Research Question

Advanced Encryption Standard (AES)-128 encrypted data in the Cloud has been intercepted by adversaries in most countries [Sachdev and Bhansali, 2013], and poses the research question, “How can data stored in the Cloud be made unreadable using chaos sources?”. The investigation to answer the research question commenced by simulating a range of chaos systems in PSpice to select a system for generating OTP TRBS for achieving true secrecy in the Cloud.

1.2.1 Research Methodology, Aims and Objectives

The thesis methodology, research aims and objectives are discussed. The research aims are itemised as PSpice, Eureqa, and the encoding prototype, as part of the strategy to answer the research question how data stored in the Cloud can be protected by making it unreadable. The research methodology in Chapter 7 applies PSpice software [Sapini et al., 2017], [Tobin et al., 2016] to investigate a range of chaos systems as a source of random binary sequences. Applying PSpice in this way meant a range of analogue chaos oscillators and discrete chaos maps could be examined in a short timeframe. However, PSpice was designed primarily for linear circuits and simulating nonlinear chaos circuits proved problematic. Solutions to solving these problems were investigated as part of the research methodology. Furthermore, new PSpice simulation parts and simulation meters were developed to make the investigation and methodology possible. All chaos sources were simulated using analogue behavioural models (ABM) but then circuit-level models were used after successful
simulation. PSpice chaos data was exported to software to measure the Lyapunov dimension (LD) [Wolf et al., 1985] 2, a metric which enabled the selection of two chaos systems from a range of chaos systems. Methods to improve sequence entropy were investigated by connecting the binary outputs from the chaos sources in different ways (series, parallel, etc.). Another novel technique for maximising the binary sequence entropy was investigated by adding an analogue delay to each chaos oscillator. The prototype encoder chaos sources were initialised from sampled data receiver electronic noise and the complete design was mounted on a two-sided professionally-produced printed circuit board (PCB) making it possible to answer the research question.

1.2.2 PSpice Research Aims

The research aims for addressing the simulation methodology are:

- To simulate circuit-based chaos systems for generating TRBS,
- To investigate new methods for plotting bifurcation diagrams,
- To create simulation meters for investigating strange attractors,
- To create simulation techniques and parts to evaluate chaos systems,
- To investigate memristance for use in CC encryption, and
- To investigate FO chaos for increasing OTP entropy.

1.2.3 Eureqa Research Aims

The research aims for Eureqa are:

- To investigate noise-producing keys evolved from natural noise, and
- To investigate if new chaos systems could be evolved.

1.2.4 Encoder System Research Aims

The research aims for the complete prototype encoder are to develop:

- A prototype encoder containing noise and chaos sources,
- Microcontroller software to process and export to a text file, the OTP binary sequences from each chaos oscillator,
- New techniques for increasing the entropy of chaos sources,

2http://www.matjazperc.com/time/
Chapter 1. Introduction to the Thesis

- A double-sided PCB containing the complete encoder,
- Encoder designs containing memristance and FO chaos models as entropy sources, and
- A system for encrypting medical images using the OTP and modulo-two arithmetic in a JavaScript interface.

1.2.5 The Research Objectives

The research objectives are to fulfil the aims in subsections (1.2.1) to (1.2.3):

- Creating new simulation parts and tools not native to PSpice,
- Creating a mathematical model for the new analogue delay,
- Creating a test to select the delay which maximises the OTP entropy,
- Creating a prototype encoder with noise-initialised chaos sources and microcontroller mounted on a double-sided PCB,
- Incorporating a von Neumann (VN) deskewing algorithm in the microcontroller algorithm to increase the OTP entropy,
- Obtaining comprehensive NIST test results for the prototype encoder,
- Creating simulation models for noise-functions and chaos systems evolved by Eureqa,
- Creating a memristance emulator in a Chua circuit as a source of entropy, and
- Creating an FO Lorenz chaos simulation model for producing OTPs.

1.3 Random and Pseudo-Random Bit Generation

Random binary number sequences used in science, mathematics and engineering are classified according to how they are generated. However, defining randomness can evolve into a philosophical discussion because there are no mathematical methods to prove that a sequence of binary numbers is truly random. This is discussed in a paper “What is a random sequence?” [Volchan, 2002]. The only valid statement that can be made about the randomness of a sequence of binary numbers is that they fall within a statistical band of randomness, there are no absolutes, or, as Gail Gasram said, “Nothing is random, only uncertain” [Bucklew, 2013]. The future behaviour of deterministic chaos systems (hereafter referred to as chaos), is determined by the value of the initialising key voltage. Natural physical noise sources such as radioactive decay, electric storms, or electronic noise, can be used

³see section 1.4.4
to initialise chaos sources to generate TRBS. However, the deterministic nature of chaotic systems is predictable for a brief period after start-up called the Lyapunov time (LT) which is a few hundred milliseconds for systems in this thesis [Bezruchko and Smirnov, 2010]. In the final prototype, OTPs are recorded and used only after an elapsed time of approximately 3LT to avoid bias.

In this thesis, two techniques are considered for generating random binary sequences for encoding sensitive data stored in the Cloud: from circuit-based chaos oscillators initialised with noise, and from Eureqa software. The research conducted in this PhD showed that binary sequences generated on a digital platform, such as a computer or microcontroller, are pseudo-random binary sequences (PRBS) but sequences generated from analogue chaos circuits initiated by noise, are TRBS. Selecting an encoding system for producing PRBS or TRBS is mainly dictated by the complexity of the security for a particular application.

### 1.3.1 Circuit-Generated Random Sequences

Protecting sensitive data stored in the Cloud, and achieving true secrecy, was possible using a circuit-based encoder developed during the research which generated TRBS from analogue chaos. A comparative analysis of simulation data from a range of analogue chaos oscillators and digital chaos maps in Chapter 3 and Chapter 6, resulted in an alpha prototype with chaos sources initialised with electronic noise from a data receiver. This noise was randomly-sampled using an electronic switch operated by random chaos sequences from the prototype.

Sampling prevents adversarial interference from injecting seeds into the receiver for cryptanalysis purposes. The prototype was built to validate the design and thesis methodology using chaos oscillators as the source of entropy for generating OTP sequences. These random binary sequences were connected in different ways to see which method produced the best sequence entropy. From experimentation, it was found that combining two chaos source binary sequences in parallel resulted in maximum entropy in the OTP. The alpha prototype and PCB are examined in Chapter 8. Results on the encoder prototype from extensive NIST statistical randomness tests for FIPS-140 certification is discussed in Chapter 9.
1.3.2 Computer-Generated Random Sequences

Eureqa evolved noise-producing functions for PRBS generation to seed chaos generators and is discussed in Chapter 4. Daniel Wheeler [Wheeler, 1989] and Julian Palmore [Palmore and Herring, 1990] said digital systems produce only pseudo-randomness. They showed how finite state arithmetic affected nonlinear dynamical systems (NDS) and produced only PRBS because the computer introduced round-off errors with sequences of finite length, before repeating. [Biham, 1991], [Alvarez et al., 2000]. Encoding data using a PRBS generator is, however, quite acceptable for applications in stream ciphers used in symmetric public encryption algorithms [Rueppel, 2012]. However, only analogue circuit methods can produce TRBS for the applications discussed later and an essential part of the prototype is a 1-bit ADC which thresholds the analogue chaos signals to produce binary sequences.

1.4 OTP Entropy and Randomness

Entropy is a mathematical metric for quantifying the randomness of the prototype OTP binary sequences. In 1865, the French physicist, Sadi Carnot, and the German physicist, Rudolf Clausius, defined entropy as the energy in a heat engine which cannot do useful work [Thomson, 1875], [Boyling, 1973]. In the 1870s, the Austrian physicist, Ludwig Boltzmann, interpreted entropy as a measure of information. Boltzmann’s kinetic theory of gases states the probability measure of entropy always increases in a closed system, and a logarithmic relationship exists between entropy, phase space volume, \( k_B \), and the macroscopic and microscopic states of gas [Frigg and Werndl, 2011]. The work expended in a thermally-isolated container of volume \( V_f = 2V_i \), pressure \( P \) is related as \( PV = k_B T \), is \( Q \):

\[
Q = \int_{V_i}^{V_f} PdV = \int_{V_i}^{V_f} \frac{k_B T}{V} dV = k_B T \ln 2
\]  

(1.1)

This equation is engraved on Boltzmann’s headstone. Work and temperature are related as entropy, \( \Delta S = \Delta Q / T \), thus, \( S = -k_B \ln 2 \). The dispersal of energy was formulated by the physicist, Willard Gibb as \( S = k_B \log W \), where \( k_B = 1.3806 \times 10^{-23} \text{ Joules/K} \), and \( W \) is the equiprobable number of microstates with the same dimension as entropy.

\[
H(x) = -k_B \sum_{i=1}^{n} p(x_i) \ln p(x_i) \text{ bits}
\]  

(1.2)

\(^4\)Max Planck calculated Boltzmann’s constant as \( k_B = 1.38065156 \times 10^{-23} \text{ Joules/K} \)
1.4.1 Entropy and Information

A relationship between information and entropy was established in 1867 by Leo Szilard, a Hungarian electrical engineer/physicist, from his solution to Maxwell’s Demon thought experiment shown in Figure 1.2, where high and low energy gas particles in a closed system are being separated by a demon operating a gate [Tobin and Blackledge, 2014].

![Figure 1.2: Maxwell's Demon separating gas particles.](image)

The high and low energy gas particles are separated to each side of the container but the velocity and temperature of the particles remain the same. The temperature drops when the particles occupy the full container but no energy was expended because the volume is constant. This paradox seemed to violate the second law of thermodynamics because entropy cannot be lowered without expending energy. Leo Szilard’s doctoral dissertation (1922), and a companion paper, “On the decrease of entropy in a thermodynamic system by the intervention of intelligent beings” [Szilard, 1929], discussed how information gathered by Maxwell’s Demon operating the gate balanced the decrease in entropy. This formed the basis of information theory showing an increase of $k_B \log_2$ units of entropy in any measurement.

1.4.2 Shannon Entropy

Claude Shannon in Bell Telephone Laboratories also established a link between entropy and information in a paper, “A Mathematical Theory of Communication”, [Shannon, Claude E, 1949]. John von Neumann, a friend of Szilard, said to Shannon that he should use the term entropy because “no one knows what entropy is, and in a debate, you’ll always have the advantage” [Tribus and McIrvine, 1971]. However, in James Gleick’s book, “The Information”, he said this was unlikely because Shannon was unaware of Szilard’s work [Gleick, 2011b]. Shannon entropy is a measure of information expressed in binary digits, or ‘bits’, to determine precisely a system state from all possible states.
Chapter 1. Introduction to the Thesis

Higher entropy in a signal means it contains a higher amount of information and is a measure of the unpredictability or randomness of a message. The probability of an event, \( x_i \), occurring from the number of states, \( n \), with a probability between 0 and 1, is \( P(x_i) \). If a random sequence, \( X \), has \( n \) outcomes, \( x_1, x_2, \ldots, x_n \), then the Shannon information for an outcome \( x \), is \( i(x) = - \ln p(x_i) \). The average Shannon entropy measure of uncertainty in bits, is:

\[
I(x) = \sum_{i=1}^{n} p(x_i) \ln \frac{1}{p(x_i)} = - \sum_{i=1}^{n} p(x_i) \ln p(x_i) \quad (1.3)
\]

Shannon’s equation is similar to Boltzmann’s equation 1.1 but has a scaling factor and is more general [Williams, 1997].

1.4.3 The Kolmogorov-Sinai Entropy

The Russian, Andrey Kolmogorov, a gifted teacher and outstanding mathematician, created a modified form of Shannon entropy in 1959, as did Y. Sinai in the same year, hence it is known as Kolmogorov-Sinai (KS) entropy (also referred to as measure-theoretic entropy). KS entropy is an essential metric to see if a time series is chaotic and takes into account the sequence of the events, unlike Shannon entropy which cannot say if a series is chaotic. However, both Shannon and KS entropy represent the rate at which information is created. There are two limits to consider: the entropy time going to infinity and a histogram bin size going to zero. KS entropy as defined in 1.4 is the average entropy per unit time (in the limit of time going to infinity and the box size going to zero).

\[
H_{KS} = \lim_{\epsilon \to 0} \lim_{t \to \infty} \frac{1}{\text{time}} \sum_{i=1}^{n} p(x_i) \ln \frac{1}{p(x_i)} \quad (1.4)
\]

KS entropy is zero for regular systems, is finite and positive for deterministic chaos, but infinite for a random process. This entropy is related to the Lyapunov Exponent (LE) as \( H_{KS} = \sum_{1 \leq d \leq D} \lambda_d \), and is proportional to the time horizon, \( T \), on which the system is predictable. Entropy and LE are approximately equal in a chaos system, and Pesin’s theorem relates KS as the sum of positive LE’s. The LE has a disadvantage because it does not consider the resolution under which the system is observed, unlike KS entropy [Wackerbauer et al., 1994], [Pawelzik and Schuster, 1987]. The entropy of the prototype bitstream was increased by adding an analogue delay to each chaos source and is discussed in Chapter 3.
1.4.4 The von Neumann Deskewing Algorithm

John von Neumann created the VN algorithm and is a method for increasing the entropy by eliminating certain patterns in pairs of bits called dibits formed from two independent bitstream sources. The VN algorithm examines each dibit pair in the OTP and rejects ‘00’ and ‘11’, and the dibit pair ‘01’ becomes ‘0’ and ‘10’ becomes ‘1’ [Von Neumann, 1951]. Thus, 1011, will become a single 1, the 11 being discarded showing the algorithm eliminates 75 per cent of the data but is not a problem because more bits can be generated. Bit independence in the prototype was achieved using two chaos sources and forming dibits from each chaos source. The original design for the prototype implemented the VN algorithm using logic gates but at a later stage, it was easier to implement it in a Nano Arduino microcontroller.

1.5 OTP Applications in the Cloud

Protecting client/patient confidentiality in legal and medical applications using the encoder are examined in Chapter 2. Figure 1.3 shows how a bijective relationship between the sensitive data (plaintext) and the OTP, is established using an exclusive-OR gate (XOR) to implement modulo-two encryption.

![Figure 1.3: Encoding data with OTP and modulo-two encoding.](image)

The ciphertext output from the gate output is stored in the infrastructure as a service (IaaS) Cloud. Applications for the encryption system are of specialised OTC type with no key distribution problems (KDP) normally associated with OTP encoding. The encoder is operated locally by the cloud client who transports the OTP key to another location for decoding stored data and is a simple unbreakable encryption method if used correctly. Traditional bidirectional OTP communication applications have a KDP because the key is shared between two parties, however, the suggested OTC applications have no KDP because only the client encodes and

\[^{5}\text{A system has a KDP when a key is distributed between two people}\]
decodes the sensitive data. Eliminating side-channel attacks and less sophisticated hacking methods is never possible irrespective of the encryption used. However, the system designed and built during the research makes the encoded data completely unreadable and hence unbreakable for adversaries who gain access to a cloud server. A large database of OTPs was generated and stored in an air-gapped computer (not connected to the Internet) at location one. When required, a plaintext-size OTP is extracted from the OTP database and written to a flash drive which is then carried by the client for decoding the same data at another location. The final OTP production model will be similar to a small flash drive type device connected to a computer via a USB port but disconnected immediately after use.

1.6 Original Thesis Contributions

The novel prototype OTP encoder in Figure 1.4 produced TRBS from two analogue chaos sources initialised with sampled noise from a data receiver and ensured the sources are started with a new voltage each time an OTP is generated. This qualifies the OTP encoder as a true entropy source [Fischer and Drutarovsky, 2002]. The plaintext data is encrypted using modulo-two arithmetic with the OTP and the ciphertext stored in the Cloud.

![Figure 1.4: Modulo-two encryption between data and OTP sequences.](image)

At the prototype development stage, JavaScript software \(^6\) combined bits from two chaos oscillators in a novel way to satisfy the VN deskewing algorithm (Section 1.4.4) which requires alternate bits to be independent of each other to remove bias from the OTP (Chapter 2). The software also processes the OTP with the medical picture data (the plaintext) using exclusive-OR (XOR) modulo-two logic. However, in the alpha prototype, the VN algorithm was implemented in an Arduino Nano microcontroller. Chaos maps were not used in the prototype because of problems.

\(^6\)https://github.com/leetobin/ChaosEncrypt
addressed in Chapter 6 but might be considered in future encoder versions. Random binary sequences from chaos sources have maximum entropy if the oscillators operate in the chaos region. Finding this region necessitates plotting bifurcation diagram in PSpice when a parameter is varied. However, this meant creating new PSpice simulation parts, tools and algorithms, as discussed in Chapter 6. The discussion centres on novel aspects of this thesis itemised into three categories: A novel encoder design, novel applications for Eureqa, and novel PSpice applications, parts and tools.

### 1.6.1 A New Cryptographic Encoder

The alpha OTP prototype in Figure 1.5 encrypts sensitive data with no KDP for the suggested OTC applications. It has produced over 1 Gbits of OTP sequences from two chaos sources initialised with sampled noise which makes it a true source of random bits.

The OTP encoder incorporates the following novel designs:

- Initialising electronic noise from an FM data receiver. The noise is sampled using chaotic sequences from the prototype (Section 4.2),
- Novel delay Lorenz and Chua hyperchaos oscillators using an analogue Padé delay which increased the entropy of the OTPs (Section 3.9),
- A novel implementation of the delay Chua chaos oscillator (Section 3.10), and
- A novel application and implementation of the VN algorithm for creating unbiased dibit binary pairs from parallel binary streams (Section 2.7).

Figure 1.5: The alpha OTP chaos prototype encoder with initialising noise source and Arduino Nano for processing the OTP.

---

7Chaos maps were used in stream ciphers [Shujun, Xuanqin, and Yuanlong, 2001]
Chapter 1. Introduction to the Thesis

1.6.2 Novel Eureqa Contributions

A natural noise source applied to Eureqa software evolved noise-producing cryptographic functions as keys. This is a novel speculative idea since noise is stochastic but generated PRBS keys which have many applications in cryptography. Eureqa also evolved new chaos systems from existing chaos simulation data exported from PSpice.

1.6.3 Novel PSpice Parts and Tools

PSpice simulation formed a substantial part of the thesis research for evaluating and selecting chaos sources for the OTP prototype. A range of chaos sources was simulated but new simulation parts and tools, not native to PSpice, were created. The nature of nonlinear chaos systems produced convergence problems in PSpice but a novel idea of placing a LIMIT part in the feedback path meant the simulation could be performed over the complete chaos range. Another novel application of VECTOR1 parts allowed multiple digital signals to be recorded and exported from PSpice to a text file. Four digital binary streams from two chaos sources were processed in a JavaScript application to form the OTP which was XORed with the data. New contributions developed for PSpice during the investigation are itemised as novel simulation parts, meters, and paradigms for simulating chaos at circuit-level:

- An analogue ABM delay part (Section 7.2),
- A meter for plotting peaks in a chaos signal (Section 7.2),
- A meter for plotting bifurcation diagrams (Section 7.2),
- A return map meter for plotting return diagrams (Section 7.5),
- A staircase generator for bifurcation plotting (Section 7.6),
- A Poincaré meter for producing Poincaré sections (Section 7.6),
- A novel averaging test for investigating the effect of the novel Padé delay on the chaos entropy (Section 9.2),
- A new Logistic map used the Padé analogue delay (Section 6.5), and
- A new ABM FO Lorenz chaos design (Section 5.5.1).

As part of the thesis work, new features recently introduced by OrCAD were discussed in [Tobin et al., 2017a] and showed how digital signals were exported from PSpice to Matlab. Another useful feature allows the Nano Arduino microcontroller to link real-world circuits in real time to PSpice simulation.

---

8The author writes a blog on simulation of chaos: https://www.pspice.com/blog/simulation-chaos-0
1.7 Thesis Organisation

- **Chapter 1**: Security in the Cloud was achieved using a chaos encoder prototype developed for localised encryption of sensitive medical and legal data. The encoder generates unlimited amounts of OTP TRBS from chaos sources initiated with sampled electronic noise and entropy is introduced as the metric for classifying randomness. PSpice simulation enabled many chaos systems to be examined which passed all the NIST tests. Novel aspects of the thesis research were itemised in three lists.

- **Chapter 2**: The need for localised client encryption is introduced for improved cloud security. The client protects sensitive medical and legal data by encoding locally prior to storing data in the Cloud. The General data protection regulations (GDPR) discussion centres on punitive fines levied for late breach discoveries, and how local encryption could reduce or eliminate these fines. OTP history shows its effectiveness for protecting important communications during WWII and the cold war period afterwards. However, operator mistakes highlighted the need for adhering to the algorithm rules when used in local encryption.

- **Chapter 3**: The OTP encoder has novel analogue delay Lorenz and Chua chaos oscillators selected from many chaos sources using LD analysis. A test was developed to find the delay which maximised the chaos entropy. The chaos sources were thresholded by a 1-bit ADC and produced OTP sequences which were NIST tested. A JavaScript application created a ciphertext by establishing a bijective relationship between the OTP from simulation and the plaintext data using modulo-two encryption and the resulting ciphertext was stored in the Cloud.

- **Chapter 4**: Two methods for creating keys for initialising the chaos sources were investigated. The first method sampled noise chaotically from a data receiver and generated different seeding voltages each time the encoder produces OTP sequences. The second method used Eureqa noise-producing key functions evolved from natural noise but produced only PRBS because of the finite-precision computer platform.

- **Chapter 5**: Memristance was investigated for generating OTP using a prototype memristance emulator because memristors became available only mid-2017. Future work on memristance is discussed in Chapter 10 for securing the Internet of Things (IoT) devices. FO cryptography using an FO Lorenz oscillator was simulated and produced chaos from a system with an order less than three.
Chapter 1. Introduction to the Thesis

- **Chapter 6**: Discrete chaos maps for generating random binary sequences were discussed. Maps require a delay in an iterative loop normally implemented with a sample and hold (SH) technique. The author created a new map design incorporating a Padé delay which addresses the circuit complexity difficulties associated with the SH method. Stability and control of chaotic systems were examined to deal with the initial condition (IC) problems associated with chaos maps. However, it was decided not to include maps in the present alpha prototype, a decision based on Lyapunov analysis results in Chapter 3.

- **Chapter 7**: The application of PSpice software for evaluating chaos systems was considered. This simulation platform was originally designed for linear mixed analogue and digital circuit simulation and not for nonlinear systems. This necessitated creating new simulation parts, tools, and paradigms which extended the simulation range without convergence problems occurring.

- **Chapter 8**: This chapter details the alpha prototype encoder mounted on a double-sided PCB. A comparison between simulation measurement to measurements from a TiePie© HS3 software oscilloscope illustrated the accuracy of the simulation environment. The Lyapunov dimension is introduced as the main metric for selecting two prototype chaos sources. Deskewing algorithms are examined for removing bias from the generated OTP. Lastly, a discussion of the steps necessary to secure FIPS-140 certification for the encoder is discussed.

- **Chapter 9**: NIST tests were applied to random binary sequences from a simulation of the complete alpha encoder, and to sequences from the prototype [Corporation, 2003] [Huergo, 2007], [NIST, 2001]. The NIST results were presented in table form, along with additional tests such as complexity and autocorrelation. A novel averaging entropy test was also developed for determining the optimum time delay in the Lorenz and Chua chaos oscillators which would produce maximum entropy.

- **Chapter 10**: The thesis conclusions and future work are discussed. Future work which could improve the performance of the prototype encoder includes FO chaos circuits and the application of the new circuit element, memristance. Results examined in Chapter 5 indicate the encryption encoder performance might be enhanced by incorporating these into future designs. Encryption applications for memristance for protecting IoT devices, which have major security issues, are earmarked for future work.

- **Appendix A**: Extra delay results from the average test are included, with corresponding attractors are also presented.

- **Appendix B**: Additional circuit analysis not included in the main body of the thesis includes the AD633 multiplier device and summing active electronic
integration analysis, is presented. Furthermore, a gyrator design by Bernard Tellegen [Tellegen, 1948] for emulating inductance in the Chua chaos oscillator, is examined.

- **Appendix C**: More digital maps and results are presented.
- **Appendix D**: Books consulted during the research are listed, and the Arduino Nano software source code is included to show how the OTP was processed and saved to a text file for NIST testing and XOR encryption of data.

## 1.8 Chapter Conclusion

Two methods for generating random binary sequences were presented. The first method combined novel circuit-based delay analogue chaos sources seeded with electronic noise and produced OTP TRBS to encrypt locally, sensitive data. The second method used Eureqa to evolve a noise function for creating noise to seed new chaos systems also evolved from existing chaos systems. However, the combination of evolved seeding functions and chaos systems produced PRBS because of finite-precision in the digital platform. The impact of the work in this thesis on OTP production shows there are no key distribution problems when clients use it in “one-to-cloud” applications. Rethinking the OTP for these OTC applications, rather than bidirectional communication between Alice and Bob, produced a system which has no key distribution problems, and, if used correctly, is unbreakable. The discussion centred on local encryption and showed how it improved security for sensitive data stored in the Cloud.

Even though CSP use public encryption algorithms in most cases to protect cloud data, nevertheless, personalising encryption provides an extra layer of protection. If the OTP encoding rules are not broken, then it is impossible to decode the encrypted data without the original OTP [Kerckhoffs, 1883]. Quantum computing (QC) will present many problems to the security of existing cryptographic systems when hardware with a sufficient number of qubits is available. The only system to resist cryptanalysis by QC is the unconditionally-secure OTP encoder [Bernstein and Lange, 2017], [Foong, Low, and Hong, 2018].
1.9 Publication List: Journals and Peer-reviewed conference papers

  Awarded best overall paper prize 2013 http://digital-library.theiet.org/content/conferences/10.1049/ic.2013.0029


- Entropy, information, Landauer’s limit and Moore’s law Tobin, P. and Blackledge, J., Irish Signals and Systems Conference (ISSC 2014) http://digital-library.theiet.org/content/conferences/10.1049/cp.2014.0716

- Cryptography using Artificial Intelligence Blackledge, J., Bezobrazov, S. and Tobin, P. In Neural Networks (IJCNN), International Joint Conference on (pp. 1-6) July 2015. IEEE. https://www.academia.edu/25990012/Cryptography_using_Artificial_Intelligence


Chapter 1. Introduction to the Thesis

https://arrow.dit.ie/engscheleart/251/

https://arrow.dit.ie/engscheleart/250/ 

http://www.iariajournals.org/security/tocv10n34.html


- **UK Financial Institutions Stand to Lose Billions in GDPR Fines: How can They Mitigate This?** Tobin, P., Tobin, L., McKeever, M. and Blackledge, J., BAFA conference, August 24, 2017. https://www.researchgate.net/publication/323200166_UK_Financial_Institutions_Stand_to_Lose_Billions_in_GDPR_Fines_How_can_They_Mitigate_This


https://arrow.dit.ie/engscheleart2/127/

- Information Hiding using Convolutional Encoding Blackledge, J., Tobin, P., Myeza, J, 21st-22nd June 2018, 28th ISSC conference Queen’s University Belfast https://arrow.dit.ie/engscheleart/267/
2 Cloud Cryptography

“Ein Messgeist, ein Schlüssel, eine Chiffre.”
One message, one key, one cipher

2.1 Chapter Overview

In this chapter, the discussion starts with a review of cloud security to validate localised OTP encryption. Personalising the security of data stored in the Cloud is necessary because documents released by Edward Snowden showed there are unknown security issues in cloud computing (CC). Furthermore, the punitive fines for late breach discovery in GDPR legislation introduced in May 2018 could be fixed using local encoding. The cloud client forms an additional layer of localised security over encryption provided by cloud service providers (CSP) using the OTP encoder developed in the research. OTP encoding between Britain and the USA during the second world war, and between the USA and the USSR during the cold war period, was never successful cryptanalysed. These historical OTP encoding examples justify using it again to protect data stored in the Cloud. OTP encryption rules, if adhered to, show OTP encryption provides an unbreakable method for protecting data in the Cloud. One-to-cloud (OTC) encryption applications for protecting client confidentiality are suggested to show localised OTP encoder is a practical encrypting method with no key distribution problems (KDP) because the key is retained by the cloud client.

2.2 Cloud Cryptography

The interdisciplinary science of cryptography combines science, engineering and mathematics to create cryptographic keys for encoding data to a ciphertext file. In this thesis, an encoder prototype generates OTP cryptographic keys whereby cloud clients encrypt personal data before uploading to the Cloud. The following literature survey validates the statement that data encrypted by OTP sequences and stored in the Cloud is unreadable if intercepted by an adversary. Vladimir Kotelnikov in
Chapter 2. Cloud Cryptography

1941 [Molotkov, 2006] proved the OTP is a theoretically-perfect and unbreakable encoding algorithm. This was four years before the electrical engineer, Claude Shannon [Shannon, 1945], described it as information-theoretic secure, “perfect secrecy” and mathematically unbreakable, even with unlimited computing power [Sachkov, 2006]-provided specific rules are obeyed. The alpha prototype can generate unlimited amounts of cryptographic keys designed to work with CC applications, where CC is defined by the National Institute of Standards and Technology (NIST) as:

“a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. The model has five essential characteristics, three service models, and four deployment models”.

Despite the advantages of CC it is not without security risks and is discussed in a paper by Kuyoro et al. [So, 2011, [Rong, Nguyen, and Jaatun, 2013], [Yang and Tate, 2009], [Jensen et al., 2009]. The discussion in Section 2.3 centres on why localised encryption is introduced forming the focus of the thesis.

2.2.1 Pre and Post-Snowden Eras in Cryptography

Edward Snowden, a past employee in the Central Intelligence Agency (CIA) and working in the National Security Agency (NSA), released documents alleging NSA placed backdoors in certain public encryption systems [Sanger and Chen, 2014], [Schneier, 2007] creating pre and post-Snowden eras in cryptology \(^1\). This declaration caused a drop in confidence because cloud clients do not know what other security weaknesses exist in public encryption [Greenwald, 2014], [Landau, 2014]. In 2013, the New York Times linked Snowden’s disclosures to a presentation, “the possibility of a backdoor in a random number generator”, by Microsoft employees, Dan Shumow and Niels Ferguson [Shumow and Ferguson, 2007], [Young and Yung, 2005]. Ultimately, cloud clients are responsible for encoding data before storing in the Cloud, and the Cloud Security Alliance (CSA) recommends organisations employ local encryption and authentication to protect stored data. In 2000, the UK Regulation of Investigatory Powers Act (RIP) was introduced to regulate and limit public bodies to carry out investigations, surveillance, and communications interception. Updated in 2015, it outlined the limits of government bodies to use data collected from the public.

\(^1\)The government have no proposals to ban local encryption and companies will not be asked to insert “backdoors” into encrypted data.
Chapter 2. Cloud Cryptography

2.3 The Need for Localised Encryption

An exponential growth in IaaS cloud traffic is forecast to increase by twenty per cent by 2019, with workloads higher than 86 per cent anticipated at CSP centres [Index, 2013]. Businesses use CC to manage information efficiently at any location [Aich, Sen, and Dash, 2015], [Badger et al., 2012], but growing poor security issues demand a rethink of the security of data stored in the Cloud. Break-ins, hacking, and side-channel attacks are increasing at an unacceptably high rate [Aich, Sen, and Dash, 2015] and question if commercially-available encryption algorithms are effective. For example, alarming facts concerning UK security were raised in 2016 by the Institute of Directors (IoD)² and the UK government:

• Seven million attacks on 5.4 million small businesses cost £5.26 billion,
• Over seventy per cent of replies to the IoD’s policy voice survey said they received a bogus invoice, and
• Cybercrime accounted for 30 per cent of recorded crimes during the period from July 2016 to July 2017.

Furthermore, 43 per cent of clients do not know where their data is stored in the Cloud, or indeed if it is encrypted by the CSP. The Enigma syndrome concerns encryption algorithms that are the product of the authorities who want them used [Blackledge, Bezobrazov, and Tobin, 2015]:

“Can code-controllers and public encryption be trusted? ”.

If the answer is no then clients should use local encryption and ensure the following are part of the cloud contract:

• **Integrity**: Be able to detect data modification by an adversary,
• **Confidentiality**: Ensure only clients can access stored data which should be accessible at all times,
• **Authentication**: Client identity and validity of stored data is guaranteed, and
• **Accountability**: An audit trail encompassing non-repudiation, intrusion detection and prevention.

CSA recommends adding a hash message authentication code (HMAC) secure hash algorithm (SHA)-256 function to encrypted data to guard against breaches by checking the integrity of encrypted data [Barker and Roginsky, 2012]. HMAC is a unique fixed-length function derived from the plaintext and added to the ciphertext to verify the received encoded data was not changed by a third-party. The hash output

Chapter 2. Cloud Cryptography

also ensures the client’s identity is authentic [Schneier, 1996] and may be combined in several ways with the encrypted message before storing in the Cloud. It is perceived that public encryption algorithms adhering to the Kerckhoff-Shannon principle are weaker than is publicly acknowledged [Blackledge and Tobin, 2013], [Kerckhoffs, 1883], and personalising encryption locally would fix this.

2.4 General Data Protection Regulations

Breaches in cloud security are rarely discovered instantly [Bellovin, 2011], [Duncan and Whittington, 2016a], with six months or more elapsing before detection. Although this has been reduced the global average is still 146 days [Tobin et al., 2017d], [Albrecht, 2016]. The European Parliament, Council, and Commission introduced the EU GDPR in May 2018 to address security and detection problems [Tobin et al., 2017d], [Tobin et al., 2017c] and replaced the Data Protection Directive 95/46/EC. The following articles extracted from GDPR outline the EU stance on local encoding:

**Article 32** of the regulations deals with security of personal data and states:

“... the controller, and the processor shall implement appropriate technical and organisational measures to ensure a level of security appropriate to the risk, including the pseudonymisation and encryption of personal data’.

**Article 34** states for any company:

“... implemented appropriate technical and organisational protection measures such as encryption”, may avoid punitive breach fines.

Besides these two articles, the regulations cover little on encryption standards but are encouraging companies to use local encryption. Interestingly, GDPR will apply to UK businesses post-Brexit. EU and UK companies and institutions failing to report data breaches within a 72-hour time frame will be fined up to four per cent of global annual turnover [Blackmer, 2016]. These fines could amount to several hundred billion Euros resulting in companies going out of business. To reduce or eliminate fines, a two-pronged solution using encryption and immutable databases was proposed in [Tobin et al., 2017d]. In this paper, accountability (an audit trail), was discussed as an additional significant weapon in the fight against attacks [Duncan and Whittington, 2016b].
Chapter 2. Cloud Cryptography

2.5 One-Time Pad Encryption

Cloud client data is not protected against side-channel and other less sophisticated cloud attacks employed by hackers, even though the data is encrypted using the advanced encryption standard (AES)-128 algorithm. This thesis explains how this security issue can be fixed using OTP localised encryption by the client before storing data in the Cloud. Figure 2.1 shows an exclusive-OR (EXOR) logic gate (modulo-two arithmetic) encrypting the binary plaintext message data with an OTP random binary sequence.

A prototype hardware-based encoder was developed which generates unlimited information-theoretically secure random binary OTP keys to encode data locally. The message, \( m \in \{0, 1\}^n \), for some \( n \), is encoded with the key, \( k \in \{0, 1\}^n \) for some \( n \), and produces an output, \( E_k(m) = m \oplus k \). The encryption function, \( E \), maps the OTP key and the plaintext message to a ciphertext, \( c \in \{0, 1\}^n \), for some \( n \), written as \( c = E_k(m) \). A decoding function, \( D \), recovers the message by reversing the encoding process mapping the key, \( k \), and the ciphertext, \( c \), back to the plaintext message, \( m = D_k(c) \).

The OTP key sequences are generated from delay analogue chaos sources connected in parallel and a VN deskewing algorithm removes residual bias which might be present. An OTP is secure against ciphertext attack if the key is truly random because an attacker cannot decode the message plaintext from the ciphertext without the key. Brute force searching the keyspace will not help because all messages are equally likely to occur. The NSA identified the OTP as “perhaps one of the most important in the history of cryptography” 3.


![Figure 2.1: Modulo-two encoding and decoding.](image)
Chapter 2. Cloud Cryptography

2.6 History of One-Time Pad Encoding

An OTP generator patent was granted to Joseph Mauborgne and Gilbert Vernam in 1917 [Angert, 2001], but Steven Bellovin in 2011 discovered how OTPs were suggested for encoding telegrams years before [Bellovin, 2011]. In an 1882 book, “Telegraphic Code to Ensure Privacy and Secrecy in the Transmission of Telegrams”, Frank Miller said OTPs could protect telegram data [Bellovin, 2016]. A Bell labs team led by A.B.Clarke and A.Turing created the 55-tonne SIGSALY encryption system in Figure 2.2 which protected conversations between Winston Churchill and Franklin Roosevelt between 1942-1946.

![Figure 2.2: The SIGSALY encoder and the OTP prototype.](image)

SIGSALY encrypted conversations using an OTP noise key generated from a vacuum tube and recorded on vinyl which was flown across the Atlantic but this created a KDP. The system was never successfully cryptanalysed [Bennett, 1983], [Weadon, 2002]. The block diagram in the inset of Figure 2.2 shows the major parts of the alpha prototype OTP encoder examined in Chapter 3 and Chapter 8. Briefly, the encoder comprises two delay chaos oscillators initialised with natural noise and a threshold circuit produces the binary sequences. Another historical OTP system is the “hotline” connecting Russian and American governments during the 1960s Cuban crisis.

2.6.1 Lessons Learned from OTP History

The United States and British intelligence agencies created the Venona project to decrypt OTP-coded ciphertext employed by the Soviet Intelligence agency during WWII and the cold war period. The project exploited the weakness introduced by

---

4There is no record that Miller, a successful banker, actually made an OTP generator prototype
5http://www.cryptomuseum.com/crypto/hotline/index.htm

24
Chapter 2. Cloud Cryptography

Russian operators who distributed more than two copies of the same key. The famous WWII German *Enigma* encryption device was very secure\(^6\) but the *Enigma* operators used phrases like “Heil Hitler” in their messages making the ciphertext easier to be cryptanalysed by Bletchley Park staff [Gladwin, 1997], [Blackledge, 2011b]. Furthermore, one of the four-rotor wiring details was known by a Polish secret service cryptanalyst, Marian Rejewski, who passed these details to the British and reduced the possible \(10^{113}\) permutations by 25 per cent. Secrecy should not depend on the complexity of the encoding system and encryption algorithms and devices should adhere to the Kerckhoff-Shannon principle ‘*A cryptosystem should be secure if everything except the key is public knowledge*’, which is the sampled initialising noise in the prototype.

2.6.2 OTP rules for Encoding Data

Encryption has protected information from interception for thousands of years and generally was successful except when encoding operators failed to obey encryption rules. One-time pad encryption is only unbreakable provided:

- The OTP is used only once,
- The OTP is destroyed once used,
- The OTP is truly random,
- The length of the OTP is the same as the plaintext, and
- The OTP is kept secret.

The lessons learned from historical cryptographic encoding are simple; obey the algorithm rules. Historical one-to-one OTP encoding systems had operator security and KDP issues and were replaced by symmetric block ciphers and asymmetrical public key algorithms [Rijmenants, 2009], [Rijmenants, 2010]. However, modern OTP encoding systems can be found in [ShreeJain, Chandrakar, and Tiwari, 2014], [Upadhyay and Nene, 2016], [Borowski and Lesniewicz, 2012].

---

\(^6\)Enigma would still require a year to decode a message using modern computers
2.7 Modern One-to-Cloud OTP Applications

The prototype in this thesis secures sensitive material before storing in the Cloud [Hamid and Abdullah, 2016] and the following applications are suggested which have no KDP because only the client retains the key and does not distribute it. Suggested OTP applications for the encoder are:

- Medical and legal encryption,
- Academics uploading exam scripts for retrieval by office staff, and
- Accessing sensitive data when making presentations at different locations.

Figure 2.3 shows an OTP encoder for OTC application, where the prototype OTPs encode the data implemented using modulo-two arithmetic in a microcontroller.

Private OTP symmetric bidirectional encryption has a KDP because the same key is shared between two people for encoding and decoding. The prototype could be used for day-to-day Internet transactions such as emails and less sensitive material and the KDP is addressed using the courier or Sneaker method to transport the key of sufficient size between two people [Devipriya and Sasikala, 2015]. A solution is to use a secure channel to distribute the key. However, there is no KDP with the suggested OTC medical and legal novel applications discussed shortly, which is the main objection raised by opponents of OTP encryption. The cloud client carries the OTP key to another location to decode data from the Cloud [Borowski, 2016]. The prototype OTPs are stored in an air-gapped computer (one not connected to the Internet), or on a flash drive for decoding at another location. The prototype OTP encoder at the development stage used a JavaScript application software to implement an exclusive-OR gate between the OTP and the plaintext data.

7The OTPs should not be stored on a computer connected to the Internet.
2.7.1 Advantages of OTC Applications

- Ability to download secure data at any location,
- Eliminates carrying sensitive documents and protects client confidentiality,
- Encrypting documents prevents intruders from understanding data, and
- Avoids punitive GDPR fines for late breach discovery.

2.7.2 Encrypting DiCOM Medical Image using OTP Encoding

The international standard for distributing, processing and storing medical images is the Digital Imaging and Communications in Medicine (DICOM) and displays un-encoded patient personal metadata on the image [Jees and Diya, 2016]. These medical images are sent from the hospital to the doctor by post or given to the patient. Both methods are insecure and if the images are lost then patient confidentiality and any prognosis that might be read from the images, is compromised. Figure 2.4 illustrates how the OTP encoder prototype protects personal data in DICOM images stored in the Cloud.

![Figure 2.4: Encoding DICOM medical images.](https://github.com/leetobin/ChaosEncrypt)

This alternative encodes patient details in medical images and is secure because the stored images in the Cloud cannot be decoded without the OTP key. Two techniques for protecting patient details on the medical image were researched: The first method extracts and encodes DICOM image metadata separately, and the second method encodes the complete image (metadata included). The first method generates a smaller key size but the image would still be readable. A client is attending a hospital daycare outpatient for magnetic resonance intensity (MRI) head scan images on the recommendation of their doctor.

---

8https://github.com/leetobin/ChaosEncrypt
9Stegacryption is a technique for hiding medical data [Blackledge, Al-Rawi, and Tobin, 2014]
Chapter 2. Cloud Cryptography

At the hospital, the MRI images are encoded using the OTP encoder and uploaded to the Cloud. The OTP key generated is then given to the patient on a flash-memory [Scanlon and Le-Khac, 2017]. At the surgery, the patient gives the key to the doctor who decodes the MRI images stored in the Cloud [Han, Hu, and Xi, 2010]. Figure 2.5 shows the JavaScript application interface to process the DICOM MRI head image pixel data array with the prototype OTP data using modulo-two arithmetic [Tobin et al., 2017c].

The original DICOM in the left pane shows all patient data is removed and is an example where only the metadata was encrypted. However, the middle pane shows the complete image encoded. The far right panel shows the image encrypted but this time with the VN deskewing algorithm applied to the OTP to remove bias (see subsection 1.4.4).

2.7.3 The Paperless Courtroom

Transporting documentation by hand to court is insecure and the alternative is to create a single OTP file stored on flash-memory for decoding the files in court. More importantly, creating a paperless environment protects client confidentiality because no documentation can be lost in transit between office and court. This system also removes the need for carrying armfuls of lever-arch folders, or a trolley full of bankers’ boxes, allowing electronic documents to be searched efficiently and quickly in court.

10 There is no KDP since the key was not distributed
Chapter 2. Cloud Cryptography

The first step in a paperless litigation courtroom is to use the prototype encoder to encrypt legal PDF documents which are then stored in the Cloud before going to court. Figure 2.6 outlines how legal documents are encoded and is similar to AES encoding [Mohsin, Mostafa, and El Bakry, 2015].

![Diagram of paperless litigation courtroom](image)

**Figure 2.6:** The paperless litigation courtroom.

The barrister in court downloads the encoded case data from the Internet to an Android device and decodes it using the OTP on a memory stick. Besides the obviously improved security factor, searching the Android for facts is much more efficient compared to searching paper documentation (The JavaScript interface was also used for encoding legal documents). This OTC application is novel but the concept of a paperless litigation court case is not. A paperless test case appeared in the Irish Supreme Court for the first time in June 2015 and was an appeal case of Lanigan v Barry. An electronic system eCúirt Teoranta™ was developed by Kieran Morris and fellow barrister, Dáithí MacCárthaigh [Courts, 2016], [Tobin et al., 2017b] and used Android tablets which contained previously scanned case documentation and had a certain amount of protection. The system did not store data in the Cloud because the Four Courts, the main legal centre in Dublin, does not have wifi.

Another ecourt system was introduced in Malaysia [Hassan and Mokhtar, 2011], [Mohsin and El-Bakry, 2015], and in 2012 the Supreme Court Justice in England, Lord Kerr, introduced paperless litigation in a commercial dispute of Berezovsky v Abramovich, which allowed Cloud data to be searched efficiently in real-time.

---

11 Losing the key in transit does not create security issues but means generating a new key
12 The Court of Justice of the European Union launched a free CURIA application for use on smartphones and tablets for allowing access to case law in court
2.8 Chapter Conclusion

With many unknowns in cloud security and a poor security history, it is necessary and prudent to encrypt sensitive data locally before storing in the Cloud. Encrypting data using a software-based system may be cost-effective, easy to implement and update, but it is only as secure as the operating system. The alternative suggested in this thesis is to apply localised hardware-based OTP encoding. This stand-alone system has excellent security, as was demonstrated by historical examples showing it has excellent information-theoretic secure properties for protecting data. A cost-effective application for the prototype is the security of court case documentation stored in the Cloud to prevent loss of documentation during transit, as this would be a serious breach of client confidentiality. The solution researched in this thesis relies on office staff securely encoding legal documentation and storing in the Cloud prior to a court case. The OTC applications suggested have no KDP because it involves only the legal/medical teams and the OTP key is carried securely in flash-memory for decoding documents for protecting client confidentiality. The OTP prototype encoder can be used by the end-user to produce unique personal encryption keys for encrypting sensitive data before storing in Google Drive, Dropbox, etc. This localised layer of encryption is added over public encryption provided by the CSP but will not stop adversaries from hacking into servers. It will, however, make any intercepted data completely unreadable.
3 Chaos-based Cipher Generation

“It may happen that small differences in the initial conditions produce very great ones in the final phenomena. A small error in the former will produce an enormous error in the latter. Prediction becomes impossible”. Jules Henri Poincaré, 1908

3.1 Chapter Overview

This chapter starts with a review of chaos theory investigated by scientists over the past 120 years. Phase space, strange attractors, and Poincaré sections are chaos concepts introduced as part of the research outlined in this thesis. Simulation data from thirty chaos systems were exported to Lyapunov dimension (LD) analysis software for selecting the best chaos oscillators for the prototype. The design of the analogue chaos oscillators started by considering the mathematical properties for each system and the threshold mechanism for converting the analogue signal to binary was designed by determining the chaos fixed points of the strange attractor. Part of the thesis research investigated methods for increasing the entropy of the chaos sources such as adding analogue delays. Binary sequences were exported from simulation to a text file for encoding medical images which were processed in a JavaScript application. For the alpha prototype encoder, however, the OTP binary sequences were processed in an Arduino Nano microcontroller and sent to a text file via a USB port connected to a computer. The OTPs were then tested using the NIST suite of tests.

3.1.1 Origin of Chaos Theory

This section introduces concepts important to the proposed encoder prototype design. Chaos research was initiated by James Clerk Maxwell, Jacques Hadamard [Aubin and Dalmedico, 2002] in the 19th century. The French mathematician and physicist, and the last of the great polymaths, Henri Poincaré, is considered the ‘Father of Chaos Theory’ because of his work in nonlinear systems [Rössler, 1998], [Hadamard, 1898], [Poincaré, 1890]. Poincaré became interested in chaos when King

---

1Mary Lucy Cartwright contributed to chaos theory but deserves more credit [Hayman, 2000], [Wang, 2009], [Rosser Jr, 2009]
Chapter 3. Chaos-based Cipher Generation

Oscar II of Norway and Sweden announced in 1889 a competition to celebrate his 60\textsuperscript{th} birthday and which had a prize of 2500 Krone \textsuperscript{2} [Nabonnand, 1999]. The prize was awarded to anyone who could prove the Solar system was stable and was suggested to the King by the mathematician, Mittag-Leffler (1846-1927). Poincaré won the prize for his paper:

"Sur le problème à Trois Corps - On the problem of three bodies"

He wrote a book on his solution [Poincaré, 1890] but the French mathematician, Edward Phragmen discovered errors in Poincaré’s orbital stability calculations resulting in Poincaré taking his book out of circulation [Yoccoz, 2010], [Poincaré and Zsuzsanna, 1978] \textsuperscript{3}. The mathematical area of topology evolved from Poincaré’s work on the study of dynamical systems, was advanced in 1966 by the mathematician, Stephen Smale, who won the Field medal for his work on the Poincaré conjecture [Smale, 2007]. In 1968 he created the chaotic horseshoe system [Smale, 1998], which was interesting because initially, he didn’t believe in chaos theory. Smale’s work was continued by Grigory Perelman, and in August 2006 he won the Field medal for proving the Poincaré conjecture but refused to accept the Field medal [Morgan, Tian, and Flow, 2007].

3.1.2 Modern Chaos

Chaos research lacked structure and focus after Poincaré, except for the mathematician, George David Birkhoff (1884-1944), and the Russian physicist, Alexandr Andronov (1901-52) [Ginoux, 2017]. In the 1940s, Claude Shannon prescient ideas on the application of chaos in cryptography in Section 3.3 didn’t generate as much interest as his work on information theory. In 1963, Edward Lorenz, the meteorologist and mathematician, and a student of Birkhoff was modelling weather patterns and discovered quite by accident the hallmark of chaos systems which is sensitivity to initial conditions (SIC), [Lorenz, 1963]. This often-quoted term can be explained by considering a weather front starting from some IC and proceeds along a certain path, but if a tiny change in the weather occurs then it makes it traverse a different path. This is the so-called "Butterfly effect", a phrase mentioned by Edward Lorenz in a lecture in 1972 -“Predictability: Does the flap of a butterfly’s wings in Brazil set off a Tornado in Texas?” [Lorenz, 1995].

\textsuperscript{2}34 thousand euros in 2015: http://www.historicalstatistics.org/Currency converter.html

\textsuperscript{3}The Finnish Mathematician, Karl Sundman, solved the three-body problem in 1912 but his solution converged very slowly
3.1.3 Classification of Chaos Systems

The term chaos was introduced by Tien-Yien Li and James A. Yorke in a paper, “Period Three Implies Chaos”, [Li and Yorke, 1975], and James Gleick popularised chaos and made it accessible to the public, albeit with some journalese in a book called “Chaos” [Gleick, 2011a]. Astronomical chaos systems are classified as conservative and called ‘Physicist’s chaos’, where the phase space volume remains unchanged but changes its overall shape in time. Lorenz and Chua chaos oscillators are examples of continuous dissipative chaos systems called ‘Engineer’s chaos’, where phase space containing ‘Strange Attractors’ shrink with time [Ott, 1981]. Discrete chaos examples are the Logistic and Hénon maps in Chapter 6, where the signals exist at discrete periods in time. Systems excited by an external periodic signal are non-autonomous, because time is involved, but autonomous for no external excitation.

3.2 The Importance of Phase Space

A mathematical representation of the state of nonlinear dynamical systems (NDS) is described in two and three-dimensional phase space showing how the trajectory evolves with time. Also called state space as it describes the ‘state’ of an n-dimensional variable space, showing how variables change with respect to each other, or with time, and was important during the thesis research when designing the prototype encryption system. Chaos systems in this thesis are classified according to “the quantity of randomness it produces”. In analogue chaos oscillators, the trajectory of a signal in phase space is a flow, and for digital chaos maps, it is called an orbit. The NDS parameters in this thesis are voltage and current state variables in circuits. David Nolte [Nolte, 2010] said the concept of phase space is attributable to Joseph Liouville and Willard Gibbs. However, Boltzmann created the idea of a phase space, but erroneously gave credit to Liouville, with the Irish Physicist, Sir William Edwin Hamilton, and the German mathematician, Carl Jacobi, also contributed. Jules Henri Poincaré considered the trajectory of an NDS as it evolves and discussed the difficulty of producing meaningful results from chaotic time series [Poincaré, 1899], [Mancosu, 2005]. Poincaré simplified the trajectory complexity by creating a 2-D transverse slice through the 3-D model. This simple Poincaré section is a visualisation technique fundamental to the prototype encoder design approach.

4both trajectories trace the history of the system as it evolves with time
Chapter 3. Chaos-based Cipher Generation

3.2.1 Strange Attractors: Hopf-Landau vs. Ruelle-Takens

The debate about how turbulence happens in a flowing liquid, and how its trajectory evolves in phase space, has generated heated discussion and controversy [Davidson, 2015]. The flow of liquid was modelled by the Navier-Stokes equations created by the Frenchman, Claude Navier and the Irishman, Sir George Stokes, and discussed by Lev Landau [Landau, 1944] and Eberhard Hopf [Hopf, 1948]. They proposed a model for explaining turbulent flow in liquids and created the Hopf-Landau theory. The Belgian-French mathematical physicist, David Ruelle, and the Dutch mathematician, Floris Takens, disputed this model in “On the Nature of Turbulence” [Ruelle, 1972] [Ruelle and Takens, 1995]. They took a topological approach to turbulence by describing how phase space variables trace out two and three-dimensional shapes, which they named ‘strange attractors’, where the number of attractor variables is its dimensionality [Ruelle and Takens, 1995]. This argument is developed in a book, “Does God play Dice”, by the mathematician, Ian Stewart [Stewart, 1997]. Ruelle and Takens proved their model was correct by constructing strange attractors from a single chaos time series and a time delay called embedding and is investigated in Section 3.4.1 and is an important part of an algorithm used to select two chaos systems from a range of systems simulated.

3.3 Chaos Cryptography

In 1949, Claude Shannon presciently described in a paper how chaos maps could be used in a symmetric key encryption configuration [Shannon, Claude E, 1949]. He based this work on Hopf’s use of the Baker map in 1934, as a mixing model with statistical regularity [Makris and Antoniou, 2012], [Hopf, 1934]:

“Good mixing transformations are often formed by repeated products of two simple non-commuting operations. Hopf has shown, for example, that pastry dough can be mixed by a sequence of operations. The dough is first rolled out into a thin slab, then folded over, then rolled, and then folded again. In good mixing, transformation functions are complicated, involving all variables in a sensitive way. A small variation of any one variable changes the outputs considerably”.

However, this paper did not generate the same interest as his paper on information theory [Shannon, 1945], [Shannon and Weaver, 1949].
3.3.1 Comparing chaotic and cryptographic properties

Shannon discussed the relationship between chaos and cryptography by comparing ergodicity and mixing in chaos to cryptographic confusion. Table 3.1 compares chaos and cryptography properties to show why chaos is a suitable candidate for encrypting data by comparing SIC in chaos to diffusion for small changes in the key [Pellicer-Lostao and Lopez-Ruiz, 2012], [Alvarez and Li, 2006].

<table>
<thead>
<tr>
<th>Chaos property</th>
<th>Cryptographic property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ergodicity, Mixing, Auto-similar</td>
<td>Confusion</td>
<td>System output similar for any input.</td>
</tr>
<tr>
<td>Parameter change and SIC</td>
<td>Diffusion</td>
<td>Small input changes produce very different outputs.</td>
</tr>
<tr>
<td>Deterministic</td>
<td>Deterministic pseudo-randomness</td>
<td>Deterministic method for producing pseudo-randomness</td>
</tr>
<tr>
<td>Complexity</td>
<td>Algorithmic complexity</td>
<td>Simple algorithms produce highly-complex outputs</td>
</tr>
</tbody>
</table>

Robert Devaney [Hirsch, Smale, and Devaney, 2012] defined a chaos system if it is SIC, topologically transitive (a mixing process of stretching and folding) and has a set of repelling periodic points that are dense so that every point in phase space is visited by periodic orbits (ergodic). W. Diffie and M. E. Hellman published, “New Directions in Cryptography” and established public-key cryptography in 1976 and demonstrated secret communication was possible without transporting a secret key between sender and receiver [Diffie and Hellman, 1976]. In the 1990s, another phase in exploring chaos in cryptography commenced when chaos encryption algorithms were developed. S.Hayes established a link between chaotic dynamics and information theory, and showed digital information could be encoded into a chaotic signal and decoded at a later time [Hayes, Grebogi, and Ott, 1993]. Papers published on chaos cryptography since 2000 show it is possible to encrypt data using chaotic maps in a multi-algorithmic format arranged on a randomised block-by-block basis [Blackledge, 2011a], [Patidar, Pareek, and Sud, 2009].
3.4 Evaluating and Selecting Chaos Systems

Simulating nonlinear chaos systems created problems not normally encountered in linear electrical engineering and necessitated creating new simulation paradigms and parts. Simulating chaos models formed a large part of the research and was necessary for selecting chaos circuits which would produce maximum entropy OTPs. More than thirty different chaos analogue and digital maps were simulated from which several were built and tested. Figure 3.1 outlines the LD algorithm used for selecting suitable chaos systems for the prototype. This algorithm was developed by Wolf [Wolf et al., 1985], [Kodba, Perc, and Marhl, 2004].

![Diagram of the LD Algorithm](image)

**Figure 3.1**: The LD Algorithm for choosing the chaos sources.

The LE introduced in Sections (1.4.2), (3.4.1), and (8.2.1), is part of the LD algorithm (also called the Kaplan-Yorke conjecture) for evaluating, identifying and selecting chaos circuits which produce OTPs with maximum entropy. This is discussed further in Chapter 8. Computing the LD by hand is tedious and illustrated here with the Hėnon map.

Apply equation (3.1) and arrange the LE’s from the largest to the smallest, where \( j \) is the index. Applying Hėnon map standard values \( a = 1.4 \), and, \( = 0.3 \) computes the LE’s as \( \lambda_1 = 0.603 \), \( \lambda_2 = -2.34 \), and \( j = (2-1) = 1 \), hence, the LD is:

\[
LD = D_{LD} = j + \frac{\sum_{i=1}^{j} \lambda_i}{|\lambda_{i+1}|}
\]

(3.1)

Summing the LE effectively produces the LD (the Kolmogorov entropy in Chapter 9) is calculated by summing the positive LE [Eckmann and Ruelle, 1985]. Similarly, for Lorenz parameters, \( \sigma = 16 \), \( \rho = 45.92 \), and \( \beta = 4.0 \), the LE was calculated 2.16, 0.0, -32.4 to give the LD dimension, \( D = 2 + 2.16/32.4 = 2.07 \). However, hand calculations

\[D_{LD} = 1 + \frac{\lambda_1}{|\lambda_2|} = 1.26
\]

(3.2)

5The LD was evaluated using a free executable program written by Matjaz Perc [Perc, 2006]
become impractical for the chaos systems simulated and free software downloaded from the Internet produced LD results used to select the chaos systems for the prototype. A significant part of the LD algorithm in this software uses Takens’s embedding technique for producing strange attractors from a single chaos time series and is explained in Section 3.4.1.

### 3.4.1 Takens’s Embedding Method

Takens’s empirical embedding technique is the main mechanism in Alan Wolf’s fixed evolution time (FET) algorithm to calculate the LD values and compare the entropy for a range of chaos systems [Wolf et al., 1985]. Takens constructed strange attractors by delaying a chaotic time series with a suitable time lag, $\Delta t$. However, there are no analogue delay parts in the PSpice library and it was necessary to create a delay symbol and attach a sub-circuit which is discussed in Chapter 7. The delay produces another sequence set to the value between consecutive points in the time series, where the delay is chosen for $m$-coupled equations as $(2m+1)$ [Sprott, 2010]. A suitable delay was necessary to obtain reliable LD statistics and to construct a correct embedded space. There are several techniques for choosing the correct delay value but most are computationally intensive so the lag, $\Delta t$, was empirically chosen [Crilly, Earnshaw, and Jones, 2012], and the derivative is approximated as $\frac{\Delta u}{\Delta t} = \frac{u(t+\Delta t) - u(t)}{\Delta t}$. |Figure 3.2 is the Lorenz ABM chaos oscillator with delay parts at the $u$ output (Explained in Section 3.5.1).

![ABM schematic for plotting a Takens strange attractor.](image)

A second delay creates the third dimension to construct a 3-D strange attractor.
Chapter 3. Chaos-based Cipher Generation

The Takens strange attractor is plotted in Figure 3.3 (a) and 3.3 (b) by changing the horizontal time axis to the two delayed variables, v(U_{delayed2}) and v(U_{delayed1}).

Figure 3.4 is a 3-D Lorenz attractor from two delays and one chaos time series.

The last column in Table 3.2 lists the LD results for the chaos systems simulated [Kuznetsov, Alexeeva, and Leonov, 2016]. The results show analogue chaos oscillators have higher LD values compared to the digital chaos maps. The delay hyperchaos Lorenz and Chua chaos oscillators have the highest LD values marked in bold red, making them obvious candidates for the alpha prototype chaos encoder. These two chaos circuits will be analysed in the next section, starting with the Lorenz model and then progressing to the Chua model. Simulated OTP binary data was exported from each chaos source and subjected to the suite of NIST test to check for randomness. The OTP sequences were processed in a JavaScript application to encode DICOM medical images and legal PDF documents as discussed in Chapter 2.
### Table 3.2: Evaluating chaos using the LD.

<table>
<thead>
<tr>
<th>Analogue Chaos Simulated Prototype LD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chua</td>
</tr>
<tr>
<td>Delay Chua</td>
</tr>
<tr>
<td>Duffing</td>
</tr>
<tr>
<td>Lorenz (FOC)</td>
</tr>
<tr>
<td>Lorenz</td>
</tr>
<tr>
<td>Delay Lorenz</td>
</tr>
<tr>
<td>Nosé-Hoover</td>
</tr>
<tr>
<td>Rössler</td>
</tr>
<tr>
<td>Rikitake</td>
</tr>
<tr>
<td>Rucklidge</td>
</tr>
<tr>
<td>RLD</td>
</tr>
<tr>
<td>Ueda</td>
</tr>
<tr>
<td>Uraglu</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chaos Maps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baker</td>
</tr>
<tr>
<td>Bernouilly</td>
</tr>
<tr>
<td>Gaussian</td>
</tr>
<tr>
<td>Hénon</td>
</tr>
<tr>
<td>Logistic</td>
</tr>
<tr>
<td>Tinkerbell</td>
</tr>
</tbody>
</table>

### 3.5 The Lorenz Weather Model

Edward Lorenz modelled clockwise and anticlockwise atmospheric thermal convective air mass rotations in a closed container containing a shallow layer of fluid which was heated uniformly from below and cooled uniformly from above to set up Rayleigh–Bénard convection flow [Hilborn, 2000]. Lorenz reduced the twelve equation model by Barry Saltzman (a reduced form of the Navier-Stokes equations), [Saltzman, 1962] to create the Lorenz chaos oscillator as a third-order nonlinear autonomous system, expressed as three first-order coupled equations in equation (3.3). Lorenz said for chaos to exist in an autonomous ordinary differential equation systems it must have three variables and two quadratic nonlinear terms [Sparrow, 2012] 6.

\[
\begin{align*}
    x(t) &= x(t_0) - P \int_{t_0}^{t} \{x(\tau) - y(\tau)\} d\tau \\
    y(t) &= y(t_0) - \int_{t_0}^{t} \{-Rx(\tau) + y(\tau) + x(\tau)z(\tau)\} d\tau \\
    z(t) &= z(t_0) - \int_{t_0}^{t} \{Bz(\tau) - x(\tau)y(\tau)\} d\tau
\end{align*}
\]  

\(^6\)This system was proved to be chaotic by B Hassard and J Zhang [Hassard et al., 1994], and also Tucker [Tucker and Crook, 1999]
Chapter 3. Chaos-based Cipher Generation

The equations are expressed in integral form because electronic integrators will implement the equations in the prototype to implement these equations (IC terms will be dropped for ease of analysis). The parameters used by Lorenz are: Prandtl number, $P = 10$, Rayleigh number, $R = 28$, and Benard number, $B = 8/3$. Equation (3.3) represents the movement of a horizontal layer of air heated from the ground, but cooled from above, where the speed of the convective air mass $x$ is positive for clockwise rotation and negative for anticlockwise rotations. The temperature difference between the descending and ascending air mass is $y$ and the temperature change profile from a linear scale is $z$. The Rayleigh parameter, $R$ determines the locations of the fixed points (FP), with a saddle FP at the origin for $0 < R < 1$ but two stables nodes for $1 < R < 1.346$. His 1963 paper showed the weather model changed from being a linear predictive one to a nonlinear chaotic model [Lorenz, 1963].

3.5.1 Modelling the Lorenz System in PSpice

A Lorenz chaotic oscillator was designed which is one of two delay chaos oscillator for a prototype which generates random OTP binary sequences.

The Lorenz chaos oscillator in Figure 3.5 was simulated using ABM parts which allowed quicker proof-of-concept simulation with fewer convergence problems compared to circuit-level simulation [Tobin, 2007b], [Tobin, Paul, 2007]. This prototype provides an additional layer of security over existing security measures in CC. The application of chaos to cryptography is not novel, but the chaos sources included
Chapter 3. Chaos-based Cipher Generation

a delay and the chaos initialised in a novel way using an electronic noise source. Integration uses the ABM INTEG part with a gain which can be expressed as the inverse of the integrator cut-off frequency, \(-\text{InvertT}\), where \(T\) is the integrator time constant defined in a PARAM part [Tobin, 2007b]. Multiplication and scaling uses MULT and GAIN ABM parts. Signals from the ABM schematic plotted in Figure 3.6 (a) shows the signal amplitudes are too large for electronic implementation and will have to be rescaled but is considered in subsection 3.7.1.

![Figure 3.6: (a) Lorenz x, y and z signals (b) Spectrum in dB.](image)

The wideband spectra in Figure 3.6 (b) displays a dominant spectral component at 13.2 Hz which is approximately the inverse of the integrator time constant, \(T\). Plotting the spectrum of \(x\) in dB, \(\text{dB}(\text{V}(x) \times \text{V}(x))\) prevents larger amplitude spectral components from masking smaller spectral components.

3.5.2 Sensitivity to Initial Conditions

Lorenz truncated his model parameters from five places of decimals to three to speed up a simulation and produced different results to the previous simulation. Thus, he discovered in a serendipitous way the SIC condition for nonlinear systems to be chaotic when a small change in IC produces a significant difference as it evolves. A quote in Lorenz’s paper to the American Association of Scientific research in 1972, “Does the flap of a butterfly’s wings in Brazil set off a tornado in Texas”, illustrates this \(^7\). This example, whilst extreme, shows how a small change can produce significant changes very quickly and is an important factor in the design of the prototype encoder Lorenz, 1972.

\(^7\)In actual fact, this quote was suggested by a conference committee member and not by Lorenz.
3.6 Stability of the Lorenz Strange Attractor

The strange attractor trajectory in Figure 3.7 oscillates between two centres with centres called equilibrium points or FP and shows where the trajectory changes direction\(^8\).

![Initial Condition X = 4 V](image)

**Figure 3.7**: Showing where the trajectory enters or leaves a loop.

Lorenz observed that the trajectory went from one attractor wing to the other wing when a certain distance from the origin was exceeded and also where the trajectory enters the next loop. FPs were chosen as the thresholding levels for converting the \(x\) signal to binary form and were calculated from the Jacobian matrix of partial derivatives of each state variable in equation (3.3).

\[
J (x, y, z) = \begin{pmatrix}
\frac{\partial P(y-x)}{\partial y} & \frac{\partial P(y-x)}{\partial y} & \frac{\partial P(y-x)}{\partial y} \\
\frac{\partial (Rx-y-xz)}{\partial y} & \frac{\partial (Rx-y-xz)}{\partial y} & \frac{\partial (Rx-y-xz)}{\partial y} \\
\frac{\partial (xy-Bz)}{\partial y} & \frac{\partial (xy-Bz)}{\partial y} & \frac{\partial (xy-Bz)}{\partial y}
\end{pmatrix}
= \begin{pmatrix}
-P & P & 0 \\
R - z & -1 & -x \\
y & x & -B
\end{pmatrix}
\]

(3.4)

The Lorenz oscillator is approximately linear at \(x = y = z = 0\) representing a stationary air mass. From the characteristic matrix, \((J-\lambda I)\):

\[
\text{det} \left( J (0, 0, 0) - \lambda I \right) = \begin{vmatrix}
-(P + \lambda) & P & 0 \\
R & -1 + \lambda & 0 \\
0 & 0 & -(B + \lambda)
\end{vmatrix}
= - (\lambda + B) (\lambda^2 + (P+1)\lambda + P(1-R))
\]

\(^8\)Smale’s fourteenth problem, “the Lorenz attractor was a strange attractor” was solved by Tucker in 2002 [Tucker, 2002]
Chapter 3. Chaos-based Cipher Generation

The behaviour of the trajectory is determined from the eigenvalues which are calculated from the discriminant roots of (3.5) and from $b^2 - 4ac$. The sign of $4ac$ determines if the roots are imaginary or real. The characteristic polynomial is:

$$\lambda_1 = -B, \lambda_{2,3} = -\frac{1}{2} (P + 1) \pm \frac{1}{2} \sqrt{(P + 1)^2 - 4P (1 - R)} = -5.5 \pm 17.327$$  \hspace{1cm} (3.6)

The eigenvalues for standard Lorenz parameters are calculated as -2.666, -22.8277, and +11.8277, for 1 mV IC on each system integrator. The system becomes unstable at the origin because of the positive eigenvalue.

### 3.6.1 Fixed Points at the Attractor Lobe Centres

The loci of each wing of the Lorenz strange attractor are the FPs of the system with coordinates $\pm x_1, \pm y_1$ and are the two coordinates chosen as the threshold levels for generating binary sequences: $\frac{dx}{dt}_{\mid P=10} = 10(y - x) = 0 \Rightarrow x = y$. Substitute $R = 28$ into the second Lorenz equation: $\frac{dy}{dt}_{\mid x=y} = 28x - x - xz = 0 \Rightarrow z = 27$. Placing this value of $z$ into the $z$ equation yields:

$$\frac{dz}{dt} = x^2 - Bz = 0 \Rightarrow x = \pm \sqrt{B(R - 1)}.$$  \hspace{1cm} (3.7)

The FP coordinates are $C_{1,2} = \{ + \sqrt{B(R - 1)}, - \sqrt{B(R - 1)}, (R - 1) \}$ and calculated for the standard Lorenz parameters coordinates as $x = y = \pm 8.485$ V, for $z = 27$ V. These points represent steady convection and are stable if:

$$R < \frac{P (P + B + 3)}{(P - B - 1)}$$  \hspace{1cm} (3.8)

A Hopf bifurcation point occurs for a value of $R$ [Sparrow, 2012]:

$$R^* = \frac{P (P + B + 3)}{(P - B - 1)} = \frac{10(10 + \frac{8}{3} + 3)}{(10 - \frac{8}{3} - 1)} \approx 24.73684$$  \hspace{1cm} (3.9)

Substituting the FPs into the Jacobian:

$$J (x = y = 8.4853, z = 27) = \begin{pmatrix}
-(P + \lambda) & P & 0 \\
(R - 27) & -(1 + \lambda) & -8.485 \\
8.485 & 8.485 & -(B + \lambda)
\end{pmatrix}$$  \hspace{1cm} (3.10)

FPs at the origin and eigenvalues at -13.856, 0.08959 ± j10.194 create a focus node. When the pair of complex conjugate eigenvalues cross the s-plane imaginary axis the
system becomes bounded-input bounded-output (BIBO) unstable. The 3-D Lorenz strange attractor trajectory in Figure 3.8 is constructed from the \(x\) and \(y\) signals and start from an IC of 1 mV.

The coordinates at the centre of each lobe are the FPs found by examining equation (3.7) and are visited by the signal trajectory in a random fashion. The three FPs constitute a saddle point at the origin, and two spiral points at \(C_{1,2}\) (two imaginary eigenvalues). It was decided that when the trajectory crosses one centre loci a binary ‘1’ is generated, and a ‘0’ for the other centre.

### 3.7 Three-Dimensional Lorenz Attractor

Phase space volume contracts exponentially fast towards zero in bounded dissipative trajectories in 3-D chaos systems [Cuomo, Oppenheim, and Strogatz, 1993]. The system diverges with time as shown by the divergence theorem [Kocarev, Galias, and Lian, 2009], hence: \(V = \int_V \nabla \cdot f dV\). For the Lorenz oscillator:

\[
\nabla \cdot f = \frac{\partial}{\partial x} \frac{dx}{dt} + \frac{\partial}{\partial y} \frac{dy}{dt} + \frac{\partial}{\partial z} \frac{dz}{dt} = \frac{\partial(P(y-x))}{\partial x} + \frac{\partial(Rx-y-xz)}{\partial y} + \frac{\partial(xy-Bz)}{\partial z} = -(P + 1 + B)
\]

Hence: \(V = \int_V \nabla \cdot f dV = -\int_V (P + B + 1) dV\). For standard parameters, a solution with constant divergence is: \(V(t) = V(0)e^{-(1+P+B)t} = V(0)e^{-13.666t}\). The volume shrinks exponentially at a rate \((P+B+1) = 13.667\) as time approaches zero.
3.7.1 Amplitude Scaling

The Lorenz chaos signals are too large for electronic implementation and hence the system variables need to be rescaled setting $X_s = 10x$, $Y_s = 10y$, and $Z_s = 10z$, where $(X_s, Y_s, Z_s) \in R^3$.

\[
\begin{align*}
X_s &= -P \int_{t_0}^{t} \{X_s - Y_s\} \, dt \\
Y_s &= -R \int_{t_0}^{t} \{-RX_s + Y_s + 10X_sZ_s\} \, dt \\
Z_s &= -B \int_{t_0}^{t} \{Bz - 10X_sY_s\} \, dt \\
\end{align*}
\]

Figure 3.9 shows the nonlinear $xy$ and $xz$ terms scaled by ten.

Each integrator has an IC of 1 mV and a gain of -10 (sets the frequency range). Figure 3.10 shows the three Lorenz signals rescaled down by ten.

Figure 3.10: Lorenz $X_s$, $Y_s$, and $Z_s$ signals scaled down by 10.
ABM parts were replaced with model parts representing actual integrated circuits after a successful proof-of-concept simulation. This technique is examined in Section 3.8, and in [Tobin, 2007b], [Tobin, Paul, 2007].

### 3.7.2 Time Scaling the Lorenz Model

Time scaling allows the chaos oscillator frequency of the prototype to be changed to a different value [Blakely, Eskridge, and Corron, 2007]. Adding more capacitance in parallel to the existing capacitors using an 8-pin dual in-line (DIL) socket changes the time scale and hence the frequency range (Chapter 8). The ratio of a parameter, \( \tau \), to \( T \) scales time as:

\[
\frac{dX_s}{dt} \bigg|_{(t=\tau/T)} = T \frac{dX_s}{d\tau} = p(Y_s - X_s) |_{p=10}
\]

\[
\Rightarrow X_s = -\frac{10}{\tau} \int (X_s - Y_s) d\tau
\]  

(3.13)

\[
T \frac{d(Y_s)}{d\tau} = RX_s - Y_s - X_sZ_s |_{R=28}
\]

\[
\Rightarrow Y_s = -\frac{1}{\tau} \int (-28X_s + Y_s + 10X_sZ_s) d\tau
\]  

(3.14)

\[
T \frac{d(Z_s)}{d\tau} = X_sY_s - BZ_s |_{B=\frac{8}{3}}
\]

\[
\Rightarrow Z_s = -\frac{1}{\tau} \int (\frac{8}{3}Z_s - 10X_sY_s) d\tau
\]  

(3.15)

The equations are scaled by \( 1/T \) allowing the circuit to operate at a different frequency by selecting a different capacitance (or resistance values). The equations expressed using general parameter values:

\[
X_s = -\frac{P}{\tau} \int_{t_0}^{t} \{X_s - Y_s\} dt
\]

\[
Y_s = -\frac{1}{\tau} \int_{t_0}^{t} \{-RX_s + Y_s + 10X_sZ_s\} dt
\]

\[
Z_s = -\frac{1}{\tau} \int_{t_0}^{t} \{Bz - 10X_sY_s\} dt
\]  

(3.16)

The gain of the x ABM INTEG part is now \(-10/T\), where \( T \) is 1 s. Scaling the equations by ten reduced the signal voltage amplitude to a suitable range for electronic devices but changes the FPs by ten to \( \pm 0.848 \) V, as examined in Section 3.7.1. The scaled 2-D Lorenz ‘butterfly attractor’ in Figure 3.11 is a fractal-type structure where the trajectories converge on a set of points.

---

\(^9\)\( T \) is the characteristic time of the system, and is the approximate time to switch between the two FP at the loci centre of each attractor wing [Vincent and Yu, 1991]
3.8 The Analogue Lorenz Chaos Oscillator

Figure 3.12 shows three operational amplifiers configured as inverting summing integrators for solving the Lorenz equations.

The nonlinear cross-product terms, \( xy \) and \( xz \), are necessary for chaos to exist and use AD633 multiplier devices to replace the ABM MULT part. The AD633 has a 0.1
internal scaling factor and hence the integrator gain must be increased by reducing the resistance between the multiplier and the integrator input by 10 (R4 and R6). The integrator gain is \(-1/(R_2C_1)\) and is the inverse of the integrator time constant \(1/(10^9\times 10^{-9}) = 10^3\). The gain sets the operating frequency by changing all three integrator capacitors by the same amount. It was found from experimentation that changing the standard Lorenz parameters, \(B = 2.666, P = 10, R = 28\), to \(B = 2.8, P = 11, R = 27.5\), increased the OTP entropy. The x integrator resistors, \(R_1\) and \(R_2\), were reduced from 1 MΩ to 100 kΩ for the Prandtl P value of ten. Thus, the resistor value is \(R/P\). Resistor values in the schematic used braces \{...\} containing the analogue device scaling factor (ADSF), and the magnitude scaling factor (MSF). These are defined in a PARAM part \(^{10}\). The 0.1 AD633 scaling factor is compensated by reducing the resistor by the ADSF and MSF factors making the 1 MΩ resistor \(R/ADSF/MSF\).

The component: 
\[\begin{align*}
R_1 &= R_2 = 100\ \text{kΩ}, \\
R_3 &= 36.3\ \text{kΩ}, \\
R_4 &= R_6 = 10\ \text{kΩ}, \\
R_5 &= 1\ \text{MΩ}, \\
R_7 &= 357\ \text{kΩ}, \text{ and } C = 330\ \text{pF}.
\end{align*}\]

### 3.8.1 Thresholding the Lorenz x Signal

The \(x\) signal was converted to binary using a 1-bit ADC threshold design with circuit components calculated using the scaled loci coordinates, \(C_{1,2} = \pm 0.848\ \text{V}, 2.7\ \text{V}\). The bipolar \(x\) was changed to a polar format by adding a 4 Volt DC bias and changed the threshold levels to 3.15 V and 4.848 V. The threshold comparator in Figure 3.13 converts the analogue signals using these two levels to produce two digital sequences.

\[\text{Figure 3.13: Thresholding the Lorenz } x \text{ signal using a 1-bit ADC.}\]

The \(x\) signal is biased by 4 V DC using the potential divider, R8 and R9 and the signal is isolated using a unity-gain buffer amplifier. The potential divider sets the

\[\text{\textsuperscript{10}The use of braces and parameters rather than component values gives greater flexibility during simulation}\]
threshold values of the $x$ signal and calculated using the FPs, a potentiometer of 1 MΩ, and a 1.24 V (Vref) reference voltage: $V_{\text{high}} = 4.84 \text{ V} = V_{\text{ref}} \frac{R_{11} + R_{12} + R_{13}}{R_{13}}$, and $V_{\text{low}} = 3.15 \text{ V} = V_{\text{ref}} \frac{R_{11} + R_{12} + R_{13}}{R_{12} + R_{13}}$. $R_{11} = 607 \text{ kΩ}$, $R_{12} = 138 \text{ kΩ}$, and $R_{13} = 256 \text{ kΩ}$.

Figure 3.14 (a) shows the biased threshold voltages on the $x$ signal.

![Figure 3.14](image)

The 1-bit ADC produced varying-width out-of-phase pulses because of the nature of chaos signals, but monostables converted them to a constant width. The set and reset pulse sequences are shown superimposed on the $(x-y)$ strange attractor in Figure 3.14 (b) and line up at the two loci centres. The original design applied the constant width set and reset pulses to an exclusive-OR gate (XOR) producing a clock stream at the gate output. This clock stream and the reset pulse stream controlled when the OTP ‘ones’ and ‘zeroes’ were written to the Arduino microcontroller. However, the final design processed the set and reset dibits through a VN deskewing algorithm in the Arduino Nano.

### 3.8.2 Creating Dibits in PSpice

The OTP bias-removing VN algorithm introduced in Chapter 1 processed pairs of bits (dibits) from each chaos source (four digital sequences) which were recorded using four PSpice VECTOR1 parts attached to each monostable output by specifying a directory location and file name in the VECTOR1 part. However, it was discovered accidentally that specifying the same name and directory in each part meant all four signals were written to the same text file name at that location. This serendipitous accident meant the dibit pairs from the Lorenz and Chua circuits were now specified on the same line in the text file. This solved the problem PSpice created because the time vectors were written to the text file. Table 3.3 shows part of the Vector1 text
Chapter 3. Chaos-based Cipher Generation

file exported from PSpice showing the header row, time and four binary column vectors.

TABLE 3.3: The PSpice Vector1 text file of digital data.

<table>
<thead>
<tr>
<th></th>
<th>PSpice</th>
<th>SLd</th>
<th>RLd</th>
<th>Scd</th>
<th>Rcd</th>
</tr>
</thead>
<tbody>
<tr>
<td>28.22 us</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>36.77 us</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>59.22 us</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>69.22 us</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Examining the time vector column shows it is not changing monotonically, i.e., not increasing by a constant time factor, but only displays the time when the binary is changing. In this example, the line showing 1011 is the set and reset binary values from the Lorenz and Chua circuits at 28.22 us. The wire net-alias names, SLd, RLd, SCd, RCd are the set and reset signal locations on the four threshold monostable circuits outputs. Net-alias is a schematic wiring technique for simplifying schematics where wires with the same net alias will connect together. The JavaScript software application, described in Chapter 2, reads the text file but ignores the time vector column and applies the VN algorithm to the dibits formed from the set and reset bitstreams. The first bit of the second column is the set pulse, and the second bit is the reset pulse and the goal was to ignore the temporal information and remove the dibit 00 and 11 pairs. The VN algorithm examines dibit pairs in the OTP and rejects ‘00’ and ‘11’, but dibit ‘01’ becomes ‘0’ and ‘10’ becomes ‘1’ [Von Neumann, 1951]. Thus, in the first line, 1011, becomes a single 1, the 11 being discarded. The application stores the dibits in a text file for NIST testing.

3.9 The Time-Delay Lorenz Chaos Oscillator

A suitable time delay, found from experimentation, was added to the z signal feedback path of the ABM Lorenz chaos oscillator in Figure 3.15 to increase the entropy. The new Lorenz equations with delay, \( \tau \):

\[
\begin{align*}
    x(t) &= -11 \int_{t_0}^{t} \{ x(t) - y(t) \} dt \\
    y(t) &= - \int_{t_0}^{t} \{ -27.5x(t) + y(t) + x(t)z(t - \tau) \} dt \\
    z(t) &= - \int_{t_0}^{t} \{ 2.8z(t - \tau) - x(t)y(t) \} dt
\end{align*}
\]

(3.17)

\[\text{\textsuperscript{11}}\text{The algorithm eliminates 75 per cent of the data, but this is not a problem because more bits can be generated}\]
Chapter 3. Chaos-based Cipher Generation

Figure 3.15: A time delay added to the $z$ signal feedback path.

Adding a signal delay increased the entropy of the model and created a hyperchaos system as discussed in subsection 3.9.2. The entropy change for a range of delay values was measured by creating a new test described in Chapter 9. Briefly, the test involves averaging the Lorenz signals for a range of delays. The test shows if the average of the signal oscillates around the horizontal axis it indicates the final sequence should consist of equal numbers of ones and zeroes but in a random order. Applying the NIST tests in Chapter 9 ensure randomness is possible but would produce an unacceptably long process time for each delay value.

3.9.1 Padé Delay Lorenz Oscillator

The Lorenz chaos encoder in Figure 3.16 shows an all-pass first-order CR analogue filter Padé delay in the $z$ feedback path [Lam, 1979]. The all-pass filter transfer function is:

$$\frac{V_{out}(s)}{V_{in}(s)} = -\frac{sC5R18}{1 + sC5R18}$$  \hspace{1cm} (3.18)

Delays in digital signal processing (DSP) are processed using the $z$-transform [Tobin, 2007c]:

$$z = e^{-s\tau} = \frac{e^{-s\tau/2}}{e^{s\tau/2}} \approx \frac{1 + s\tau/2}{1 - s\tau/2}$$  \hspace{1cm} (3.19)

12The test determined heuristically which Lorenz parameters produced maximum entropy.
An expression for the delay between $z$ and $zd$ in Figure 3.17 is obtained from (3.18) and (3.19) as $\tau = 0.5*C5R18 = 0.5$ us. The chaos oscillators ceased to work when a delay was added to all three signal paths in the Lorenz oscillator, thus the final configuration was a single delay which produced maximum entropy sequences.

3.9.2 Hyperchaos

Hyperchaos is relatively new and introduced by Otto Rössler in 1979 [Rossler, 1979]. As discovered in this research, adding a delay to chaos systems created hyperchaos, as evidenced from the typical funnel-shape in the strange attractor in Figure 3.17. Hyperchaos autonomous system must have four or more state space variables and at least two positive Lyapunov exponents (LE) [Sprott, 2010]. This requirement is
met by adding a time delay dimension to the z-path of the third-order Lorenz system. The delay was varied to investigate the effect on the entropy (more delay results can be examined in Appendix B). Figure 3.18 (a) shows the sampled initialising noise had very little effect on the dynamics of the hyperchaos systems. The noise and x signal are displayed in the left pane with the attractor and pulse sequences superimposed in the middle pane of Figure 3.18 (b). Figure 3.18 (c) shows the set and reset threshold signals from the comparators and the constant-width pulse sequences from the monostables at the top of the diagram.
Figure 3.18: (a) Noise and Xbias signal (b) Hyper-attractor with set and reset signals (c) Constant width digital signals and set and reset signals.
Chapter 3. Chaos-based Cipher Generation

3.10 The Chua Chaos Oscillator

In 1983, Leon Chua was trying to prove the Lorenz oscillator was chaotic and in the process created a new analogue chaos oscillator system shown in Figure 3.19. This design was not used in the prototype and instead, a novel implementation of the Chua equations using standard multiplier integrated circuits produced a more robust design.

![Figure 3.19: The standard Chua chaos oscillator.](image)

The Chua chaos system three first-order coupled equations in equation (3.20) has capacitor voltages, $V_{C_1}, V_{C_2}$, and inductor current, $I_L$ state variables.\(^{13}\)

\[
\begin{align*}
C_1 \frac{dV_{C_1}}{dt} &= \frac{(V_{C_2} - V_{C_1})}{R_1} - f(V_{C_1}) \\
C_2 \frac{dV_{C_2}}{dt} &= -\frac{(V_{C_2} - V_{C_1})}{R_1} + I_L \\
L \frac{dI_L}{dt} &= -R_{coil}I_L - V_{C_2}
\end{align*}
\] (3.20)

Inductors can radiate energy and generally are more awkward to use in circuits so a gyrator design was used.\(^{14}\) The standard Chua oscillator consists of a parallel-tuned $LCR$ type circuit, across which a ‘Chua diode’ is connected [Kennedy, 1993], [Kennedy, Michael Peter, 1992], [Kennedy, Peter, 2013]. The ‘Chua diode’ produced segmented nonlinear negative resistances formed from two current inverter stages and explained in the next section, are used to overcome losses in the tuned-circuit and also introduce nonlinearity. The component values shown in this design were those used in Matsumoto’s 1995 design et al. [Kennedy, Michael Peter, 1992], [Kiliç, 2010].

\(^{13}\) An analysis of the Chua circuit is given in [Kennedy, 1993]

\(^{14}\) The inductor was emulated with a gyrator design and can be examined in Appendix D
3.10.1 The ‘Chua Diode’

Figure 3.20 shows a dual operational amplifier configured so that a segmented negative nonlinear resistance appears across the input terminals.

![The Chua diode diagram](image)

The non-inverting operational amplifier has a gain, $A_V$, hence the current in $R_1$ and $R_4$ is proportional to the voltage difference across each resistance (assuming very little current flows into the positive input). Increasing the input voltage increases the output voltage because of stage gain and causes the current to decrease producing a negative resistance characteristic. The non-inverting gain is:

$$A_V = \frac{V_o}{V_{in}} = 1 + \frac{R_2}{R_3} \Rightarrow V_o = V_{in}(1 + \frac{R_2}{R_3}) \quad (3.21)$$

The input current is

$$I_{in} = \frac{V_{in} - V_{out}}{R_1} \quad (3.22)$$

The input resistance is calculated by substituting 3.21 into 3.22:

$$R_{in} = \frac{V_{in}}{I_{in}} = \frac{V_{in}}{\frac{V_{in} - V_{out}}{R_1}} = \frac{V_{in}R_1}{V_{in} - V_{in}(1 + \frac{R_2}{R_3})} = -R_1 \frac{R_3}{R_2} \quad (3.23)$$

The second stage resistor $R_4 = 22 \, k\Omega$ is ten times $R_1$ to avoid loading it and the input resistance is $-R_1$ for $R_2 = R_3$. The Chua diode segmented negative resistance characteristic in Figure 3.21 has three nonlinear piece-wise linear segments whose break points were calculated from the saturation voltage (see Appendix B Section B.1) and component values:

$$B_{p1} = -\frac{R_3}{R_2 + R_3}E_{sat} = -\frac{2.2 \, k\Omega}{2.42 \, k\Omega} 10.5 = -9.5 \, V \quad B_{p2} = -\frac{R_6}{R_5 + R_6}E_{sat} = -1.39 \, V \quad (3.24)$$
Chapter 3. Chaos-based Cipher Generation

Figure 3.21: The Chua diode nonlinear segmented negative resistance characteristic.

The slope of the piece-wise segments were $-0.429 \text{ mS} = -3.37 \text{ mA }/ (9.1-1.25)$, and $-2 \text{ mA}/2.5 \text{ V} = -0.8 \text{ mS}$: The slopes for $R_1 = R_2$, and $R_4 = R_5$:

$$m_1 = -\frac{1}{R_3} - \frac{1}{R_6} = -\frac{1}{2200} - \frac{1}{3300} = -0.75 \text{ mS}$$

$$m_0 = -\frac{1}{R_3} - \frac{1}{R_4} = -\frac{1}{2200} - \frac{1}{22000} = -0.5 \text{ mS}$$

(3.25)

The standard Chua chaos oscillator produced the time series shown in Figure 3.22 (a). The spectra for each signal is shown in Figure 3.22 (b).

Figure 3.22: (a) Chua chaos time series (b) Chua Spectrum.
Chapter 3. Chaos-based Cipher Generation

The Chua strange attractors in Figure 3.23 were plotted by replacing the horizontal time axis after simulation with the $x$ signal. The vertical axes are $y$ and $z$.

3.10.2 New Chua Chaos oscillator Implementation

Experiments on the standard Chua oscillator showed certain non-robust characteristics and hence a new implementation was created using equations (3.26) obtained from [Kiliç, 2010]. This system proved a more robust design for the prototype.

\[
\begin{align*}
x(t) &= -\int_{t_0}^{t} \{-1.66x(t) - 10y(t) + 0.625x^3(t)\}dt \\
y(t) &= -\int_{t_0}^{t} \{-x(t) + y(t) - z(t)\}dt \\
z(t) &= -\int_{t_0}^{t} \{14.286y(t)\}dt
\end{align*}
\] (3.26)

A novel ABM implementation of equation (3.26) is drawn in Figure 3.24.
Chapter 3. Chaos-based Cipher Generation

The attractors produced by this circuit are shown in Figure 3.25.

![Figure 3.25: (a) Chua attractors (b) Cubic and square plots.](image)

The cubic and squared characteristics are displayed in the right pane. The continuous nonlinear negative characteristic can be compared to the Chua diode segmented negative resistance characteristic examined in Section 3.10.1.

### 3.11 Novel Delay Chua chaos oscillator

A new implementation of equation (3.26) in Figure 3.26 shows an analogue delay in the $y$ signal path.

\[
x = -\text{int}(-10y - 1.66x + 0.625x^3)
\]
\[
y = -\text{int}(-x + y - z)
\]
\[
z = -\text{int}(14.286(y))
\]

![Figure 3.26: A novel delay Chua chaos oscillator.](image)
Chapter 3. Chaos-based Cipher Generation

The location for the Padé delay was determined from experimentation for best signal entropy. Two series-connected AD633 devices implemented the cubic term in (3.26), and a 1-bit ADC threshold circuit for producing random binary sequences is similar to the one used in the Lorenz oscillator in Page 51 in Section 3.9.1. The three coupled equations for the Chua system with a Padé delay in the $y$ signal path, are:

$$
\begin{align*}
    x(t) &= -\int_{t_0}^{t} \left\{-1.66x(t) - 10y(t - \tau) + 0.625x(t)^3\right\} dt \\
    y(t) &= -\int_{t_0}^{t} \left\{-x(t) + y(t - \tau) - z(t)\right\} dt \\
    z(t) &= -\int_{t_0}^{t} \left\{14.286y(t)\right\} dt
\end{align*}
$$

Figure 3.27 (a) shows the Chua $z$ and $x_{bias}$ attractor with the FP used to calculate the threshold points. Digital signals produced from the thresholded biased $x$ signal are shown in Figure 3.27 (b).

The $x$ signal was biased to make the design of the threshold mechanism an easier task. The FPs were calculated to design the threshold circuit and components using the same technique considered in the Lorenz threshold design examined in Subsection 3.8.1. The final prototype circuit comprising the delay Lorenz and Chua chaos, with threshold circuits, are examined in Chapter 8.
3.12 Chapter Conclusion

A brief history of chaos as a new branch of science and the scientists involved was discussed. Modern chaos was inadvertently initiated in the 1960s by Lorenz when he discovered SIC in weather patterns. Phase space forms an integral part of the OTP threshold encoder design for producing OTP sequences. Takens’s embedding method for reconstructing attractors from a time series formed part of Wolf’s algorithm in Section 3.4 for calculating the LD for selecting chaos sources with maximum entropy for the prototype.

The encoder design was simulated to show how true random binary sequences could be produced from the chaos sources thresholded at the FPs. OTP sequences from PSpice simulation were successfully processed in a JavaScript application to encode DICOM medical images and legal PDF documents. The entropy of the OTP binary sequences was increased by adding a novel delay mechanism into the feedback path of both chaos sources. Adding a time delay dimension turned the regular chaos system into hyperchaos systems as observed from the funnel-shaped Lorenz strange attractor. The standard tuned-circuit Chua oscillator was examined, but a novel implementation was designed and implemented in the final encoder prototype.

The prototype delay multi-chaos encoder design was simulated in PSpice and a 1 Mbit binary stream exported and tested for randomness using the non-parameterised NIST tests. The other parameter tests required impractically-long OTPs (Chapter 9) and were not simulated but left until the prototype was built, where very large OTP could be produced.

---

15Recent weather forecasting gives improved results using a range of SIC figures in an ‘ensemble forecasting’ technique by running multiple simulations.
4 Keys for Seeding Chaos

“Anyone who attempts to generate random numbers by deterministic means is, of course, living in a state of sin.” John von Neumann

4.1 Chapter Overview

This chapter describes two methods for producing keys or initial conditions for the prototype chaos systems and is a process known as seeding. The first method investigated electronic noise from a 433 MHz data receiver which was sampled using an electronic switch operated by chaotic pulses from the chaos oscillator. The second method evolves noise-producing functions in evolutionary computing (EC) software for producing an unlimited amount of keys for seeding chaos systems. The evolved function from natural noise was post-processed to generate an iterative function.

4.2 Initialising Noise from a Data Receiver

OTPs are theoretically unbreakable and are computationally and unconditionally secure, provided the OTP encryption rule, one message, one key, one cypher, is adhered to. A chaos oscillator starting from a known voltage level makes the chaos trajectory predictable. To ensure this does not happen, the chaos oscillators are initialised from a true random initialising voltage audio electronic noise from a commercial detuned 433.29 MHz data receiver integrated circuit. This classifies the encoder prototype as a source of true randomness [Fischer and Drutarovsky, 2002]. Receiver noise is made up from lightning storms, cosmic and solar radiation, and also man-made electrical noise \(^1\). The noise also comes from internal receiver circuits and the overall effect is to create an audio hiss when the receiver is detuned. However, the receiver antennae could pick up a signal transmitted by an adversary for cryptanalysis purposes but randomly sampling the initialising noise ensures the chaos oscillators start from a random voltage level each time an OTP is generated. The frequency modulation

\(^1\) Man-made electrical noise is generated whenever the current is interrupted rapidly, as in car ignition systems.
(FM) receiver shown in Figure 4.1 [Tobin, 2007a] receives a microvolt signal from the antennae and amplifies it up to volts. The receiver translates the incoming radio frequency (RF) signal down to the standard 10.7 MHz intermediate frequency (IF) using the superheterodyne principle.

The detector block is a phase lock loop (PLL) to extracts the audio noise from the IF radio signal which is amplified using an audio power amplifier.

4.2.1 Quantifying Receiver Noise

The chaos oscillators seeded with white noise from a data receiver has a large bandwidth. Part of this noise is thermal (Johnson) noise from internal resistors and integrated circuits and is generated from the random interaction between free electrons and vibrating ions in the crystal lattice \(^2\). The root-mean-square (RMS) noise voltage in a resistance, \(R\), is \(v_n = \sqrt{4k_B TB R}\) Volts, where Boltzmann’s constant, \(k_B\) is \(1.38 \times 10^{-23}\) J/K, \(T\) is the resistor temperature in degree Kelvin (K) = \((273^0 + \text{room temp }^0\text{C})\), and \(B\) is the bandwidth in Hz. Maximum power is transferred when the noise source resistance, \(R\), equals the load, \(R_L\), hence the voltage across \(R_L\) is half the input voltage yielding thermal noise power:

\[
P_n = \left(\frac{v_n}{2}\right)^2 = \frac{\left(\sqrt{4k_B TB R}\right)^2}{4R} = k_B TB Wats\]

This shows white noise is proportional to bandwidth and temperature and has a spectrum from 0 Hz to \(10^{13}\) Hz, with a constant power spectral density measured per Hertz of bandwidth \((k_B T Wats/Hz)\). \(N_0\) is the amount of noise in each Hz of the bandwidth, with units of watts per Hz in the range \(10^{-7}\) to \(10^{-21}\) watts per Hz (White noise). Thus, if \(N = k_B TB\), then \(N_0 = N/B\), or \(N = N_0 B\).

\(^2\)Shot noise and frequency dependent noise is also generated in electronic circuits
4.2.2 Noise Factor and Noise Figure

Electronic receivers add noise to the RF signal and noise picked up by the antennae and is quantified as the ratio of signal to noise (SNR) power ratio in dB:

\[
SNR = 10 \log_{10} \frac{S}{N} \text{dB} \tag{4.2}
\]

Thus, the output SNR is always less than the input SNR, and the ratio of the two SNRs is the noise ratio \(F\), a factor which degrades the system SNR at the system input. \(F\) is defined: \(F = NR = \frac{S_{in}}{S_{out}}\). The FM block diagram for the superheterodyne receiver is shown in Figure 4.2.

The FM receiver in Figure 4.3 is the source of initialising noise for the encoder.

The pin information for the FM data receiver is shown in Table 4.1.

<table>
<thead>
<tr>
<th>Pin</th>
<th>RRFQ2</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>+Vcc</td>
<td>Used</td>
</tr>
<tr>
<td>2, 15</td>
<td>Gnd</td>
<td>Used</td>
</tr>
<tr>
<td>1</td>
<td>Data input (Antennae)</td>
<td>Not used (future design)</td>
</tr>
<tr>
<td>14</td>
<td>Received Signal output</td>
<td>Not used (future design)</td>
</tr>
<tr>
<td>17</td>
<td>Audio Output</td>
<td>Used</td>
</tr>
<tr>
<td>18</td>
<td>Data output</td>
<td>Not used</td>
</tr>
</tbody>
</table>
The prototype chaos oscillators will have a predictable trajectory unless the initialising voltage is different each time OTPs are generated. To make sure this happens, the prototype chaos systems are initialised from a random unpredictable noise source. The noise could be corrupted by an external periodic signal but sampling the noise means the random key (seed) is presented to the Chua and Lorenz chaos (Chapter 3) oscillators. Sampling the receiver noise prevents adversaries from successfully cryptanalysing the ciphertext by adding a periodic RF interference signal to the receiver input. The circuit for sampling the receiver noise for initialising the chaos sources is shown in Figure 4.4.

![Circuit diagram](image)

**Figure 4.4:** Sampling the receiver noise using an electronic switch.

The noise from the receiver in Figure 4.5 (c) is sampled using an electronic switch. The switch is operated by the reset pulse sequence from the Lorenz generator (discussed in Chapter 3) shown in Figure 4.5 (a). The sampled noise thus supplies a seed which exists for a brief time only and is shown in Figure 4.5 (b).

![Graphs](image)

**Figure 4.5:** (a) Chaos pulses operating the sampling switch (b) Sampled receiver noise (c) Audio noise signal from the receiver.

³For simulation only, the noise was supplied from a noise generator part called RND
4.3 Noise from Evolutionary Computing

The second method investigated for generating keys for seeding the chaos oscillators applied an EC software called Eureqa, developed by Lip Hodson and Michael Schmidt et al. in Cornell Creative Machines Lab, Nutonian.Inc [Stoutemyer, 2013], [Techniques, 1996]. The concept of EC first appeared in an MIT academic journal in 1993 [Bäck and Schwefel, 1993], where *survival of the fittest* algorithmic techniques mimicked Nature’s evolutionary selection process was discussed. Eureqa genetic algorithms try to fit equations to applied natural noise data using symbolic regression using trigonometric, arithmetic, and exponential operator functions [Mckhann, 2011], [Schmidt and Lipson, 2009], [Gupta, Mittal, and Mittal, 2008]. Symbolic regression optimisation minimises a fitness function between input and output data by creating and rejecting up to 10 billion equations per second. The optimisation can be terminated when the evolved function almost matches the input signal.

4.3.1 Evolving Noise-Producing Functions in Eureqa

Figure 4.6 illustrates how Eureqa generates key-producing functions from natural noise source 4.

![Figure 4.6: Evolving noise and chaos functions in Eureqa.](image)

Eureqa evolves a function to represent the noise [Haahr, 2010], [Haahr, 1999] and is a novel, highly-speculative idea, since true noise is stochastic. Nevertheless, noise-generating functions were produced for seeding chaos oscillators and for generating OTP sequences. Block A shows natural noise downloaded from a trusted noise source and applied to Eureqa. Block B shows the noise imported into Eureqa’s spreadsheet to evolve a noise-producing function which could be used as a key.

---

4Noise was downloaded from Random.org which has generated 190.7 GiB (1 Gibibyte is ≈ 1.074GB) of random bits for the Internet community)
Chapter 4. Keys for Seeding Chaos

Block C converts the evolved function into an iterative noise-producing function using Block D, which is a delay function. Block E converts the noise into pseudo-random sequences (pseudo-random because of digital finite state arithmetic). The Eureqa user-interface (UI) in Figure 4.7 shows a function evolving in the top left pane from input noise data.

![Figure 4.7: Eureqa evolving a new function from noise.](image)

Eureqa required almost a day of processing using an Intel(R)-Core(TM) i3 CPU M330 @2.13 GHz processor. The FIT column changes colour from red through orange to green as a new function evolves. The evolutionary progress is displayed as an error function versus complexity in the bottom right pane. The evolving signal in the top right pane is shown superimposed on the input natural noise data.

4.3.2 Processing Evolved Functions in PSpice

The evolved functions were post-processed to produce an iterative function using a delay in an iterative loop. The nomenclature for the noise function is $x_n$ for the present value and $x_{nd}$ for the delayed value. The delay in digital signal processing (DSP) is $x(n)$ and $x(n-1)$ [Tobin, 2007c]. The evolved noise function is:

$$a \cdot \cos(b \cdot x_{nd}^2 + \tan(c \cdot x_{nd}) + \sin(d \cdot x_{nd} + \cos(e + f \cdot x_{nd}^2))) - g \cdot \cos(h \cdot x_{nd}^2)$$ (4.3)

Equating $x_n$ to equation (4.3) produces the iterative nonlinear equation in (4.4).

$$x_n = a \cdot \cos(b \cdot x_{nd}^2 + \tan(c \cdot x_{nd}) + \sin(d \cdot x_{nd} + \cos(e + f \cdot x_{nd}^2))) - g \cdot \cos(h \cdot x_{nd}^2)$$ (4.4)
The newly-evolved key function parameters are: $a = 0.767, b = 0.465, c = 0.327, d = 2.353, e = 0.9189, f = 0.465, g = 27.8, h = 4.463$. Figure 4.8 (a) shows the schematic with a delay and a PSpice ABM1 part containing the new post-processed multivariate iterative function equation (4.4).

\[
a \cos(b \cdot v(xnd)^{2}) + \tan(c \cdot v(xnd)) + \sin(d \cdot v(xnd)^{2}) + \cos(e \cdot f \cdot v(xnd)^{2})) - g \cos(h \cdot v(xnd)^{2})
\]

The delay part developed by the author (see Chapter 7) connects the ABM1 part output to the input forming an iterative loop, where the equation coefficients and delay parameter are defined in a PSpice PARAM part. A PSpice LIMIT part in the loop was an important discovery as it allowed the simulation to continue, otherwise, the simulation would have stopped because of convergence problems. Figure 4.8 (b) shows an 8-bit ADC connected as a 1-bit ADC to the loop and converts the analogue noise signal to binary digits. The binary sequences are written to a text file using a PSpice VECTOR1 part attached as shown for statistical analysis at a later stage. The top left pane in Figure 4.9 (a) shows the random binary sequence produced by the evolved function.

**Figure 4.8:** (a) Simulating the evolved noise function using PSpice ABM parts (b) 1-bit ADC producing PRBS from the noise function.

**Figure 4.9:** (a) Eureqa random binary sequence (b) Eureqa noise Spectrum (c) Eureqa Noise (d) A noise attractor ($x_n$ vs $x_{nd}$).
Chapter 4. Keys for Seeding Chaos

The evolved noise signal and corresponding wideband spectrum are plotted in the bottom left pane in Figure 4.9 (b) and (c). The noise attractor in Figure 4.9 (d) is a plot of \( x_n \) vs \( x_{n-1} \) and is similar to a true noise attractor showing no correlation between past and present values.

4.3.3 Limitations of Eureqa

The Eureqa method for seeding chaos produced PRBS and has many applications in cryptography. This method, however, was deemed not suitable for the local encoding applications outlined in Chapter 2 for the following reasons:

- Generates pseudo-random sequences because of finite state arithmetic,
- Costs are levied when Eureqa is used commercially, and
- Overall lack of control for generating noise functions.

In the next section, a radial basis neural network was investigated as an alternative method for overcoming some of the limitations listed.

4.3.4 Generating Noise Functions using Neural Networks

A Radial Basis neural network (NN) was researched for evolving noise-producing functions in a similar fashion to Eureqa, and the results were published [Blackledge, Bezobrazov, and Tobin, 2015]. Nevertheless, this technique didn’t address the problem of finite arithmetic but did overcome the last two reasons listed above. Although not used in the present prototype encoder, one solution for overcoming finite state arithmetic is to implement the NN using an analogue design comprising memristance as the NN variable weight (see Chapter 5).

4.3.5 Final Choice of Seed

Two methods were investigated for creating keys to seed the prototype chaos oscillators. The Eureqa and ANN methods produced PRBS from a digital platform and have valid applications but were not chosen because of the reasons stated previously. The prototype used sampled electronic noise from a data receiver and hence qualified the generated sequences as TRBS.

\(^5\) Attractors are discussed in Chapter 3
4.4 Chapter Conclusion

Two methods for creating keys for seeding chaos were discussed in this chapter. The first method used audio noise from a data receiver which was sampled randomly to prevent successful cryptanalysis using an external periodic signal injected into the data receiver from adversaries corrupting the initialising random noise. The second method applied natural noise to Eureqa to generate a noise-producing function which can be used as a source of random binary signals, or as a seed for chaos systems but this method generated PRBS only. Both techniques have applications in cryptography, but only the first method was used in the prototype for encoding sensitive medical and legal data as explained in Chapter 2. Some of the problems associated with using Eureqa as a noise seed source for chaos systems were addressed by researching and creating, a Radial basis ANN, but again this method only generated PRBS.
5 Methods for Generating OTPs

“Radio has no future. Heavier-than-air flying machines are impossible. X-rays will prove to be a hoax”. William Thomson, Lord Kelvin British scientist

5.1 Chapter Overview

Two further techniques were investigated for generating OTPs. The first technique employs a new circuit device called memristance, the missing circuit element proposed in 1971 by Professor Leon Chua [Chua, 1971]. Several models of memristance were simulated in PSpice and added to the Chua chaos oscillator investigated in Chapter 3 to generate OTP sequences. An emulated memristor built from standard circuit components, replaced the negative resistance, ‘Chua diode’, in a Chua chaos oscillator [Muthuswamy and Kokate, 2009]. The second technique employs a fractional-order (FO) Lorenz chaos oscillator system to generate OTP and is similar to the method used in Chapter 3.

5.2 Chua Memristance

The standard Chua oscillator presented in Chapter 3 has a segmented negative resistance “Chua diode” as part of a tuned circuit to generate chaos and subsequently, generate random binary sequences. In this chapter, a memristor (a portmanteau of memory-resistor) circuit element is introduced which has a negative cubic characteristic to replace the segmented negative Chua diode. A company called Knowm, manufactured memristors in mid-2017 costing two hundred euros for eight memristors in a dual-in-line (DIL) package ¹. This is expensive but with commercial applications, it will become affordable [Tobin and Blackledge, 2014]. Consequently, an emulated memristor was constructed from standard electronic devices for generating random binary sequences.

¹https://knowm.org/product/bs-af-w-memristors/
Chapter 5. Methods for Generating OTPs

5.2.1 The Memristance Model

Figure 5.1 illustrates fundamental equations and relationships between the six possible combinations between charge, current, voltage and flux in capacitance, inductance and resistance. For these circuit elements, there is no branch which connects charge with flux, and this represents a missing circuit element, memristance, $M$, as deduced by Chua.

![Diagram of circuit elements](image)

**Figure 5.1**: Relationships between charge, $q$, and flux $\phi$.

The term flux in memristor technology is not the same as magnetic ‘flux linkages’ in a coil but Chua used flux linkages as the time integral of the voltage across the memristance and charge $q$ (time integral of current), to define memristance as:

$$M = \frac{v}{i} = \frac{\frac{d\phi}{dt}}{\frac{dq}{dt}} = \frac{d\phi}{dq} \Omega \quad (5.1)$$

The relationship between this ‘flux’ and the integral of the memristance voltage is:

$$\phi(t) = \int_0^t v(\tau)d\tau \quad (5.2)$$

Similarly, charge, $q$, is the integral of the memristor current:

$$q(t) = \int_0^t i(\tau)d\tau \quad (5.3)$$

Charge in one direction increases memristance, but decreases if in the opposite direction. Current is the rate of change of charge with time, so, differentiating:

$$v(t) = \frac{d\phi}{dt} = \frac{d\phi}{dq} \frac{dq}{dt} = M(q) \frac{dq}{dt} = M(q)i(t) \quad (5.4)$$
Chapter 5. Methods for Generating OTPs

This shows memristance is a function of the charge passing through it, \(\phi = f(q)\) [Biolek, Biolek, and Kolka, 2011], [Itoh and Chua, 2014], [Biolek, Biolek, and Biolkova, 2009]. Removing the applied voltage stops the flow of charge but the resistive state is remembered. In 2008, a Hewlett-Packard (HP) team led by Stan Williams developed a memristor-type device with a modest switching speed performance [Williams, 2008]. They replaced this device in 2010 with a 3 nm by 3 nm memristor operating at one nanosecond, with an operating speed of approximately 1 GHz, and electron/hole mobility of 1 m/s. This was used in a USB memory key with a performance similar to flash memory technology but which stores information as a resistive state. However, external power is not required to refresh the data stored and so there is nothing to leak off resulting in zero power dissipation.

5.2.2 Memristance Structure

The HP memristance with 50 nm of titanium dioxide \((TiO_2)\) is sandwiched between two 5 nm platinum plates as shown in Figure 5.2.

\[\text{Figure 5.2: (a) Doped-undoped regions (b) On/Off regions represented as two potentiometers.}\]

The high resistance titanium dioxide semiconductor is doped with oxygen atoms which are heated to remove negatively-charged oxygen atoms \((TiO_{2-x}, \text{with} \ x = 0.05)\) to create positively-charged oxygen vacancies \(V_0^+\) as charge carriers \(^3\). The polarity of the DC voltage moves the negatively-charged oxygen deficiencies from the doped region into the undoped region reducing the resistance. The memristor becomes more resistive by reversing the polarity of the DC source, but the oxygen vacancies stay in place when the bias is removed, making it a non-volatile memory.

\(^2\)This is different to storing charge in a capacitor which requires energy for storage

\(^3\)The Knowm company manufactured memristors in July 2017 and uses a pinched eight lying on its side symbol
Chapter 5. Methods for Generating OTPs

5.3 The Linear-Drift Memristance Model

Memristance was simulated using ABM parts and electronic model parts [Meuffels and Soni, 2012], [Zhong, 1994], [Joglekar and Wolf, 2009] and the models were created from equation (5.4) relating flux and charge. The relationships between device width, \( D = 10 \) nm, memristance, and \( w(t) \) (variable width of the doped region depends on the applied voltage polarity across the end plates) is now considered. The \( \text{on} \) resistance, \( R_{\text{on}} \) is when the doped region expands to the full length, and \( w = D = 10 \) nm. The \( \text{off} \) resistance, \( R_{\text{off}} \) occurs when \( w(t) = 0 \). The ratio of \( \text{off} \) to \( \text{on} \) resistance typically is \( 10^2 \) to \( 10^3 \). The normalised width of the doped region with respect to \( D \), is \( x \):

\[
x = \begin{cases} 
0 \leq (w/D) \leq 1 \\
0 \text{ elsewhere}
\end{cases}
\]  

(5.5)

Memristance is the resistance of the doped and undoped regions [Strukov et al., 2008] defined as:

\[
M(t) = \frac{w(t)}{D} R_{\text{ON}} + \left(1 - \frac{w(t)}{D}\right) R_{\text{OFF}}
\]  

(5.6)

Or, in terms of \( x \):

\[
M(x) = R_{\text{OFF}} - (R_{\text{OFF}} - R_{\text{ON}})x \approx R_{\text{OFF}}(1 - x)
\]  

(5.7)

Equation 5.7 assumes that the \( \text{on} \) resistance is much less than the \( \text{off} \) resistance. The doped region width, \( w(t) \), is proportional to the charge, \( q(t) \) flowing in the device, where the average dopant mobility is \( \mu_v \approx 10^{-14} \text{m}^{-2}\text{s}^{-1}\text{V}^{-1} \), and the width is:

\[
w(t) = \frac{\mu_v R_{\text{on}}}{D} q(t)
\]  

(5.8)

The speed of the boundary layer between the doped and undoped regions in the simple linear-drift model is constant and is the rate of change of \( w(t) \) with time making the derivative \( w(t) \) a function of the current:

\[
\frac{dw(t)}{dt} = \frac{\mu_v}{D} R_{\text{on}} \frac{dq(t)}{dt} = \frac{\mu_v}{D} R_{\text{on}} i(t)
\]  

(5.9)

Substituting (5.9) into (5.7), yields:

\[
M(q) = R_{\text{OFF}} \left\{ 1 - \frac{\mu_v R_{\text{on}}}{D^2} q(t) \right\} = R_{\text{OFF}} \left\{ 1 - \frac{R_{\text{on}}}{\beta} q(t) \right\}
\]  

(5.10)

where \( \beta = D^2 / \mu_v \), where the \( D^2 \) factor has a greater effect in the nanoscale range.
5.4 Hewlett-Packard Memristor Model

The HP sub-circuit model by Biolek represents the nonlinear doping drift model [Biolek, Biolek, and Biolkova, 2009]. In this model, the initial on and off resistance parameters are defined in a header file in the subcircuit. Increasing the input voltage increases the current but in a nonlinear fashion. An increase in the memristor current decreases the resistance even further and this increases the current, but at an increasing rate. The current continues to rise even when the input voltage maximum is reached because the resistance is still increasing. When the current reaches a maximum it will not return on the same path and will form a loop. The effect of current feedback flowing in the device is represented by a feedback term $v(m)(1+v(q)^2)$ placed in an ABM1 part in Figure 5.3 and applying the voltage at the integrator (INTEG part) output $v(q)$ to the ABM1 part.

5.4.1 Hysteresis and Memory in the Memristor

The ABM schematic signals are shown in Figure 5.4.
Hysteresis means to lag behind and the present state of the memristor depends on its past state, giving the device memory because the oxygen vacancies do not migrate back to their original positions in the lattice. It can be shown that the hysteresis loop shrinks when the excitation frequency is increased and the memristor will assume a linear resistance characteristic [Chua and Kang, 1976]. The current and voltage are zero at the origin and indicates no power is dissipated at this point, hence the name pinched hysteresis loop. The total resistance change depends on the length of time the voltage is applied. Applying a negative voltage causes the resistance to increase with time to a maximum value, but a positive voltage causes the resistance to reach its minimum value. Removing the voltage leaves the device with its resistance frozen in time but can be reset by applying a voltage.

5.4.2 The nonlinear Memristor Model

Small voltages applied to nanoscale devices produce a large field intensity (V/M) resulting in a nonlinearity in the ion transport mechanism [Strukov et al., 2008]. The speed of the boundary region between the doped and undoped areas decreases gradually to zero at either end in a nonlinear way. The linear-drift model fails to account for this nonlinearity and Joglekar and Wolf [Joglekar and Wolf, 2009] proposed a window function defined: \( f(x) = \{1 - (2x - 1)^2p\} \). The memristor nonlinear dopant drift SPICE model window has a positive factor, \( p \in [1, 10] \) entered into the ABM1 part in the inset in Figure 5.5 [Biolek, Biolek, and Biolkova, 2009]. The window is simulated using a \texttt{vpulse} part configured as a sawtooth signal applied to the ABM part. Figure 5.5 (a) shows plots for \( p = 1, 4, 10^4 \). Setting \( P \) to 10 in the HP model produced the bow-tie display in Figure 5.5 (b).

**Figure 5.5:** (a) Three Window functions (b) Bow-tie for \( P = 10 \).

\(^4\)Here time represents the normalised width of the memristor, \( x = \frac{w}{D} \)
Figure 5.6 shows the memristance symbol created during the research which has a HP sub-circuit model attached where the black band is the negative polarity end.

![Figure 5.6: Memristor symbol containing an HP sub-circuit model.](image)

### 5.4.3 Memristance Emulation

Figure 5.7 shows an electronic memristance emulator constructed from standard electronic components [Muthuswamy, 2010].

![Figure 5.7: A memristance emulator from electronic parts.](image)

Memristance has a negative cubic characteristic implemented using AD633 multiplier devices [Zhong, 1994] producing a charge-flux cubic relationship characteristic written as: $q(\phi) = \alpha \phi + \beta \phi^3$. The emulated memristor cubic characteristic in Figure 5.8 has roots at $\pm 3.14$ V [Zhong, 1994].

![Figure 5.8: A cubic function with three real roots at y = 0.](image)
Chapter 5. Methods for Generating OTPs

The voltage across the 100 Ω resistor is the y-axis input current with the DC voltage swept from -5 V to + 5 V as the x-axis.

5.4.4 The Chua Chaos Memristor Oscillator

Figure 5.9 shows the standard Chua oscillator discussed in Chapter 3 but with the segmented nonlinear negative resistance - the ‘Chua diode’, replaced with a memristor emulator [Zhang, Zhang, and Zhang, 2009]. Figure 5.10 shows the Chua chaos oscillator with the Chua diode replaced with a memristance emulator.

Inductors are difficult to work with and can produce problems. Because of this, the inductor was replaced with a Telegen gyrator circuit design [Itoh and Chua, 2008]. Analysis of this gyrator is examined in Appendix C.

The mechanism for producing random binary sequences uses the same design in Chapter 4, Page 67, Section 4.3.2.
Chapter 5. Methods for Generating OTPs

The memristor-Chua capacitor voltage is plotted in Figure 5.11.

![Figure 5.11: The Chua memristor oscillator voltage across C2.](image)

The two FPs were chosen as the threshold levels for producing OTP sequences. The Chua memristor chaos oscillator strange attractor in Figure 5.12 has a different shape to the Lorenz and Chua attractors plotted in Chapter 3.

![Figure 5.12: Memristor-Chua oscillator attractor v(C2) vs. vC1.](image)

5.4.5 Memristance and Cryptography

A random binary sequence generator was constructed using a memristance emulator attached to the Chua oscillator. However, initial results show that this configuration is not as effective as the chaos oscillator with delay feedback discussed in Chapter 3. Another technique for generating TRBS is to use memristance connected as the variable weight in an artificial discrete-time cellular neural network, [Itoh and
Chapter 5. Methods for Generating OTPs

Chua, 2014, [Buscarino et al., 2012], [Loverro, 2004], [Kozma, Pino, and Pazienza, 2012]. Widrow’s five three-terminal ‘memistor’ was the variable weight in the ADA-LINE (adaptive linear neuron) artificial neural which was controlled by integrating the current into the third pin of the memistor [Widrow, 1960].

5.5 Fractional-Order Chaos

The concept of FO calculus was created independently by Newton and Leibniz. In a letter from L’Hopital to Leibnitz on Sept 30th, 1695, he posed the question, “if the nth derivative of a function was replaced by \( n = 0.5 \), what would be the result”? [Loverro, 2004], [Ortigueira, 2008]. This area of FO calculus was developed by Grunwald, Letnikov, Riemann and Leibnitz, and has attracted interest in science and engineering disciplines [Chen, 2010]. In this chapter, FO calculus was used to create an FO Lorenz chaos oscillator thresholded to produce OTP. Preliminary tests on this oscillator, with an order less than three, show it produced OTP sequences with excellent entropy.

5.5.1 A Fractional-Order Lorenz Chaos Oscillator

In cryptography, integer chaos systems are easier to cryptanalyse because FO chaos systems state variables are theoretically infinite. The effect of FO on the spectrum and attractor was investigated by incrementing \( a \), the order of the complex frequency variable \( s \) between 0 and 1 in increments of 0.1. However, the circuit failed to oscillate for values of \( a \) greater than 0.5 until it reached unity. Fractional integral calculus applied the \( s \)-domain in the Laplace Transform:

\[
L\{f(t)\} = \int_0^\infty e^{-st}f(t)dt
\] (5.11)

The reactance of an FO inductive reactance, \( L \), is \( L\{\frac{df(t)}{dt^a}\} = s^a L\{f(t)\} \). The reactance of a FO capacitor, \( C \), of order \( a \) is \( X(s) = \frac{1}{s^a C} \) with a phase shift, \( a\pi/2 \), for a sinusoidal input and is 90\(^o\) for unity order. A generalised low-pass transfer function in \( s \) is:

\[
F(s) = \frac{V_{out}(s)}{V_{in}(s)} = \left(1 + \frac{s}{p}\right)^a \approx \frac{\prod_{i=0}^{n-1} \left(1 + \frac{s}{z_i}\right)}{\prod_{i=0}^{n} \left(1 + \frac{s}{p_i}\right)}
\] (5.12)

\(^5\)Professor Bernard Widrow proposed a device called a memistor similar to Chua’s memristance [Widrow and Angell, 1962]
Chapter 5. Methods for Generating OTPs

Table 5.1 shows linear transfer function approximations for a range of fractional orders which can be used to implement FO systems in electronics.

<table>
<thead>
<tr>
<th>a</th>
<th>N</th>
<th>$H(s)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>2</td>
<td>$\frac{1584.89(s+0.166)(s+27.83)}{(s+0.1)(s+16.68)(s+2783)}$</td>
</tr>
<tr>
<td>0.2</td>
<td>2</td>
<td>$\frac{79.43(s+0.0562)(s+1)(s+17.78)}{(s+0.031)(s+0.562)(s+10)(s+177.8)}$</td>
</tr>
<tr>
<td>0.3</td>
<td>4</td>
<td>$\frac{39.81(s+0.041)(s+0.372)(s+3.34)(s+29.94)}{(s+0.015)(s+0.1)(s+0.631)(s+3.981)(s+25.12)(s+158.5)}$</td>
</tr>
<tr>
<td>0.4</td>
<td>5</td>
<td>$\frac{35.48(s+0.038)(s+0.261)(s+1.778)(s+12.12)(s+82.54)}{(s+0.0177)(s+0.121)(s+0.825)(s+5.62)(s+38.31)(s+261)}$</td>
</tr>
<tr>
<td>0.5</td>
<td>5</td>
<td>$\frac{15.849(s+0.039)(s+0.2512)(s+1.585)(s+10)(s+63.1)}{(s+0.015)(s+0.1)(s+0.631)(s+3.981)(s+25.12)(s+158.5)}$</td>
</tr>
<tr>
<td>0.6</td>
<td>5</td>
<td>$\frac{10.79(s+0.046)(s+0.3162)(s+2.154)(s+14.68)(s+100)}{(s+0.0146)(s+0.1)(s+0.681)(s+4.642)(s+31.62)(s+215.4)}$</td>
</tr>
<tr>
<td>0.7</td>
<td>5</td>
<td>$\frac{9.363(s+0.0645)(s+0.578)(s+5.179)(s+46.42)(s+416)}{(s+0.0139)(s+0.124)(s+1.116)(s+10)(s+89.62)(s+803.1)}$</td>
</tr>
<tr>
<td>0.8</td>
<td>4</td>
<td>$\frac{5.308(s+0.133)(s+2.371)(s+42.17)(s+749.9)}{(s+0.0133)(s+0.237)(s+4.217)(s+749.9)(s+1334)}$</td>
</tr>
<tr>
<td>0.9</td>
<td>2</td>
<td>$\frac{2.267(s+1.292)(s+215.4)}{(s+0.013)(s+2.154)(s+359.4)}$</td>
</tr>
</tbody>
</table>

The ABM FO Lorenz chaos oscillator in Figure 5.13 shows a PSpice Laplace part for FO integration, where the order of the complex frequency variable $s$ was varied between 0 and 1. It was necessary to introduce a gain parameter, $Q$ to increase the loop gain in the second coupled circuit for sustaining chaos oscillations. The FO $x$ signal was thresholded using the design explained in Chapter 3 and produced random binary sequences. This chaos system produces chaos even when the order is less than three. According to the Poincaré-Bendixson (PB) theorem, generating chaos requires a third-order chaos system for the autonomous Lorenz system [Cafagna, 2007], so perhaps this theorem may have to be redefined.
Chapter 5. Methods for Generating OTPs

Figure 5.13: An ABM FO Lorenz circuit with thresholding.
Chapter 5. Methods for Generating OTPs

Figure 5.14 shows PSpice signals from the FO Lorenz chaos system. The top pane shows constant-width digital set and reset signals combined in an exclusive-OR gate to generate OTP sequences.

![Figure 5.14: (a) Output from an FO Lorenz system (b) Biased X signal with set (green) and reset (red) pulse superimposed.](image)

The bottom pane shows the analogue set and reset output signals superimposed on the Lorenz X signal to which a DC bias was added. The attractor for the ABM FO Lorenz system is plotted in Figure 5.15.

![Figure 5.15: (a) FO Lorenz attractor with set and reset signals.](image)
Chapter 5. Methods for Generating OTPs

The analogue set and reset pulse sequence trajectories are superimposed to show how they line up at each FP. A practical Lorenz FO chaos system used the summing integrator in Figure 5.16 [Jia et al., 2016].

\[ x = -\text{Int}(F(x-y)) \]

Note: because of time constraints NIST results were not been obtained for the FO system but will be considered in future research.

5.6 Chapter Conclusion

The nonlinear Memristance charge-dependent resistance with memory was simulated using several memristance models. Expensive commercial memristance became available mid-2017, but the research here used a circuit memristance emulator constructed from standard electronic components. This emulator replaced the Chua diode in the nonlinear Chua oscillator for generating OTPs Random binary sequences, but current results show it may not be as effective as the chaos oscillator with delay feedback, investigated in Chapter 3. There are many potential applications for memristance in the area of cryptography [Merrikh-Bayat and Shouraki, 2011] and future research will advance when the new 16-pin DIL memristor devices become affordable. An FO Lorenz chaos oscillator with thresholding was simulated for a range of orders and produced random OTP sequences which were not subjected to NIST tests at the time of writing the thesis but is reserved for future work.
6 Chaos Maps

“Deviner avant de démontrer! Ai-je besoin de rappeler que c’est ainsi que se sont faites toutes les découvertes importantes. Guessing before proving! Need I remind you that it is so that all important discoveries have been made. The scientist does not study nature because it is useful; he studies it because he delights in it, and he delights in it because it is beautiful. If nature were not beautiful, it would not be worth knowing, and if nature were not worth knowing, life would not be worth living”, Jules Henri Poincaré

6.1 Chapter Overview

This chapter examines the Logistic and Hénon chaos maps selected because of their high LD value in Table 8.1 in page 127. Chaos maps operate in sampled time and are modelled using difference equations [Oishi and Inoue, 1982]. Two techniques for generating delays in the iterative map loop were researched; a standard sample-and-hold delay, and a novel non-sampling Padé analogue delay technique examined in Chapter 3. The discrete uni-modal nonlinear dynamic system (NDS) Logistic map can model growth and death in population dynamics, and the discussion introduces new simulation techniques and novel delay parts developed during this research.

Random binary sequences are produced in chaos maps for certain map parameter values. To determine these parameters for the Logistic map, for example, requires plotting the output for a range of growth parameters. This is called a bifurcation plot and during the research, it was discovered that the pitchfork bifurcation shape, as presented in chaos literature, was not achieved, but instead was tangential in shape and this is discussed here. Methods for controlling a stability problem associated with initialising voltages were investigated. The last part of this chapter investigated Eureqa (see Chapter 4 Page 66) for generating keys for initialising continuous chaos systems. The research also investigated how Eureqa evolved new chaos maps from chaos simulation data exported from PSpice.
Chapter 6. Chaos Maps

6.2 The Continuous Logistic Equation

The continuous model by the Belgian Pierre Verhulst [Nåsell, 2001], is based on the work by the philosopher, Malthus and defined by equation (6.1):

\[
\frac{dy}{dt} = Ry(1 - y) \tag{6.1}
\]

The ABM schematic in Figure 6.1 includes a sigmoidal function model solution to the Logistic map, \( f(k) = 1/(1 - ae^{-kt}) \).

The growth factor, \( R \), is represented by an ABM ramp generator part where the triangle-shaped integrator part solves the equation. Sweeping the input voltage, \( V_{\text{sweep}} \), over a voltage range produced the sigmoidal characteristic in Figure 6.2 (a) which grows exponentially but flattens out to unity.

**Figure 6.1:** (a) The continuous Logistic equation schematic (b) The Logistic map solution (c) A growth factor generator sweep part.

**Figure 6.2:** (a) Sigmoidal function and integrator output (b) Continuous Logistic orbit plot for \( R = 4 \).
Chapter 6. Chaos Maps

The sigmoidal characteristic from the PSpice ABM1 part is similar to the output characteristic, $v(yn)$ from the integrator. The continuous Logistic trajectory in Figure 6.2 (b) is called an orbit diagram and defines the path of a point starting from an initialising voltage. The parabolic trajectory for $R = 4$ is plotted by changing the horizontal time-axis to $v(yn)$. Plotting $v(yn)$ also produces an identity line which intersects the orbit trajectory $v(int\_in)$ at a fixed point (FP) equal to 0.75 V.

6.2.1 The Discrete Logistic Map

Verhulst’s first-order differential equation population growth model is continuous in time [Ciecka, 2012] and cannot model population growth which relies on discrete factors such as food, animal fertility. The mathematician, Robert May, developed the realistic Logistic map equation population growth model [May, 1976], [May, 2004]. The digital Logistic map equation is derived from the continuous Logistic equation (6.1) using finite difference equations with a time increment, $\Delta t$, to yield:

$$\frac{\Delta y}{\Delta t} = \frac{y_n - y_{nd}}{\Delta t} = ky_n(1 - y_{nd}) \Rightarrow y_n - y_{nd} = \Delta t ky_{nd}(1 - y_{nd})$$

(6.2)

Thus:

$$y_n = (1 + k\Delta t)y_{nd}(1 - y_{nd}) = R y_{nd}(1 - y_{nd})$$

(6.3)

The growth factor, $R = (1 + k\Delta t)$ has a range $R \in [0, 4]$. and the model has a growth rate factor $R y_{nd}$ proportional to a growth-limiting factor, $(1-y_{nd})$ which keeps the population from growing exponentially. As the population approaches a maximum it is multiplied by this factor and keeps the population in check.

6.2.2 The Predator-Prey Population Model

The Logistic map models predator-prey population such as cats and mice in a sealed house from which it is impossible to escape [Kocarev and Jakimoski, 2001]. The cats eat the mice but then have no food which causes them to die off producing another growth spurt in the mice population. In the discrete predator-prey model, $y_n$ represents the population and is a function of the previous year’s population, $y_{nd} \in [0, 1]$ (maximum value of one because it is the ratio of the past population to the maximum sustainable population).

---

1The preferred notation for a delay is a $d$ subscript rather than the standard DSP delay nomenclature $x(n-1)$ [Tobin, 2007c]
The recursive Logistic map schematic in Figure 6.3 has the output connected to the input via a customised analogue delay developed during the research.

During the research, a LIMIT ABM part was placed in the feedback path and allowed the simulation to continue over the complete growth factor range. This simple important discovery prevented the system from going to infinity but without affecting the model dynamics. The maximum and minimum excursions of the LIMIT part can be adjusted but it is important that these limits are not less than the natural signal excursion of the map, otherwise, the dynamics of the model will change.

### 6.2.3 The Logistic Orbit Diagram

The orbit diagram in Figure 6.4 (a) is a plot of the population, $y_n$, against the past population $y_{nd}$, and the orthogonal $z$-axis is time.

The parabola height changes with $R$ and has a maximum value of $R/4$ but the width remains constant. The Logistic orbit plot has a maximum value of 0.25 for $R = 0.5$. 

---

**Figure 6.3:** The LIMIT part for solving convergence problems at $R=3.57$ and delay = 100 us.

**Figure 6.4:** (a) The Logistic map $y(0) = 0.2 V$, $R = 2.8$ (b) 3-D Logistic.
Chapter 6. Chaos Maps

The FP is read from the intersection of the orbit trajectory and the identity line. PSpice cannot plot 3-D diagrams and data was exported from Probe to Matlab to plot the 3-D Logistic map orbit map in Figure 6.4 (b). The Logistic map trajectory stretches due to $Ry_{nd}$, and the $1-y_{nd}$ factor is responsible for the trajectory folding back on itself forming a closed loop. Stretching and folding are necessary for chaos to exist [Hirsch, Smale, and Devaney, 2012] and ensures the system converges.

6.2.4 Logistic Map Fixed Points

There are two FPs displayed in Figure 6.4 (a): a stable FP where the identity line intersects the orbit trajectory, and an unstable FP at the origin because it tends to move away from the origin. For $R = 2.8$, the Logistic equation parameter, $y_n$, and the identity line, meet the parabola at $x = 0.642$, which is a stable FP because all trajectory points converge on it. FP values are obtained by equating the Logistic map function to itself: $f(y_n) = y_n$. This is proved by $y_{nd} = Ry_{nd}(1 - y_{nd})$ $\Rightarrow y_{nd} = 1 - 1/R = 1 - 1/2.8 = 0.642$.

6.3 Logistic Map Bifurcation

The complexity of the Logistic model was mainly investigated in PSpice but Figure 6.5 was plotted in Matlab. This is a bifurcation plot of the Logistic map signal which splits up into smaller self-similar fractals for a range $R$ values.

![Logistic Map Bifurcation Diagram](image)

FIGURE 6.5: Logistic map bifurcation diagram plotted in Matlab.

---

2 This is equivalent to a zero population growth where the derivative is zero.
A Logistic map bifurcation diagram plots the chaos variable with the horizontal time parameter replaced by the swept voltage, V(R), representing the growth factor and is discussed in Chapter 7 Page 7.3). To generate random binary sequences the Logistic map must operate in the chaos region $R \in [3.57 - 4]$. Bifurcation shows how a chaos system behaves when a system parameter changes the stability of an FP causing the trajectory to move from one quasi-stable state to another. The Logistic map eigenvalues determine the stability and are calculated setting $f(y_n) = y_n$. The one-dimensional Logistic system Jacobian matrix is the differential of the function at the FP.

The population dies for $R$ between 0 and 1, but approaches a value of $(1-1/R)$ for $R$ between 1 and 2. The population decays in an oscillatory manner for $R$ between 2 and 3 to a fixed value, $(1-1/R)$. A growth factor greater than three causes the population to oscillate between two values referred to as period-doubling. The population bifurcates to a high value one year and a low value the next year (it takes two iterations for the population to return to its original value). Similarly, the period doubles again to four when $R$ is 3.45, with a four-year cycle. At $R= 3.57\ldots$, a point called the accumulation point or Feigenbaum’s number) is reached where chaos starts and the population never settles to a fixed population value.

### 6.3.1 Stability Loss Delay

The stability of continuous chaos system is analysed in the $s$-plane, but discrete chaos maps apply the unit circle $z$-plane [Tobin, 2007c]. Bifurcation in chaos maps occurs whenever an FP loses stability. The map is stable if the pole (eigenvalue) locations are within the unit circle, but bounded-input bounded-output (BIBO) unstable if the poles are outside the unit circle and the map trajectory moves away from the FP at the origin. Poles on the unit circle lead to conditional stability, but if the poles cross the unit circle, then a bifurcation occurs. Classification of the bifurcation type depends on where a pole crosses the unit circle. A period bifurcation (also called a flip or fork bifurcation) occurs for $R$ between 3 and 4. A Hopf bifurcation starts oscillatory behaviour from a steady state value where the poles have zero real and complex conjugate imaginary parts. To plot a bifurcation diagram in PSpice requires changing the time-axis in Probe to a voltage that represents the swept variable. In this analysis, a saw-tooth voltage represents the change in the growth factor with time.
Chapter 6. Chaos Maps

The pitchfork shape at each bifurcation point was not observed after simulation but had "tangential distortion" shown in Figure 6.6 [Blackledge, Bezobrazov, and Tobin, 2015]. This is caused because the saw-tooth was swept too fast in time.

![Figure 6.6: (a) Period bifurcation plots (b) Tangential distortion.](image)

The distortion was caused by the slope of the ramp generator signal which is the rate of change of the parameter under investigation. It was discovered that setting the ramp time to a larger value will reduce this distortion but may produce simulation convergence problems and longer simulation times. This phenomenon is not observed in software bifurcation plots because 'FOR' loops allows the 'Simulation World' to pause while the parameter is changed. In the Real World, time does not stop, and the bifurcation shape from sweeping a parameter with time is tangential and not pitchfork.

The author presented a paper at a nonlinear mathematics conference in Dublin where this distortion was discussed [Tobin, Blackledge, and Bezobrasov, 2016]. Professor Vassili Gelfreich (Warwick University) suggested this could be stability loss delay (SLD), an important and interesting phenomenon not completely understood. SLD was first observed by M.A.Shishkova [Shishkova, 1973] under the supervision of Lev Pontryagin [Pontriagin and Rodygin, 1960], [Morris and Moss, 1986]. In a private discussion, Gelfreich referred to a paper by Anatoly Neishtadt in which SLD is examined [Neishtadt, 2009]. The behaviour of systems at the bifurcation region depends on parameter factors close to a critical bifurcation point. When a bifurcation parameter changes slowly over time and passes through a value which produces bifurcation, the system delays the bifurcation, hence the name. At the point where it bifurcates quite abruptly, the trajectory heads off at a tangent rather than the cusp bifurcation customarily observed. Gelfreich and Neyshtadt suggested adding small amounts of Gaussian noise to the model to correct the SLD. This was experimented in PSpice and produce some reduction in SLD. Another experiment
eliminated SLD and used a step generator which in a sense emulated a programming ‘FOR’ loop and is explained in Chapter 7 Page 122.

6.3.2 Period Three in Bifurcation Plots

James Yorke and T.Y. Li wrote an article “Period Three Implies Chaos” in a 1975 issue of American Mathematical Monthly. They stated if a period 3-cycle exists, then it implies cycles of every other period will exist too, and also chaotic cycles with no period at all [Li and Yorke, 1975]. There are Logistic map values of $R$ which produce ‘Period Three’ window regions called intermittencies, as shown in Figure 6.6 (a). For $R = 3.6786$, an odd-period cycle appears, which according to the Li-Yorke theorem, produces period three cycles in chaotic zones, but only after period-doubling bifurcations have ended [Saha and Strogatz, 1995]. The ‘Period Three’ window is observed by magnifying a portion of the attractor at specific values of the growth factor to show the same figtree pattern occurring again at each of the three lines - a fractal self-similarity phenomenon shown in Figure 6.6 (a) [Li and Yorke, 1975], [Saha and Strogatz, 1995].

Figure 6.6 (b) shows the ‘Period Three’ in greater detail around the region $R = 3.77$ to 3.84 to reveal the fractal nature of the bifurcation diagram. However, the presence of periodic window regions in a bifurcation diagram would make this region unsuitable for generating OTP because it is no longer deterministic and occurs for $R = 1 + \sqrt{8} \approx 3.83$, where the system oscillates between three values. This intermittency window shows a self-similar fractal pattern (a pattern that is scale-free), which can be observed in finer detail at each bifurcation point. The same pattern occurs irrespective of the scale.

6.3.3 Feigenbaum Constants

Mitchell Feigenbaum discovered the period-doubling mechanism for systems to became chaotic and showed relationships exist between recurring ratios in period-doubling called the Feigenbaum constants nearly as familiar as $\pi$ and $e$. Figure 6.7 shows how these constants are obtained from measurements on the plot. The Logistic map period-doubling first occurs for $R = 3$, the second at 3.455, and the third at 3.533, etc., [Briggs, 1991].

3Feigenbaum is German for figtree which has a fractal structure, and… a mathematician’s joke… [Stewart, 1997]
The second Feigenbaum constant was determined from a PSpice simulation by measuring the width of the bifurcation opening at a datum point of 0.5 V between successive bifurcation period-doubling shown.

\[ \delta = \lim_{y \to \infty} \frac{R_n - R_{n-1}}{R_{n+1} - R_n} = \frac{\text{Delta}_3}{\text{Delta}_4} = \frac{3.455 - 3}{3.553 - 3.455} = 4.64 \ldots \] (6.4)

This is close to the Feigenbaum constant, 4.6692\ldots [Briggs, 1991] but an exact value cannot be measured as it requires evaluating the limit \( n \) to infinity. The second Feigenbaum constant, \( \alpha = 2.5029\ldots \), is determined by adding a 0.5 V reference line to measure the bifurcation ‘opening’. This constant was derived from the ratio of the openings at each period-doubling at the same location on the bifurcation plot.

The value measured for this constant in Figure 6.7 is close to the theoretical value of 2.502907875 \ldots.

\[ \alpha = \lim_{y \to \infty} \frac{\delta_n}{\delta_{n+1}} = \frac{\text{Delta}_1}{\text{Delta}_2} = \frac{(854 \text{ mV} - 438 \text{ mV})}{(521 \text{ mV} - 359 \text{ mV})} = 2.54 \ldots \] (6.5)

The Feigenbaum constants apply to any NDS and can be explained using a mathematical process called renormalization [Stewart, 1997].
6.4 Logistic Map Sample-and-Hold Delay

The sample-and-hold (SH) delay use switches operated by anti-phase clock signals to move charge between two capacitors [Miliotis et al., 2009], [Suneel, 2006]. Figure 6.8 shows the ABM SH delay using SBREAK parts operated by out-of-phase clocks [Tobin, Paul, 2007], [Lakshmanan and Senthilkumar, 2011].

The stages are buffered with a unity gain amplifier to prevent loading.

6.4.1 Biphase Clock Delay

Figure 6.9 (a) shows anti-phase clock sequences operating the sampling switches to produce the delay shown in the bottom pane.

Figure 6.9 (b) shows a 10 us delay between peaks of the sine wave and the output from the second sampler. The AD633 device multiplies the growth factor with the...
growth-limiting factor, \((1-y_n)^4\). The ABM delay is replaced by LM4066 electronic analogue switches operated by anti-phase clock signals as shown in Figure 6.10.

The Logistic equation is implemented using four-quadrant multiplier parts, U1 and U2. Resistors R1 and R2 are connected to supply two signals to form the Logistic map equation. A CONST ABM part supplies the growth factor, \(R\). R3 and R4 ensure a small positive mV voltage drives the circuit in a positive direction to prevent the circuit from saturating towards the negative supply. A 555 digital clock is divided by two using a dual J-K flip-flop (FF) and divided again using a second FF. \(\bar{Q}(A)\), \(\bar{Q}(B)\) and the clock from the first J-K FF is applied to the AND gates to form the OTP.

### 6.5 A New Logistic Digital Map

A new implementation of the Logistic map is shown in Figure 6.11.

---

4the design of this circuit can be examined in Appendix B Page 172)
This is a new design developed during the research and contains fewer integrated circuits compared to the design just examined. The spectrum of this new map will not contain multiples of the sampling frequency because the signal is not sampled. The author speculates that sampling introduces a weakness in an encryption system because of the presence of sampling frequency multiples in the spectrum ad infinitum. Figure 6.12 (a) shows the signal from the new Logic map, with the spectrum in Figure 6.12 (b).

The attractor in Figure 6.12 (c) has a different shape to the Logistic orbit attractor discussed previously in Figure 6.4.

### 6.5.1 The Logistic Map in Cryptography

Claude Shannon suggested the stretching and folding mechanism in digital chaos maps such as the Logistic and Hénon maps, could be utilised in cryptography for generating random binary sequences [Ursulean, 2004]. A chaos map was first used in cryptography and appeared in a paper by Habatsu [Habatsu et al., 1991], and since then maps have been used in cryptography and communications [Sengupta and Andro, 2003], [Pressing, 1988]. Baptista used the Logistic map for encoding data and is described in a paper [Baptista, 1998], but his method was heavily criticised for its serious security flaws and for being slow [Arroyo, Alvarez, and Fernandez, 2008]. His method required a large number of iterations to encrypt a single character thus making it a slow encryption technique. Robert Matthews also used a modified Logistic map to generate OTPs [Matthews, 1989], as did James Kotwal [Kotwal, 2008].
Chapter 6. Chaos Maps

The Logistic map power spectral distribution (PSD) is not uniform, with more points distributed at both ends of the spectrum shown in Figure 6.13 which weakens its robustness for encoding.

![Logistic PSD](image)

**Figure 6.13**: The non-uniform Logistic map PSD.

6.5.2 Logistic Map Limitations

The Logistic equation is bounded in a region called the *basin of attraction*, with a range of zero to one, but unbounded and unstable outside this range. Simulating the Logistic map in PSpice can produce instability in the chaos region \(^5\) producing simulation convergence errors. In general, discrete chaotic maps are easier to analyse compared to continuous analogue chaos systems in Chapter 3 but are more challenging to implement electronically because of the time delays required [Rodriguez-Vazquez et al., 1987]. However, a new modified Logistic map was created which incorporates a Padé delay examined in Chapter 3 page 50, and is a simpler design with a reduced number of electronic components. Since sampling is not involved, then the robustness of the encryptors produced might be increased because the spectrum does not contain multiples of the sampling frequency. Initial conditions are important when implementing the Logistic equation in electronics because certain IC values are a source of convergence errors, both in PSpice and in the electronic circuit. These errors may give no output or an output that quickly saturates to the

---

\(^5\) The Logistic map becomes chaotic when \(R\) is greater than 3.57
Chapter 6. Chaos Maps

DC power supply. There are an infinite number of IC, \( y(0) \), that will not produce chaotic behaviour [Williams, 1997], [Sprott, 1997]. This is a serious problem when generating random binary sequences because the system will converge for some IC, but not for others. Setting IC = 1 uV for \( R = 3.99 \) produces a chaotic time series, but convergence errors will occur for IC = 1 mV, so the Logistic map IC must be chosen carefully as there an infinite number of unstable points and the choice of IC must meet the conditions: \( y(0) \neq 1 - 1/R \), and, \( y(0) \neq 0.5 \left\{ 1 \pm \sqrt{1 - 4/R} \right\} \). Thus, for \( R = 4 \), the IC, \( y(0) \), should not be set to 0.5 V or 0.75 V. These critical conditions make the Logistic map unsuitable for the encoder prototype unless a feedback mechanism is considered which increments the IC to another value when this problem occurs.

6.5.3 Difficulties with the Logistic Map

Difficulties using digital maps as an entropy source made it necessary to investigate ways of controlling the Logistic and Hénon maps. Three researchers in 1990 showed it was possible to control chaos, an idea thought impossible at the time A surprising discovery by Pecora-Carrol showed how chaotic systems could synchronise with each other [Carroll and Pecora, 1991]. This goes against the idea that the trajectory of two chaotic systems will diverge away from each with increasing time, even for identical chaotic systems. There are several techniques for controlling chaos, and the OGY method \(^6\) is now considered.

\(^6\)from the initials of Otto, Grebogi and Yorke
6.6 Controlling Chaos

Chaotic attractors in phase space have an infinite number of unstable periodic orbits (UPOs), and unstable steady states (USS). The OGY method stabilises one of the orbits using data from a Poincaré map. Changing a control parameter over a small region over time moves the trajectory away from a UPO onto a stable region. UPOs appear with almost periodic oscillations over short intervals and can be stabilised by making small adjustments to a system parameter when it is close to a particular UPO. The first step in the OGY method for continuous chaos systems is to plot a Poincaré section shown on the 3-D Lorenz attractor in Figure 6.14.

![Figure 6.14: Small parameter perturbation moves the Poincaré trajectory point towards P1 or P2, where stable and unstable directions are shown by arrows.](image)

6.6.1 Poincaré Section

A Poincaré section (see subsection (7.6)), is produced by sampling at regular intervals the state-space trajectory of the system. A forcing input function is applied to a chaotic system and measurements are made at the end of the forcing input signal cycle. This produces a map in which a period-one orbit appears at one point, a period-two orbit appears as two points, and so on. A point near a UPO is forced to move to an FP on the unstable periodic orbit by nudging a system parameter. However, the perturbation should be small enough so as not to change the dynamics of the chaotic system significantly.
Chapter 6. Chaos Maps

6.7 Controlling Digital Maps

It is necessary to control the digital Logistic map because of the problems with IC, where some values cannot be chosen. Also, there is an infinite number of UPOs which cause stability problems. If the Logistic map is to be incorporated in future encoder designs then these issues must be addressed. Figure 6.15 shows an analogue delay part developed to examine the FPs at each period-doubling.

![Figure 6.15: FPs of the Logistic map.](image)

6.8 Unstable Periodic Oscillations

The three UPOs are shown superimposed in Figure 6.16.

![Figure 6.16: Superimposing the three UPOs.](image)

The mechanism for controlling the Logistic map is shown in the ABM schematic in Figure 6.17.
Physical switches operate at a certain time to produce two different values for the growth factor, $R$. Switching it off for $R = 3.7$ uses a Sw_tOpen switch part at 10 ms, then switching it on for $R = 3.1$ using a Sw_tClose part simultaneously. Figure 6.18 shows the result for a step time equal to half the 50 μs delay time.

![Figure 6.18: Controlling the Logistic equation](image)

The output switches from a chaotic region to a limit cycle. The identity line on the orbit diagram shows the FPs ($1/R = 1/3.1 = 0.3226$, and $1/R = 1/3.7 = 0.2703$). Substituting these two values into the Logistic equation gives the FP as: $V(y_{nd}) = 0.3226 \times 3.5 \times (1 - 0.3226) = 0.677$ V. Repeating this for $R = 3.7$ gives $V(y_{nd}) = 0.2703 \times 3.7 \times (1 - 0.2703) = 0.7297$ V. This process can be automated by replacing switches with a decision mechanism to reduce the growth factor when the output is near an FP. Substituting $R = 3.9$ into the Logistic map equation 6.3 and plotting $y_n$ against $y_{nd}$ for shows a stable FP where the identity line crosses the parabola at $x = 0.74358$, for $f(x) = x$. Equating equation (6.3) to $y_{nd}$ yields $y_{nd} = R y_{nd} (1 - y_{nd})$. Thus, $y_{nd} = 1 - 1/R$
= 0.743589. There is a stable FP where the trajectory and the identity line intersect and an unstable fixed point at the origin. Period-one FP is at $y^*$ and occurs when $y_n = y_{nd}$. The FP is represented in Figure 6.19 by a horizontal line at 0.742589 Volts [Gauthier, 2003]. Figure 6.19 shows how the trajectory visits in a chaotic manner points quite close to the unstable FP, which in this case is a period-one UPO.

![Figure 6.19: Points close to the at FP = 0.7425 V.](image)

Figure 6.20 shows there are many UPO near the 0.743589 V horizontal line\(^7\).

![Figure 6.20: UPOs near the FP line = 0.743589 V.](image)

A few points from the trajectory are encircled to highlight their nearness to a UPO. In the phase plane, this is a sphere formed into an ellipsoid under the action of the dynamics of this method.

\(^7\)This plot was obtained by making the trajectory lines the same colour as the background.
Figure 6.21 is a schematic for controlling the Logistic map for \( R = 3.9 \) and the following analysis by [Gauthier, 2003] describes the control mechanism in the Logistic map by linearising around the desired UPO to stabilise the system near the unstable FP.

Small adjustments are made to the growth factor \( \delta R_n \) as: \( R_n = R_o \pm \delta R_n \), where the deviation is proportional to the distance from an unstable FP \( \delta R_n = -\gamma (y_n - y^*) \).

Equation (6.6) represents approximately the dynamics of a linear map when the trajectory, \( y_n \) is close to an FP, \( y^* \):

\[
y_n = y^* + \alpha (y_{n-1} - y^*) + \beta \delta R_n
\]

Linearising around a FP yields the eigenvalue of the Jacobian matrix:

\[
\alpha = \frac{\partial f(y, R)}{\partial y} \bigg|_{y=y^*} = \frac{R(1 - y^2)}{y^*} \bigg|_{y=y^*} = R(1 - 2y^*) |_{y=0.7436} = -1.9001
\]

The correction signal sensitivity is called the perturbation sensitivity:

\[
\beta = \frac{\partial f(y, R)}{\partial R} \bigg|_{y=y^*} = y^*(1 - y^*) |_{y=0.74358974358974} = 0.19066
\]

When \( y_n = y^* \) then \( dR_n = 0 \) and the control gain is calculated:

\[
\gamma = \frac{\alpha}{\beta} = \frac{-1.9001}{0.1907} = -9.9638
\]

The deviation from the FP is calculated \( y_n = y_n - y^* = y_n - 0.743589 \). The behaviour of the system near a FP is described by

\[
y_n = (\alpha + \beta \gamma)y_{nd}
\]
where the perturbation size is: $\Delta R_n = \beta \gamma y_n$. The no-control situation is $g = 0$ and means $y_n = a \ y_{n,d}$ and any input makes the FP unstable for $|a| \geq 1$. Applying a control signal will make an initial perturbation shrink when:

$$|\alpha + \beta \gamma| < 0$$  \hspace{1cm} (6.11)

Figure 6.22 shows the output from the Logistic map for $R = 3.9$.

![Graph showing output from Logistic map with control applied at $R = 3.9$.](image)

**Figure 6.22: Controlling the Logistic map at $R = 3.9$.**

This method has long delays while the desired orbit on the chaotic attractor approaches an FP and proportional control is applied. The perturbations, $dr_n$, is reduced once control is established. The control gain $g = -a/b = 9.9638$. 

Chapter 6. Chaos Maps

6.9 The Hénon Map

Michael Hénon, a French astronomer, investigated the complex trajectory of stars in a galaxy and applied a Poincaré section to simplify the complex 3-D picture to a simpler 2-D picture. The result was a Hénon chaos map and is a generalisation of the Logistic equation [Saha and Strogatz, 1995] which Strogatz said is similar to the Lorenz system involved stretching and folding [Strogatz et al., 1994]. However, the Hénon schematic in Figure 6.23 produces a strange attractor which does not form closed surfaces as observed in the Lorenz attractor. The Hénon discrete-time chaos system equation is

\[ x_n = 1 + y_{nd} - ax_{nd}^2 \]

where,

\[ y_{nd} = bx_{nd}^2 \]

giving:

\[ x_n = 1 + bx_{nd}^2 - ax_{nd}^2 \] (6.12)

The Hénon map coefficients, \( a \) and \( b \), control nonlinearity and dissipation. FPs are obtained by substituting partial derivative for each parameter in a two-dimensional Jacobian matrix. Let \( x_n = x_{nd} = x^* \), then:

\[ x^* = 1 - ax^2 + bx \Rightarrow ax^2 - (b - 1)x - 1 = 0 \]

The roots of the equation for \( a = 1.4, b = 0.3 \) (the original Hénon values), are:

\[ \frac{-(b - 1) \pm \sqrt{(b - 1)^2 - 4a}}{2a} = -1.1313 and 0.631 \] (6.13)

The Jacobian matrix linearises the Hénon map:

\[ J = \begin{pmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y} \end{pmatrix} = \begin{pmatrix} \frac{\partial x_n}{\partial x_n} & \frac{\partial x_n}{\partial y_n} \\ \frac{\partial y_n}{\partial x_n} & \frac{\partial y_n}{\partial y_n} \end{pmatrix} = \begin{pmatrix} -2ax & 1 \\ b & 0 \end{pmatrix} \] (6.14)

The magnitude of the determinant of the Jacobean matrix is \( b \).

\[ ^8 \text{If the } b \text{ coefficient is zero the the map reduces the Hénon map to a quadratic map which is a conjugate of the Logistic map} \]
6.9.1 Hénon Spectrum, Series and Attractor

The spectrum and time series for the Hénon maps in Figure 6.24 (a) and the attractor in 6.24 (b).

![Figure 6.24: (a) Hénon Spectrum (b) Hénon signal (c) Attractor.](image)

The identity line in Figure 6.24 (c) intersects the map at an FP of 631 mV but not all intersections are FPs.

6.10 Hénon Bifurcation

The ABM circuit in Figure 6.25 produces a Hopf bifurcation using a PSpice VPWL generator as the horizontal ‘a’ sweep signal.

![Figure 6.25: ABM schematic for plotting a Bifurcation diagram.](image)
Figure 6.26 is a Hénon bifurcation plot showing tangential distortion occurring at each bifurcation. This distortion was discussed in Subsection (6.3.1) and occurs whenever the parameter variation causing the bifurcation is swept too fast.

6.10.1 Disadvantages of Digital Maps

Digital maps have the potential for generating random binary sequences if the problems as discussed previously can be solved. Digital maps had a much lower LD value as shown in Table 8.1 compared to the analogue chaos oscillators. Furthermore, there are stability problem associated with certain IC values used in maps and this must be addressed before it can used in future encoders.

6.10.2 Evolving New Chaos Functions in Eureqa

The Eureqa EC software in Chapter 1 and Chapter 4 created noise-producing functions, but in this chapter new chaos functions were created by exporting PSpice simulation data from existing chaos systems to Eureqa. The example explained here simulated the one-dimensional chaotic Tent map and the simulation time-voltage data was exporte and paster to the Eureqa spreadsheet [Slezák and Dostál, 2014]. Eureqa evolved a function which approximated the original Tent map equation but

---

9A high value for the LD is desirable because the source entropy is proportional to it.
had no discontinuities unlike the Tent map. The new function folds back on itself (an endomorphism) similar to the Logistic map. The Tent map is defined over two regions in (6.15):

\[
f(x)|_{R=0.5} = x(n) = \begin{cases} 
R x(n-1) = 0.5 x(n-1) & \text{for } 0 \leq x \leq 0.5 \\
R - R x(n-1) = 0.5 - 0.5 x(n-1) & \text{for } 0.5 \leq x \leq 1 
\end{cases}
\]  

(6.15)

\(R \) is 0.5, where \(x \in [0,1] \). The Tent map circuit in Figure 6.27 (a) was initialised with noise from a voltage piece-wise linear (VPWL) ABM generator part \(^{10}\).

\[\text{FIGURE 6.27: (a) A Tent map schematic (b) The Eureqa evolved Tent map schematic.}\]

The evolved Tent map schematic in Figure 6.27 (b) is similar to the Tent map circuit in the left pane in Figure 6.28 (a).

\[\text{FIGURE 6.28: (a) Tent map signal (b) Tent map attractor (c) Evolved Tent map attractor.}\]

The two attractors from the original and evolved schematics are displayed in the right pane. Plotting \(x_{nd}\) on the y-axis creates an identity line which intersects the

\(^{10}\)The noise was created in Matlab\(^\circledR\) with the \texttt{rand} function to generate random numbers uniformly distributed in the interval (0, 1)
strange attractor plot of $x_n$ versus the delayed variable $x_{nd}$ at a FP. The two attractors are similar but the evolved Tent map attractor is also similar to a Gaussian map attractor, with rounding off at the apex because the process was stopped prematurely (Gaussian map in Appendix C).

6.11 Evolving Chaos Systems from Lorenz data

Michel Hénon introduced the Hénon map as a simplified model of the Poincaré section of the Lorenz model [Ivancevic and Ivancevic, 2007]. To verify this, PSpice simulation data from a Lorenz Poincaré section, which is a slice through a Lorenz 3-D strange attractor, was exported into Eureqa. As can be seen from Figure 6.29, a new Poincaré section was evolved which was similar to the original, and the evolved equation was also similar to the Hénon equation examined earlier in this Chapter.

![Figure 6.29: Eureqa evolving a Lorenz Poincaré map.](image)

6.12 Chapter Conclusion

The Logistic and Henón maps were researched as sources of entropy for the prototype. However, chaos sources for the prototype were selected based on results from the LD analysis, as explained in Chapter 3. The LD results show chaos maps didn’t perform as well as the analogue chaos oscillators, but it is possible with a new approach to controlling the maps, and using analogue delays, that this may change in future research. For example, the novel non-sampling delay Logistic-type map was
not tested in the present research but it is expected to produce a higher value of LD when compared to the delay chaos maps. The delay in digital maps is normally produced using an SH method. The Nyquist sampling theorem states that the sampling rate must be at least twice the highest frequency in the signal source but produces multiples of the sampling frequency in the spectrum ad infinitum. In the author’s opinion, this method might weaken the security of the encryptors because of these sampling frequency components.

There are stability problems using digital maps which require further investigation. Linearising the discrete system close to an FP produces the Jacobian from which the eigenvalues indicate the nature and type of the FP. This is important since bifurcation occurs when an FP loses stability. In discrete systems, the eigenvalues should remain inside the unit circle for stability but in nonlinear chaos systems, the type of bifurcations depends on where the eigenvalues cross the unit circle. The control signal stabilises near the FP when the condition in equation 6.11 is satisfied. Critical values of IC make chaos maps unattractive as an entropy source at this time since the present initialising method (the seed) is based on a random noise value. Future work will investigate feedback methods to overcome the problems of critical ICs.
7 PSpice Simulation of Chaos

“There’s no sense in being precise when you don’t even know what you are talking about.” John von Neumann

7.1 Chapter Overview

This chapter describes how Cadence® OrCAD PSpice, the world industrial standard for simulating mixed-analogue and digital linear electronic circuits, was re-appropriated by creating new parts, simulation meters, and techniques, to facilitate modelling nonlinear chaos circuits. PSpice simulation formed a large part of this thesis, allowing many new ideas and chaos circuits to be investigated in a realistic time frame. However, at an early stage of the research, PSpice presented many challenges simulating nonlinear chaos oscillators and maps, because convergence problems stopped the simulation. It was necessary to develop four novel meters, a staircase generator, and customised delay parts, to evaluate certain chaos phenomena for assessing thirty chaos circuits critiqued for maximum entropy during the prototype design. Furthermore, exporting the simulation data to software, permitted the evaluation and selection of suitable chaos generators for the prototype. Novel application of existing PSpice LIMIT and VECTOR parts also extended the simulation horizon of chaos.

7.2 New PSpice Parts

Analogue Behavioural Modelling (ABM) simulation produce fewer convergence problems compared to simulation with integrated circuit models. This high-level simulation is useful for proof-of-concept testing before component model simulation. Convergence problems encountered when simulating chaos circuits operating in the chaos region necessitated creating new techniques to fix this. New additions were created which extended the usefulness of chaos simulation. For example, a delay part was created for investigating Taken’s embedding technique examined in Chapter 3, as part of the evaluation and comparison of chaos systems researched in this thesis. The delay part was used in continuous and digital chaos systems and is
examined in the next section. Novel PSpice meters were created to investigate chaos phenomena such as bifurcation and for plotting Poincaré sections and return maps.

### 7.2.1 Novel Analogue Delay

Discrete recursive, one-dimensional quadratic chaos systems, have analogue delays in the feedback iterative loop. However, there are no native analogue delay parts in PSpice, and two methods were investigated to address this. The first method in Figure 7.1 used a Laplace part which contains a transfer function, which is the ratio of a system output to input, where \( \text{EXP}(-st) \) was entered into the numerator part of the transfer function, and one in the denominator.

However, the Laplace delay was not as robust as the transmission line delay. The Laplace part produced undesirable signal artefacts due to causal problems, but this was partially cured by adding an exponential term to the transfer function. [Tobin, 2007d]. The Laplace delay also produced more convergence problems. The second delay method used a transmission line part, \( T \), which must be terminated correctly by placing resistances equal to the transmission line characteristic impedance, \( Z_0 \), at the input and output, otherwise, signal reflections will be reflected back to the input [Tobin, Paul, 2007], [Tobin, 2007c] \(^1\). This correctly terminated \( T \) transmission line formed the subcircuit which is attached to a symbol created for this part. The subcircuit also had unity-gain buffer amplifiers to prevent external chaos circuitry from changing the line impedance levels.

\(^1\)The input impedance of a long length of open-circuit transmission line is called its characteristic impedance, \( Z_0 \)
7.2.2 Application of the Limit Part

The LIMIT ABM part discussed in Section 6.2.1 was placed in the feedback path of digital chaos maps and allowed the simulation to continue over the complete growth factor range. Without this part, convergence problems would stop the simulation. This very simple, but important discovery, prevented the system from attempting to go to infinity but did so without affecting the overall dynamics. The maximum and minimum excursions of the LIMIT part can be adjusted but it is important that the limits are not less than the normal signal excursion, otherwise, the dynamics of the model changes.

7.3 Simulation Meters

Simulation meters were created to evaluate analogue chaos systems in PSpice. For example, a peak meter is required for identifying the bifurcation phenomenon for a range of system parameters.

7.3.1 Peak Meter

This meter produces meaningful bifurcation plots in continuous chaotic systems. Bifurcation plots require the analogue output to be sampled at the peak values to identify bifurcation points. Regular sampling cannot be used since peaks do not occur at regular intervals and the peak meter samples the state variable at peak values. The peak meter subcircuit schematic in Figure 7.2 shows two delays which are necessary to find the peak value in adjacent points in the time series.

\[ \text{EXP1} = \{\text{if}(\text{V(B)} > \text{V(INPUT)}) \& (\text{V(B)} > \text{V(C)}), \text{V(B)}, 0)\} \]

**Figure 7.2**: Peakmeter subcircuit.

---

\(^2\text{A peak meter is not required for observing bifurcation in discrete maps because of the discrete nature of the map}\)
Chapter 7. PSpice Simulation of Chaos

The decision-making is in an “IFTHENELSE” statement of the transitive relation between delayed versions of the signal and is attached to an ABM2 part. The T part has a delay parameter box whereby the delay defined in a subparam part is entered as a @Delay parameter. The delay subcircuit is attached to a new symbol created which was then added to the author’s library called mylib.lib. Two unity GAIN parts buffer the input and output terminals to prevent any external loading by the external chaos circuit. Buffering ensures the transmission delay line is correctly terminated at all times, otherwise, the impedance of the circuit would load the terminating line resistors, R1 and R1 and generate extra signal reflections. The peak meter netlist is displayed in Table 7.1.

Table 7.1: The Peak meter subcircuit netlist.

```
.SUBCKT peak meter input output PARAMS: DELAY={1u}
T_T2 B 0 C 0 Z0=1k TD={Delay}
T_T1 N16919 0 B 0 Z0=1k TD={Delay}
R_R1 C 0 1k TC=0,0
E_GAIN2 OUTPUT 0 VALUE {1 * V(N16975)}
E_ABM1 N16975 0 VALUE { If((V(B)>v(INPUT) & (V(B)>v(C))),v(B),If((V(B)<v(INPUT) & (V(B)<v(C))),v(B),0))}
_GAIN1 N16919 0 VALUE {1 * V(INPUT)}
R_R2 0 N16919 1k TC=0,0
.ends
```

Figure 7.3 shows the z signal sampled at its peak value. While this looks like regular sampling, it is not. The z output signal is superimposed at the peaks of the continuous Lorenz z signal and the time signal is replaced by the variable resistance explained shortly.

Figure 7.3: Detecting peaks in the Lorenz Z signal (resistance on the x-axis).
Chapter 7. PSpice Simulation of Chaos

7.3.2 Resistance Meter

A resistance meter is required for bifurcating plotting and is explained in Section 7.4, but a brief explanation is now given. To obtain a bifurcation plot requires varying a system parameter which generally is a resistance and observing the effect of parameter variation on the output at the peaks. This variable resistance uses a three-terminal, ZX part as shown in Figure 7.4 (a). This is a voltage-controlled resistance (VCR) as part of the system for obtaining a bifurcation plot in chaos systems [Tu-inenga, 1988]. An input signal applied between pin 1 and ground, produces a parameter variation, and a reference resistance is connected between pin 3 and ground.

![Figure 7.4: (a) VCR ZX part (b) The ZX subcircuit.](image-url)

The ZX subcircuit model in Figure 7.4 (b) has a subcircuit netlist shown in Table 7.2.

<table>
<thead>
<tr>
<th>TABLE 7.2: The ZX model subcircuit netlist.</th>
</tr>
</thead>
<tbody>
<tr>
<td>subckt VARIRES 1 2 CTRL.</td>
</tr>
<tr>
<td>R1 1 2 1E10</td>
</tr>
<tr>
<td>G1 1 2 Value = ( V(1,2)/(V(CTRL)+1 \text{ u}) )</td>
</tr>
<tr>
<td>*Resistor R and current I develops a voltage V. Resistor R is represented</td>
</tr>
<tr>
<td>*by a current source I, where 1 and 2 are the resistor terminals:</td>
</tr>
<tr>
<td>I=( V(1,2)/R )</td>
</tr>
<tr>
<td>* Variable impedance: Zout = Zref*V</td>
</tr>
<tr>
<td>* control input: voltage</td>
</tr>
<tr>
<td>*</td>
</tr>
<tr>
<td>*/ | output: floating impedance</td>
</tr>
<tr>
<td>* + - 1 / \</td>
</tr>
<tr>
<td>.subckt zx 1 2 3 4 5</td>
</tr>
<tr>
<td>eout 4 6 poly 2 (1,2) (3,0) 0 0 0 0 1</td>
</tr>
<tr>
<td>fcopy 0 3 vsense 1</td>
</tr>
<tr>
<td>rin 1 2 1G</td>
</tr>
<tr>
<td>vsense 6 5 0</td>
</tr>
<tr>
<td>.ends</td>
</tr>
</tbody>
</table>
Figure 7.5 (a) shows a control voltage connected to pin 1, where the voltage at pin 4 is an input voltage, $V_{in1} = 1 \text{ V}$. The resistance characteristic is plotted in Figure 7.5 (b) with the vertical as $R_{in} = \frac{V_{in}}{-I(Rp2)}$, plotted against time.

![Figure 7.5: (a) Zx part (b) Resistance versus time.](image)

### 7.4 Applications for the New Meters

Three virtual instruments were created to investigate continuous analogue chaos system bifurcation in PSpice: a VCR meter, a bifurcation meter, and a peak detector meter. It is essential that a nonlinear system operates in the chaotic region to generate random binary number sequences with maximum entropy. This necessitates plotting bifurcation plots for the chaos systems under investigation. In standard English, bifurcation means splitting in two, but in chaotic systems, it has a broader meaning because the path divides more than twice. To investigate this phenomenon means sweeping a state variable parameter over a range of values and observing the effects of the parameter change.

Bifurcation is classified as *pitchfork*, *(flip, or period-doubling)*, but other bifurcation types, such as the *hopf*, occurs for imaginary eigenvalues (FP). A *transcritical* bifurcation occurs where the system flips between stable and unstable FPs. When the real part of the eigenvalue passes through zero it produces a *saddle node* or fold bifurcation. The type of bifurcation depends on how the fixed points (FP) are created or destroyed. A bifurcation occurs when an FP becomes unstable, and observing this in continuous chaotic systems requires sampling the output of a variable at a maximum point in the signal. However, sampling the output at the Nyquist frequency does not work since the peak values occur randomly.

---

3 Other methods for evaluating chaos are discussed in Chapter 9
7.4.1 Lorenz Oscillator Bifurcation Plotting

Two meters are required for plotting a continuous chaotic system bifurcation diagram. The 'bifurcation meter' replaces the Lorenz Rayleigh 37.5 kΩ resistance and which sweeps a resistance over time, and a peakmeter which samples the z output at its maximum value. The resistance meter is a ramp voltage generator, Vsweep, (VPWL), with the following time and voltage settings: $T_1 = 0, V_1 = 5k$, and $T_1 = 10 V_2 = 100 k$, as shown in Figure 7.6.

**Figure 7.6:** The Lorenz bifurcation schematic.

A current probe is positioned on the Rsense resistor, as shown, and the resistance range is determined by the values of $V_1$ and $V_2$ in the Vsweep generator. After simulation, a plot appears automatically and the horizontal time axis is changed to the variable $(v(x) - v(Rv_2))/I(Rsense)$ to display the bifurcation plot in Figure 7.7.

**Figure 7.7:** Sampled Lorenz (Z) vs. $(v(x) - v(Rv_2))/I(Rsense)$.

---

4Probe is a PSpice program for displaying simulation results.
A ‘Period-Three’ window can be observed for $R$ approximately 12.9 kΩ but disappears for $R$ equal to 13.3 kΩ [Li and Yorke, 1975]. This shows how sensitive the system is for very small resistance changes.

### 7.4.2 Chua Chaos Oscillator Bifurcation Plot

The Chua bifurcation schematic in Figure 7.8 shows the same virtual instruments used for plotting the Lorenz bifurcation diagram.

![Bifurcation meter diagram](image)

**Figure 7.8:** A Chua bifurcation schematic.

The Chua bifurcation plot in Figure 7.9 displays a period-three window at $R = 1.835$ kΩ. The time axis was replaced by the meter resistance, $(v(x)-v(Rv2))/I(Rsence)$.

![Bifurcation plot diagram](image)

**Figure 7.9:** Chua bifurcation plot period-three (1.82 kΩ to 1.87 kΩ).

In the Chua chaos system, a small change in the resistance produced a large change in the bifurcation plot.
7.5 The Lorenz Return Map

A return map is a useful technique for establishing if there is a correlation between one point and the next point in a random binary sequence. There should be no correlation between any points in a sequence if it is truly random. The Lorenz return map is a graph of successive maximum values of $Z$ plotted against a delayed $Z_d$ as shown in Figure 7.10 is a plot of the $Z$ signal.

The Return map shows a cloud of points near the apex of the plot rather than a continuous line. Thus, it shows how difficult it is to predict one point from a future point. The Identity line intersects the map at a fixed point.

7.5.1 Plotting the Return Map

A little-known technique in PSpice creates an ASCII text file from a previous simulation where the variables are copied and used as an input source for another simulation, as described in [Tobin, Paul, 2007]. The peaks in the Lorenz $Z$ signal was created using the peak meter developed and exported from Probe using copy and paste to a text file comprising time and voltage columns and saved in a text file similar to the digital tent map introduced in Chapter 6 Section 6.10.2)
Chapter 7. PSpice Simulation of Chaos

called Lorenz_return.txt. The file was then attached to a VPWL_FILE PSpice generator forming the input signal in Figure 7.11 and simulated.

\[
\text{if}((\text{V(in)} >= 39.5 \& \& \text{V(in)} <= 45.5) , \text{V(in)}, 0)
\]

Two one second delays enable decisions to be made about the Lorenz peak series. An ABM2 part has three input signals as shown and an ‘IFTHENELSE’ statement entered as \text{if}((\text{V(in)} >= 39.9 \& \& \text{V(in)} <= 45.5) , \text{V(in)}, 0). The two values, 39.9 and 45.5, are voltage levels from the Lorenz Z data on either side of the peak voltage.

7.6 Poincaré Section Plotting

Henri Poincaré visualised a complex chaotic trajectory in 3-D phase space by reducing the dimensions of a chaotic system using an intersecting hyperplane. The circuit in Figure 7.12 shows two open-collector comparator circuits connected via a pull-up resistance, \(R_4\), and connected to the positive DC supply.

\[
\text{if}(\text{V(z)} = \text{zsection-dz}) \& \& \text{V(z)} = (\text{zsection} + \text{dz}) , \text{V(z)}, 0)
\]

Two one second delays enable decisions to be made about the Lorenz peak series. An ABM2 part has three input signals as shown and an ‘IFTHENELSE’ statement entered as \text{if}((\text{V(in)} >= 39.9 \& \& \text{V(in)} <= 45.5) , \text{V(in)}, 0). The two values, 39.9 and 45.5, are voltage levels from the Lorenz Z data on either side of the peak voltage.

\[\text{if}(\text{V(z)} = \text{zsection-dz}) \& \& \text{V(z)} = (\text{zsection} + \text{dz}) , \text{V(z)}, 0)
\]

Figure 7.11: A PSpice schematic for plotting a Lorenz return map.

Figure 7.12: Novel design for producing a Poincaré section.
Chapter 7. PSpice Simulation of Chaos

Sampling phase space along a particular axis creates a digital Poincaré map. Driving a system with a periodic function, i.e., a non-autonomous (dependent on time) effectively samples the chaos system at the period of the forcing function. For autonomous system, sampling in any hyperplane produces a cross-section of the attractor. Sampling an attractor in any plane produces a 2-dimensional Poincaré section producing a series of dots which represent the attractor trajectory turning a continuous signal system to discrete. A Poincaré section can only be applied to continuous chaos systems and not to discrete chaotic systems such as the Logistic map examined in Chapter 6. The FPs at the centre of each lobe of the Poincaré section are displayed as two discontinuities and may be compared to the Poincaré section plotted in Figure 3.8, Chapter 3.

The Lorenz chaos generator in the hierarchical block is connected to the comparator which detects where the attractor is intersected by the hyperplane, $Z_{sec}$. The window circuit produces a clock pulse every time the $Z$ plane cuts the axis at 1.3 and operates two sampling analogue electronic switches to produce sampled $x$ and $y$ signals. These signals drive the oscilloscope inputs, or in a simulation setting, the horizontal axis is changed from time to $v(x)$. The ABM1 part in Figure 7.12 creates samples of $x$ and $y$ outputs from the Lorenz generator using the ‘IFTHENELSE’ statement in the ABM1 EXP1 part. This uses a window defined $z_{sec} \pm dz$ and ensures the section at 27.3 for the non-scaled system is inside this window. The parameter, $dz$, fixed the width of the sampled variable at 0.01. Figure 7.13 displays sampled signals and a Poincaré plot from the electronic circuit.

![Figure 7.13: (a) Sampled Xs and Ys (b) Strange Attractor (c) A Poincaré section.](image-url)
7.6.1 Stability Loss Delay

The phenomenon of stability loss delay (SLD) discussed in Chapter 6, Section 6.3.1, is caused by sweeping a system variable too fast when investigating bifurcation. In this section, the ramp generator used for Logistic map bifurcation analysis was replaced with a staircase signal generated in Matlab® using the native m file called stairs.m. The signals were exported from Matlab® in ASCII form as a text file and attached to a PSpice VPWL generator part. In a staircase signal, the growth parameter change occurs very quickly during the riser part of the generator waveform but remains at a constant value during the step part. This eliminated SLD. The staircase generator was replaced by the novel design shown in Figure 7.14 (a) which generates a staircase signal using a clock and a 12-stage binary ripple counter connected to a 12-bit digital-to-analogue (DAC) converter.

The staircase characteristic in the right pane of Figure 7.14 (b) looks like a ramp but when magnified the riser and step can be clearly seen in the inset. The growth parameter, $R$, changes state rapidly during the riser but the parameter remains constant during the step and the map is evaluated over this period. The maximum amplitude of the staircase is four volts and corresponds to a growth rate factor of four in the Logistic map.

7.7 Chapter Conclusion

This chapter discusses the important role played by PSpice and how many chaos systems could be evaluated for maximum entropy thus ensuring the prototype met the correct standard for certification. The encoder design was simulated in PSpice and the randomness of the OTP sequences tested using the important NIST tests.
Chapter 7. PSpice Simulation of Chaos

However, it was not practical to simulate the encoder for the NIST parameterised tests as this test requires at least 20 Mbit sequence length and was not practical (These tests were carried out on the prototype). PSpice was not designed for simulating nonlinear circuits and the author had to design new parts and tools and adopt new simulation paradigms, to handle the many convergence problems that occurred during the research. Lyapunov analysis explained in Chapter 3, processed data exported from many chaos circuits simulated and resulted in two chaos systems selected for the prototype.

An insight into the design of the encoder was achieved by showing how the chaos systems behaved during simulation of the chaos circuits in PSpice. Here, parameters could be changed to see almost instantly how the system responded and this saved much prototyping time by eliminating wasted effort and meant many models and systems to be investigated in a shorter time. Signals from the prototype were measured with a Tiepie software oscilloscope (Chapter 8) and were compared to simulation signals and showed the accuracy of the simulation environment. The complete encoder prototype is discussed in Chapter 8 and comprehensive NIST test results on the complete encoder in Chapter 9 verified the excellent randomness of the OTPs produced.
8 The Prototype OTP Encoder

“Why,” said the Dodo, “the best way to explain it is to do it.” Alice’s Adventures in Wonderland- Lewis Carroll

8.1 Chapter Overview

This chapter examines the prototype encoder which has two delay chaos sources seeded with noise from a 433 MHz data receiver. The encoder, mounted on a two-sided printed circuit board (PCB), facilitated exact and extensive testing using a software oscilloscope. Prototype and simulation results were compared to show the accuracy of the PSpice simulator. Finally, the FIPS-140 international standard for obtaining security certification for encryption devices is presented in Section 8.6.

8.2 Prototype Chaos Source Selection

It was necessary to construct an encoder prototype OTP encoder to validate the design by testing large OTP sequences as outlined in the NIST tests. The NIST suite of tests has certain tests which require very large binary files and it wasn't practical to simulate and produce large binary files (at least 20 Mbits), hence a prototype was required for these tests. Chaos systems were investigated by evaluating the entropy of the data exported from each source. Intuition played a part in the random evaluation process but the Lyapunov exponent (LE) was introduced as part of the Lyapunov Dimension (LD) (also called the Kaplan-Yorke dimension (KYD)). This was the main metric for evaluating the entropy of each chaos source and subsequently selecting two chaos systems for the prototype. The following section explains how the LD results in Table 8.1 were obtained. However, calculating the LD by hand for many chaos systems proved tedious and the solution was to use software downloaded from the Internet.

1The NIST tests presented in Chapter 9 form part of this standard
8.2.1 Lyapunov Exponent

The Russian mathematician, Aleksandr M. Lyapunov, proposed a single number called the LE to represent the change in a trajectory caused by perturbing a nonlinear system [Kuznetsov, Alexeeva, and Leonov, 2016]. The LE tracks how a trajectory deviates with time and is obtained by dividing the perturbation size at an instant in time by its size a moment before. This is repeated at intervals and the results are averaged [Dendney, 1991]. The trajectory is compared to a fiduciary trajectory a moment later and allows the LE to be computed to establish if a time series is chaotic [Wolf et al., 1985]. Figure 8.1 (a) shows a growing LE calculated from two identical Lorenz systems simulated in PSpice, where the IC to both systems differed only by 1.000000001 Volts. The LE will grow and stabilise to a mean positive value.

A perturbation to a system will die out if the LE is less than zero, showing how the system is stable. However, if the LE of a system is greater than one, it is unstable, and the perturbation will cause the system output to grow. The LE measures how fast two chaotic trajectories diverge from each other and must be positive if a chaos system is to generate chaotic trajectories within a few iterations and produce random binary sequences.
8.2.2 Lyapunov Time

Figure 8.1 (b) shows $v(x)$ and $v(xx)$ signals deviating from each other at approximately 359 ms (the Lyapunov Time, (LT)). This is the characteristic timescale on which a dynamical system is chaotic and is the inverse of a system’s largest LE. In the Lorenz system, the difference between the two IC in the variables, $v(x)$ and $v(xx)$, is amplified exponentially as:

$$v(x) - v(xx) \propto e^{\lambda t}$$

(8.1)

$$\lambda = \frac{1}{t} \log_e \{v(x) - v(xx)\}$$

(8.2)

The time taken for the large differences to occur is $1/\lambda$. Starting chaos oscillators from an unknown seed value means the prototype encoder produces true OTP sequences as discussed in Chapter 4.

8.2.3 Computing the Lyapunov Exponent

The LE, $\lambda(t_0)$, for an IC at $t_0$, measures the exponential divergence of trajectories infinitely close to each other [Schuster, Martin, and Martienssen, 1986]. For a one-dimensional system time series function, $f(t)$:

$$|f_n(t_0 + \epsilon) - f_n(t_0)| = \epsilon e^{n\lambda(t_0)}$$

(8.3)

where $\epsilon$ is a small perturbation from an IC $t_0$, and $n$ is the number of iterations. Generally, $\lambda$ depends on the IC and only an average value is estimated. In a measure-preserving system $\lambda$ is constant for all trajectories:

$$\lambda(t_0) = \lim_{n \to \infty} \lim_{\epsilon \to 0} \frac{1}{n} \log \left| \frac{f_n(t_0 + \epsilon) - f_n(t_0)}{\epsilon} \right|$$

(8.4)

Or,

$$\lambda(t_0) = \lim_{n \to \infty} \frac{1}{n} \sum_{k=1}^{n} \log |f'(t_k)| = \lim_{n \to \infty} \frac{1}{n} \log \prod_{k=1}^{n} |f'(t_k)|$$

(8.5)

For each $k$, $f'(t_k)$, shows how much the function changes with respect to its argument at the point $t_k$. The derivative expresses the magnitude of change in the transition from $t_k$ to $t_{k+1}$. The limit of the average of the derivative logarithms, over $n$ iterations, shows how fast the orbit changes with time. The LE is calculated for a
discrete stochastic time series $R(t_1), R(t_2), ..., R(t_N)$ as:

$$\lambda = \frac{1}{N} \sum_{n=1}^{N} \log_2 \left| \frac{R_{t_{n+1}}}{R_{t_n}} \right|$$  \hspace{1cm} (8.6)

### 8.2.4 The Prototype Encoder

Table 8.1 shows a range of chaos systems tested for maximum LD using software downloaded from 2. Column two and three show which systems were simulated and which were built. Two novel delay chaos oscillators were chosen for the encoder prototype which had a maximum LD metric, as highlighted in red in the last column.

<table>
<thead>
<tr>
<th>Analogue Chaos</th>
<th>Simulated</th>
<th>Prototype</th>
<th>Lyapunov Dim.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chua</td>
<td>Y</td>
<td>Y</td>
<td>1.997</td>
</tr>
<tr>
<td>Delay Chua</td>
<td>Y</td>
<td>Y</td>
<td><strong>2.659</strong></td>
</tr>
<tr>
<td>Duffing</td>
<td>Y</td>
<td>N</td>
<td>0.187</td>
</tr>
<tr>
<td>Lorenz (FOC)</td>
<td>Y</td>
<td>Y</td>
<td>2.6</td>
</tr>
<tr>
<td>Lorenz</td>
<td>Y</td>
<td>Y</td>
<td>2.07</td>
</tr>
<tr>
<td>Delay Lorenz</td>
<td>Y</td>
<td>Y</td>
<td><strong>2.327</strong></td>
</tr>
<tr>
<td>Nosé-Hoover</td>
<td>Y</td>
<td>Y</td>
<td>0.147</td>
</tr>
<tr>
<td>Rössler</td>
<td>Y</td>
<td>N</td>
<td>2.01</td>
</tr>
<tr>
<td>Rikitake</td>
<td>Y</td>
<td>Y</td>
<td>0.874</td>
</tr>
<tr>
<td>Rucklidge</td>
<td>Y</td>
<td>Y</td>
<td>0.997</td>
</tr>
<tr>
<td>RLD</td>
<td>Y</td>
<td>Y</td>
<td>-1.7</td>
</tr>
<tr>
<td>Ueda</td>
<td>Y</td>
<td>N</td>
<td>0.902</td>
</tr>
<tr>
<td>Uraglu</td>
<td>Y</td>
<td>N</td>
<td>-2.183</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chaos Maps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baker</td>
</tr>
<tr>
<td>Bernouilly</td>
</tr>
<tr>
<td>Gaussian</td>
</tr>
<tr>
<td>Hénon</td>
</tr>
<tr>
<td>Logistic</td>
</tr>
<tr>
<td>Tinkerbell</td>
</tr>
</tbody>
</table>

In chaos systems, a small change in the IC will produce a different trajectory within a short time. Similarly, in cryptography changing a small portion of the key will produce a different ciphertext.

2http://www.matjazperc.com/time/
8.2.5 Prototype Overview

Figure 8.2 is an overview of the prototype chaos encoder. The delay Lorenz and Chua analogue chaos oscillators examined in Chapter 3 were connected to two threshold circuits which converted the four chaos analogue voltage levels to binary signals.

Effectively, the 1-bit ADC threshold circuit produces OTP sequences to encode sensitive data locally and producing a ciphertext which is stored in the Cloud [Barker and Kelsey, 2012]. It was necessary to convert the four variable-width analogue pulses to four constant-width bitstreams using monostable devices to ensure the digital logic circuits operated correctly. The four bitstreams were converted to dibit binary pairs and processed by the Arduino Nano microcontroller into a 1-bit stream. The microcontroller introduces a VN deskewing algorithm to increase the entropy by removing certain bias and writes the final OTP to a text file which was tested for randomness, as outlined in Chapter 9.

8.3 The Complete Prototype Schematic

The chaos circuits from Chapter 3 Section 3.9, and Subsection 3.10.2, are shown in the prototype encoder circuit diagram in Figure 8.3. The threshold circuits for converting the Lorenz and Chua analogue signals to digital are identical except for the potential divider values because these values depend on the equilibrium points (FP) for each attractor. The XOR gates at the output were included at the design stage but, subsequently, were not used because the Arduino Nano microcontroller performed the logical operations. The microcontroller USB port communicated with the desktop interface to download the compiled Nano data. Following on from this, the OTP sequences are then sent from the microcontroller via the USB port in a text file to the desktop for NIST testing.
Figure 8.3: The complete prototype OTP encoder circuit diagram.
Chapter 8. The Prototype OTP Encoder

8.3.1 Printed Circuit Board Prototype

The two-sided PCB in Figure 8.4 was created in a cloud-based facility, EasyEDA, in China\(^3\).

An intricately-laid out schematic was constructed which was carefully checked for errors before Gerber files were uploaded to the PCB manufacturer at the same location. A ground plane was formed from the top-side PCB copper pour and the bottom copper pour connected to the 5 V DC supply. EasyEDA processed the uploaded Gerber files and eventually sent five PCB to the author. The PCB in Figure 8.5 was populated by the author with parts which includes many blue multi-turn potentiometers for post-build investigation.

\(^3\)https://easyeda.com/

However, these potentiometers will be replaced by resistors making the final encoder smaller with dimensions similar to a flash memory device using surface-mounted component design.
8.4 Testing the Lorenz Chaos Oscillator

Measurements using a Tipe Pie HS3 USB software oscilloscope allowed prototype signals to be saved as portable network graphics (PNG) images. Figure 8.6 shows the $x$ and biased $x$ signals from the Lorenz prototype and are almost identical to signals from simulation in Chapter 3.

![Figure 8.6: (a) Scaled $x$ and biased $x$ signals with Fixed Points on the $x$ signal (b) Reset signal on the $x$.](image)

Set and reset signals are generated each time the value of the $x$ signal equals the two threshold levels. Two horizontal lines are superimposed on the Lorenz $x$ signal to identify the FP values threshold levels as calculated in Chapter 3. The 1-bit ADC circuit generates binary signals forming the OTP. Figure 8.7 shows the scaled Lorenz strange attractor plotted with a Tipe Pie software oscilloscope showing two loci at FPs $= \pm 0.8485$ V.

![Figure 8.7: The scaled Lorenz strange attractor.](image)

---

4https://www.tiepie.com/en
8.4.1 Testing the Delay Chua Chaos Oscillator

The delay Chua prototype system strange attractor in Figure 8.8 is different because of the delay in the feedback path when compared to the basic Lorenz and Chua attractors plotted previously.

![Figure 8.8: The prototype Chua strange attractor.](image)

8.5 Late Prototype Developments

Figure 8.9 shows extra circuits not included in Figure 8.3 because they were included in the PCB design for alpha prototype testing.

![Figure 8.9: (a) Nand Clock (CD4011BE) (b) 5 V regulator (LM7805E) (c) Noise sampling (HCF4016B) (d) The data receiver antennae switch.](image)

The NAND digital clock design in Figure 8.9 (a) was originally included to operate the electronic LM4016 switch in Figure 8.9 (c) to sample the initialising noise from the 433 MHz data receiver. However, it was decided not to use this NAND.

---

5 This clock was not used in the final prototype
clock and instead use one of the chaos pulse streams from the Lorenz generator as it provides an extra degree of security over regular sampling. The 5-Volt regulator (LM7805E) in Figure 8.9 (b) supplies DC to the logic circuits and is derived from the positive 12 V supply.

8.5.1 VN and XOR Deskewing Correctors

Two methods for deskewing and removing residual bias in the OTP are the VN algorithm discussed in Section (1.4.4) and the XOR modulo-two arithmetic using an exclusive-OR gate in Figure 8.10. The XOR operates on blocks of 4 bits to generate one output bit to eliminate bias in the random bit streams. However, removing bias in the output bit stream using the VN algorithm is valid only when the bit streams are independent and was the reason for two different chaos streams in the prototype. Most of the logic gate circuits shown here were not used in the final prototype design because the VN algorithm and XOR logic were implemented in the Arduino Nano board mounted on the main board. Whether the VN algorithm is used depends on the level of secrecy required in the encoding process.

\[\text{Figure 8.10: (a) XOR and 7402 NOR gates (b) The Arduino Nano with a USB port.}\]
Chapter 8. The Prototype OTP Encoder

Figure 8.11 shows a switch for isolating each chaos circuit for testing purposes.

![Diagram of switch and connections]

**Figure 8.11:** (a) Tuning capacitor dual-in-line (DIL) holder (b) Chaos isolation switch (c) Connector.

A DIL integrated circuit holder was included whereby extra capacitors could be added to change the operating frequency of each chaos source. Adding capacitance in parallel with the existing integrating capacitors reduces the return frequency. A physical switch isolated each chaos source so each source could be tested separately.

### 8.6 FIPS standard of certification

All new cryptographic devices that protect sensitive, but unclassified information in the U.S., must complete a Federal Information Processing Standards Publications (FIPS) 140-2 validation ensuring they have a high degree of security, assurance, and dependability. All cryptographic devices used by the U.S. Federal government are required to have FIPS 140-2 validation. This standard has four qualitative levels of security increasing in security intensity from Level 1 up to Level 4 and covers a wide range of applications and environments for cryptographic modules and issued by NIST [Carnahan and Smid, 1994]. This is the U.S. and Canadian co-sponsored security standard for hardware, software, and firmware solutions.

Figure 8.12 shows the complicated procedure necessary for obtaining certification for any encryption security devices used in the USA.

---

6The European Union also recognize the FIPS 140-2 standard
Chapter 8. The Prototype OTP Encoder

8.7 Chapter Conclusion

The selection of chaos sources initialised with sampled electronic noise for the prototype encoder was based on the results from LD analysis. Measurements on the encoder using a TiePie software USB oscilloscope showed results almost identical to those obtained from simulation. The OTP binary sequences were processed in the Arduino Nano microcontroller which achieved maximum OTP entropy from pairs of dibits using a VN de-skewing algorithm and modulo-two arithmetic. The Nano microcontroller USB port interfaced the encoder and desktop using software which is included in Appendix C. The software created OTP binary sequences by combining four digital binary signals from the Lorenz and Chua delay oscillators forming the OTP encoder which was then written to a file for NIST testing. The alpha PCB prototype encoder was made much larger for testing purposes containing many components which will not be in the final version. The FIPS-140 certification process shows the rigorous testing procedure required for any cryptographic device used in public encryption.
9 Testing the OTP Encoder Prototype

“The First World War was the chemists’ war because mustard gas and chlorine were employed for the first time. The Second World War was the physicists’ war because the atom bomb was detonated. The Third World War would be the mathematicians’ war because mathematicians will have control over the next great weapon of war —information.” Simon Singh

9.1 Chapter Overview

In this chapter, NIST test results on the OTP binary sequences from the encoder simulation data and also from the prototype are given. Short and long OTP encoder sequences from the prototype met all the NIST standards for randomness (other tests were carried out which overlap and complement these NIST tests). Entropy is an important metric for measuring randomness and techniques for quantifying it were introduced in Chapter 1. A new test was created to measure the entropy for a range of analogue delays added to the chaos oscillators to create a hyperchaos system.

9.2 Testing the OTP Encoder Prototype

It is impossible to say if a sequence of binary numbers is truly random but only that they are statistically random, with a probability distribution containing no recognisable patterns. The following additional tests were also carried out on the prototype OTP sequences:

- Autocorrelation,
- Power Spectral Density (PSD),
- Shannon entropy,
- Kolmogorov-Sinai entropy and Algorithmic Complexity,
- Histogram distribution,
- Probability Distribution Function (PDF) test, and
- Averaging entropy test for the analogue delay.
9.2.1 Autocorrelation Test

The autocorrelation function (ACF) checks for a correlation between a digital binary sequence, \( x_n \), and a delayed version of it, \( x_{nd} \).

\[
ACF_d = \frac{1}{N - d} \sum_{n=0}^{N-d} x(n)x(nd) \quad (9.1)
\]

The ACF is a single Kronecker delta function for truly random data. Figure 9.1a shows the ACF for true noise and for uni-polar prototype OTPs which inherently contain a DC bias and shows the delta function on top of a triangle display. However, Figure 9.1b shows the ACF for a bipolar random sequence which has no DC.

![Figure 9.1](image1.png)

Additional correlation peaks on either side of the delta function indicate the binary stream is not truly random.

9.2.2 Power Spectral Density

An encoder key should have a uniform PSD to prevent spectral attacks by an adversary. The PSD is the absolute value of the square of the magnitude of the Fast Fourier Transform (FFT) shown in equation (9.2).

\[
S_{xf} = \lim_{T \to \infty} \left\{ \frac{1}{2T} \left| \int_{-T}^{T} x(t)e^{-j2\pi ft} dt \right|^2 \right\} \quad (9.2)
\]
Chapter 9. Testing the OTP Encoder Prototype

9.2.3 Probability Distribution Function and Histograms

The PDF determines the number of occurrences of a random variable within a certain range and is determined by integrating the PDF over this range. A PDF is a function from strings $L = \{a_j\}$, to nonnegative real numbers, i.e., $Pr : L \to [0, 1]$, such that $\sum_{a \in L} Pr(a) = 1$. A string $a$ is truly random if, for any substring $\beta_n, \gamma_n \in a$, $0 > n > \text{length}(a)$ $Pr(\beta_n) = Pr(\gamma_n)$. If a string of bit is truly random string it is impossible to predict a particular bit, i.e., for any symbol $s_i \in a$, the conditional probability $Pr(s_i|s_{i-1}, s_{i-2}, \ldots) = Pr(s_i)$. Knowledge of a previous state does not affect the probability of a successful prediction of the next state. OTP prototype sequences imported into Matlab and XORed with the Lena bitmap image produced a uniform histogram and PSD plots in Figure 9.2.

![Image](image.png)

**Figure 9.2:** (a) The Lena image (b) Histogram of unencoded image (c) PSD of unencoded image.

The decoded Lena image shows no visible picture degradation after recovery.
Chapter 9. Testing the OTP Encoder Prototype

9.2.4 Average Entropy Test

A novel test was developed for quickly assessing the effect on the chaos entropy when a Padé delay was added in the Lorenz and Chua chaos oscillators. The average value of a chaos analogue signal, \( x \), should oscillate around the horizontal axis. If it displays more positive (or negative) excursions then the signal contains a bias and is less random. Figure 9.3 shows the average value of the Lorenz \( X \) signal for three delays and is an indicator of the signal entropy.

![Figure 9.3: Averaging the \( x \) signal for three delays.](image)

The prototype delay was adjusted by varying the potentiometer \( R_d \) and observing which resistance value brought the average signal closest to the zero axis (in this example, a delay between 1.2 us and 1.4 us produced the best chaos entropy). Averaging a non-random sequence of alternating bipolar ones and zeroes would, of course, show an average of zero, but the method is very useful for assessing the presence of bias. The \( x \) signal was averaged by selecting an averaging function available from the PSpice Probe menu after simulating for a range of resistance values. The start-up transient in the average of \( x \) shows the signal is biased during this period and binary sequences should be rejected. This testing method is useful because NIST tests would not be practical when investigating the effects of varying a parameter such as a delay for best OTP entropy.
9.2.5 Kolmogorov Complexity

Another randomness metric in cryptography is the Kolmogorov complexity (KC). This was created simultaneously by Kolmogorov and Solmonoff and specifies the minimum length to which a binary sequence can be compressed (a true random sequence is incompressible) [Landauer, 1961]. Essentially, KC is Shannon’s entropy for a sequence of bits [Tobin and Blackledge, 2014], [Grassberger and Procaccia, 1983]. Complexity and entropy are “cause and effect”; the more complex a system is the more unpredictable its behaviour resulting in greater entropy. Positive KS-entropy value is proof of chaotic behaviour and randomness and is related to algorithmic complexity (AC), where the system is ergodic. Complexity is the size of an “internal program” that generates a binary sequence, whereas entropy is computed from the probability distribution of that sequence. For certain conditions in phase space, Brudno’s theorem states that KS is the AC over all the trajectories [Frigg, 2003].

9.3 Encoding Dicom Images

Figure 9.4 shows a JavaScript application interface \(^1\) for encoding medical images and documents output with the OTP. A DICOM MRI head image pixel data array was XORed with the OTP data from the prototype.

The original DICOM in the left pane is an example where only the patient personal metadata was encrypted. However, the middle pane shows the results of encoding the complete image. The right pane shows another encrypted image but this time the OTP was first processed through a VN deskewing algorithm to remove bias, as

\(^1\)https://github.com/leetobin/ChaosEncrypt
discussed in subsection 9.3.1. There is little discernible difference between the two encoded images because the OTP was not biased.

9.3.1 Bias in the OTP

Bias should always be removed from OTPS because it makes the encoded image susceptible to cryptanalysis and shows up as lines in the encoded image in Figure 9.5. The VN deskewing algorithm discussed in Chapter (3) removes bias in OTP sequences to increase the stream entropy [Von Neumann, 1951]. The algorithm processes pairs of bits called dibits, where each bit is independent of each other. This bit independence is often overlooked in research papers but here it is achieved by combining bits from two independent chaos streams.

A p-value test results of the OTP is shown in the JavaScript window at the top right pane. This is a useful quick assessment to see if the OTP entropy is acceptable (see Chapter 9) before and after the VN correction algorithm was applied.

9.3.2 NIST Suite of Statistical Tests

NIST developed a comprehensive suite of statistical tests (NIST SP 800-22 publication) to evaluate and compare, random number generators (RNG), where randomness is measured as a probabilistic p-value [NIST, 2001]. These tests introduce hypothesis testing shown in Table 9.1 to establish if OTP sequences fall within a statistical probability band of randomness. Hypothesis testing verifies the null hypothesis,

<table>
<thead>
<tr>
<th>Outcome</th>
<th>Accept $H_0$</th>
<th>Reject $H_0$, $H_a$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data is Random</td>
<td>No error</td>
<td>Type I error</td>
</tr>
<tr>
<td>Data is not Random</td>
<td>Type II error</td>
<td>No error</td>
</tr>
</tbody>
</table>

$H_0$, is true [Alcover, Guillamón, and Ruiz, 2013], [Bahi et al., 2010], [Hamano, Sato,
Chapter 9. Testing the OTP Encoder Prototype

and Yamamoto, 2009]. If the OTP statistical p-value test is within the significance level, 0.01 to 1, it is random, but if outside which is the alternative hypothesis, \( H_a \), it is not. The OTP is random if the null hypothesis, \( H_0 \), is accepted, but a conclusion can be reached for truly random sequences where the \( H_0 \) is rejected, but this has a small probability of occurring and is Type I error probability. Alternatively, if a sequence is non-random then Type II error occurs and \( H_0 \) is accepted. The probability of Type I error is called the level of significance and is set prior to a test. NIST recommends a probability should be in the range 0.001 to 0.01, where the objective is to minimise the probability of Type II errors. A proper sample size is chosen and the probability of Type I error is set to calculate the p-value. Setting Type I error probability to 0.001, means 1 in 1000 sequences are rejected. The sequence is considered random for p-value > 0.001, with an accuracy of 99.9, and is not random for a p-value < 0.001. Table 9.2 explains what each test is [Soto, 1999].

**Table 9.2:** NIST tests.

<table>
<thead>
<tr>
<th>NIST Test</th>
<th>Test Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency</td>
<td>Equality of ones and zeroes (global)</td>
</tr>
<tr>
<td>Block Frequency</td>
<td>Equality of ones and zeroes in a block (local)</td>
</tr>
<tr>
<td>Cumulative Sums</td>
<td>Too many zeros or ones at the start of sequences (local)</td>
</tr>
<tr>
<td>Longest Runs Of Ones</td>
<td>Deviation of the distribution of long runs of ones (global)</td>
</tr>
<tr>
<td>Runs Large (small)</td>
<td>Total number of runs show bit stream oscillation is too fast (too slow and local)</td>
</tr>
<tr>
<td>Rank</td>
<td>Deviation of the rank distribution from a corresponding random sequence, due to periodicity</td>
</tr>
<tr>
<td>DFT (spectral)</td>
<td>Periodic features in the bitstream.</td>
</tr>
<tr>
<td>Non-overlap Temp Match</td>
<td>Too many occurrences of non-periodic templates</td>
</tr>
<tr>
<td>Overlap Temp Match</td>
<td>Too many occurrences of m-bit runs of ones</td>
</tr>
<tr>
<td>Universal Statistical</td>
<td>Compressibility and Regularity</td>
</tr>
<tr>
<td>Random Excursions</td>
<td>Deviation from the distribution of the number of visits of a random walk to a certain state</td>
</tr>
<tr>
<td>RandExcursion Variant</td>
<td>Deviation from the distribution of the total number of visits (across many random walks) to a certain state</td>
</tr>
<tr>
<td>Approximate entropy</td>
<td>Small values imply strong regularity</td>
</tr>
<tr>
<td>Serial</td>
<td>Non-uniform distribution of m-length words similar to approximate Entropy</td>
</tr>
<tr>
<td>Linear Complexity</td>
<td>Deviation from the distribution of the linear complexity for finite length (sub)strings</td>
</tr>
</tbody>
</table>
Chapter 9. Testing the OTP Encoder Prototype

These tests ascertain if patterns are present in binary sequences and hence not random. The NIST parametric tests are applied to bit sequences of several million, while the non-parameter tests for short sequences of 1000 bits [Alcover, Guillamón, and Ruiz, 2013], [Bahi et al., 2010], [Hamano, Sato, and Yamamoto, 2009]. For best entropy, the p-value should be uniformly distributed in the interval (0:1). The NIST SP 800-22 tests require different parameters to be entered which depend on the test and the length of the sequences shown in Table 9.3 [Sýs and Říha, 2014].

Table 9.3: NIST requirements for OTP binary sequences.

<table>
<thead>
<tr>
<th>Test</th>
<th>Distribution</th>
<th>$n$</th>
<th>$n$ or $M$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency (monobits)</td>
<td>Half normal</td>
<td>$n &gt; 100$</td>
<td></td>
</tr>
<tr>
<td>Frequency Block</td>
<td>$\chi^2$</td>
<td>$n \geq 100$</td>
<td>20 ≤ $M$ ≤ $n/100$</td>
</tr>
<tr>
<td>Runs</td>
<td>$\chi^2$</td>
<td>$n \geq 128$</td>
<td></td>
</tr>
<tr>
<td>Longest run of 1’s</td>
<td>$\chi^2$</td>
<td>$n \geq 38912$</td>
<td></td>
</tr>
<tr>
<td>Rank</td>
<td>$\chi^2$</td>
<td>$n \geq 387 840$</td>
<td></td>
</tr>
<tr>
<td>DFT (spectra)</td>
<td>Normal</td>
<td>$n \geq 1000$</td>
<td></td>
</tr>
<tr>
<td>Non-overlap</td>
<td>$\chi^2$</td>
<td>$2 \leq m \leq 21^2$</td>
<td></td>
</tr>
<tr>
<td>Overlap T.M</td>
<td>$\chi^2$</td>
<td>$n \geq 10^6$</td>
<td></td>
</tr>
<tr>
<td>Maurer’s Universal</td>
<td>Half Normal</td>
<td>$n \geq 387 840$</td>
<td></td>
</tr>
<tr>
<td>Linear complexity</td>
<td>$\chi^2$</td>
<td>$n \geq 10^6$</td>
<td>500 ≤ $M$ ≤ 5000</td>
</tr>
<tr>
<td>Serial</td>
<td>$\chi^2$</td>
<td>$3 \leq m \leq \log_2 n-3$</td>
<td></td>
</tr>
<tr>
<td>Approx Entropy</td>
<td>$\chi^2$</td>
<td>$m \leq \log_2 n-6$</td>
<td></td>
</tr>
<tr>
<td>Cumulative sums</td>
<td>Normal</td>
<td>$n \geq 100$</td>
<td></td>
</tr>
<tr>
<td>Random Excursions</td>
<td>$\chi^2$</td>
<td>$n \geq 10^6$</td>
<td></td>
</tr>
<tr>
<td>Rand Excursions Var</td>
<td>Half Normal</td>
<td>$n \geq 10^6$</td>
<td></td>
</tr>
</tbody>
</table>

Table 9.4 show NIST results for OTPs from a simulation of the complete encoder.

Table 9.4: NIST results for short OTP sequences.

<table>
<thead>
<tr>
<th>Test</th>
<th>Noise p-value</th>
<th>Chaos p-value</th>
<th>Pass/Fail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency</td>
<td>$P = 0.412$</td>
<td>$P = 0.612$</td>
<td>Pass</td>
</tr>
<tr>
<td>Bl frequency</td>
<td>$P = 0.116$</td>
<td>$P = 0.100$</td>
<td>Pass</td>
</tr>
<tr>
<td>Runs</td>
<td>$P = 0.784$</td>
<td>$P = 0.055$</td>
<td>Pass</td>
</tr>
<tr>
<td>Bl Long Run Ones</td>
<td>$P = 0.538$</td>
<td>$P = 0.585$</td>
<td>Pass</td>
</tr>
<tr>
<td>Bin Matrix Rank</td>
<td>$P = 0.713$</td>
<td>$P = 0.437$</td>
<td>Pass</td>
</tr>
<tr>
<td>DFT spectra</td>
<td>$P = 0.520$</td>
<td>$P = 0.684$</td>
<td>Pass</td>
</tr>
<tr>
<td>Olap Tplate Match</td>
<td>$P = 0.772$</td>
<td>$P = 0.971$</td>
<td>Pass</td>
</tr>
<tr>
<td>Lin Complexity</td>
<td>$P = 0.95$</td>
<td>$P = 0.469$</td>
<td>Pass</td>
</tr>
<tr>
<td>Serial</td>
<td>$P1 = 0.197$</td>
<td>$P1 = 0.083$</td>
<td>Pass</td>
</tr>
<tr>
<td></td>
<td>$P2 = 0.544$</td>
<td>$P2 = 0.487$</td>
<td>Pass</td>
</tr>
<tr>
<td>Appr Entropy</td>
<td>$P = 0.114$</td>
<td>$P = 0.060$</td>
<td>Pass</td>
</tr>
<tr>
<td>Cum Sums</td>
<td>$P = 0.444$</td>
<td>$P = 0.675$</td>
<td>Pass</td>
</tr>
</tbody>
</table>
Chapter 9. Testing the OTP Encoder Prototype

The second column shows p-test results for binary sequences downloaded from [Haahr, 2010]. Table 9.5 shows NIST test results on long binary PSpice sequences of several million bits [Corporation, 2003], [Cristina and Eugen, 2012].

<table>
<thead>
<tr>
<th>Rand Excursion Test</th>
<th>$\chi^2$ test</th>
<th>p-value</th>
<th>Pass/Fail</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x = -4)</td>
<td>$\chi^2 = 3.7052$</td>
<td>P = 0.5925</td>
<td>Pass</td>
</tr>
<tr>
<td>(x = -3)</td>
<td>$\chi^2 = 5.0654$</td>
<td>P = 0.4079</td>
<td>Pass</td>
</tr>
<tr>
<td>(x = -2)</td>
<td>$\chi^2 = 2.1114$</td>
<td>P = 0.8335</td>
<td>Pass</td>
</tr>
<tr>
<td>(x = -1)</td>
<td>$\chi^2 = 0.7659$</td>
<td>P = 0.9791</td>
<td>Pass</td>
</tr>
<tr>
<td>(x = 1)</td>
<td>$\chi^2 = 1.5392$</td>
<td>P = 0.9084</td>
<td>Pass</td>
</tr>
<tr>
<td>(x = 2)</td>
<td>$\chi^2 = 0.5213$</td>
<td>P = 0.9913</td>
<td>Pass</td>
</tr>
<tr>
<td>(x = 3)</td>
<td>$\chi^2 = 2.2011$</td>
<td>P = 0.8206</td>
<td>Pass</td>
</tr>
<tr>
<td>(x = 4)</td>
<td>$\chi^2 = 11.649$</td>
<td>P = 0.0399</td>
<td>Pass</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rand Excursion Variant Test</th>
<th>Total visits</th>
<th>p-value</th>
<th>Pass/Fail</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x = -9)</td>
<td>Total visits = 362</td>
<td>P = 0.0388</td>
<td>Pass</td>
</tr>
<tr>
<td>(x = -8)</td>
<td>Total visits = 412</td>
<td>P = 0.0645</td>
<td>Pass</td>
</tr>
<tr>
<td>(x = -7)</td>
<td>Total visits = 413</td>
<td>P = 0.0479</td>
<td>Pass</td>
</tr>
<tr>
<td>(x = -6)</td>
<td>Total visits = 445</td>
<td>P = 0.0591</td>
<td>Pass</td>
</tr>
<tr>
<td>(x = -5)</td>
<td>Total visits = 504</td>
<td>P = 0.1208</td>
<td>Pass</td>
</tr>
<tr>
<td>(x = -4)</td>
<td>Total visits = 525</td>
<td>P = 0.1228</td>
<td>Pass</td>
</tr>
<tr>
<td>(x = -3)</td>
<td>Total visits = 547</td>
<td>P = 0.1192</td>
<td>Pass</td>
</tr>
<tr>
<td>(x = -2)</td>
<td>Total visits = 596</td>
<td>P = 0.2144</td>
<td>Pass</td>
</tr>
<tr>
<td>(x = -1)</td>
<td>Total visits = 658</td>
<td>P = 0.6435</td>
<td>Pass</td>
</tr>
<tr>
<td>(x = 1)</td>
<td>Total visits = 673</td>
<td>P = 0.9565</td>
<td>Pass</td>
</tr>
<tr>
<td>(x = 2)</td>
<td>Total visits = 692</td>
<td>P = 0.7893</td>
<td>Pass</td>
</tr>
<tr>
<td>(x = 3)</td>
<td>Total visits = 669</td>
<td>P = 0.9417</td>
<td>Pass</td>
</tr>
<tr>
<td>(x = 4)</td>
<td>Total visits = 614</td>
<td>P = 0.5303</td>
<td>Pass</td>
</tr>
<tr>
<td>(x = 5)</td>
<td>Total visits = 620</td>
<td>P = 0.6178</td>
<td>Pass</td>
</tr>
<tr>
<td>(x = 6)</td>
<td>Total visits = 663</td>
<td>P = 0.9215</td>
<td>Pass</td>
</tr>
<tr>
<td>(x = 7)</td>
<td>Total visits = 754</td>
<td>P = 0.5509</td>
<td>Pass</td>
</tr>
<tr>
<td>(x = 8)</td>
<td>Total visits = 851</td>
<td>P = 0.2161</td>
<td>Pass</td>
</tr>
<tr>
<td>(x = 9)</td>
<td>Total visits = 899</td>
<td>P = 0.1392</td>
<td>Pass</td>
</tr>
<tr>
<td>(x = 9)</td>
<td>Total visits = 899</td>
<td>P = 0.1392</td>
<td>Pass</td>
</tr>
</tbody>
</table>

Chi-squared evaluation is the standard for hypothesis testing and for checking the uniformity of the p-value to see if the generated sequences deviate from values expected for randomness. The $\chi^2$ for observed data, $O$, and expected data, $E$, is:

$$\chi^2 = \sum_{i} \frac{(O_i - E_i)^2}{E_i}$$ (9.3)

Here, $E_i$ is the number of expected occurrences, and $O_i$ is the number of observed occurrences for all $i$. Chi-square is the significance test for a significance level $\alpha$. 

144
Chapter 9. Testing the OTP Encoder Prototype

9.3.3 Prototype NIST Results

The uniformity of the prototype p-values for the proportion of the 100 bit sequences tested are in the second last column in Table 9.6.

<table>
<thead>
<tr>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
<th>C6</th>
<th>C7</th>
<th>C8</th>
<th>C9</th>
<th>C10</th>
<th>p-val</th>
<th>Pass</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>6</td>
<td>10</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>13</td>
<td>7</td>
<td>12</td>
<td>12</td>
<td>0.798</td>
<td>1</td>
<td>Frequency</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>5</td>
<td>10</td>
<td>11</td>
<td>10</td>
<td>14</td>
<td>12</td>
<td>10</td>
<td>14</td>
<td>0.455</td>
<td>1</td>
<td>BLFrequency</td>
</tr>
<tr>
<td>13</td>
<td>9</td>
<td>10</td>
<td>9</td>
<td>11</td>
<td>7</td>
<td>10</td>
<td>14</td>
<td>11</td>
<td>6</td>
<td>0.798</td>
<td>0.99</td>
<td>Runs</td>
</tr>
<tr>
<td>15</td>
<td>13</td>
<td>8</td>
<td>10</td>
<td>12</td>
<td>12</td>
<td>10</td>
<td>8</td>
<td>5</td>
<td>7</td>
<td>0.494</td>
<td>0.98</td>
<td>LongRun</td>
</tr>
<tr>
<td>11</td>
<td>8</td>
<td>10</td>
<td>14</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>10</td>
<td>12</td>
<td>14</td>
<td>0.637</td>
<td>1</td>
<td>Rank</td>
</tr>
<tr>
<td>10</td>
<td>12</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>14</td>
<td>8</td>
<td>10</td>
<td>7</td>
<td>6</td>
<td>0.798</td>
<td>1</td>
<td>DFT</td>
</tr>
<tr>
<td>13</td>
<td>13</td>
<td>8</td>
<td>9</td>
<td>8</td>
<td>10</td>
<td>14</td>
<td>9</td>
<td>13</td>
<td>3</td>
<td>0.334</td>
<td>0.98</td>
<td>NolapTemp</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>11</td>
<td>4</td>
<td>15</td>
<td>7</td>
<td>12</td>
<td>10</td>
<td>7</td>
<td>10</td>
<td>0.419</td>
<td>0.98</td>
<td>OvlapTemp</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.000</td>
<td>1</td>
<td>Universal</td>
</tr>
<tr>
<td>36</td>
<td>24</td>
<td>10</td>
<td>9</td>
<td>6</td>
<td>4</td>
<td>2</td>
<td>6</td>
<td>2</td>
<td>1</td>
<td>0.000</td>
<td>0.87</td>
<td>AppEntropy</td>
</tr>
<tr>
<td>12</td>
<td>6</td>
<td>9</td>
<td>8</td>
<td>15</td>
<td>14</td>
<td>9</td>
<td>10</td>
<td>8</td>
<td>9</td>
<td>0.616</td>
<td>1</td>
<td>Serial</td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td>8</td>
<td>14</td>
<td>8</td>
<td>15</td>
<td>8</td>
<td>10</td>
<td>11</td>
<td>8</td>
<td>0.719</td>
<td>0.99</td>
<td>CumSums</td>
</tr>
<tr>
<td>15</td>
<td>7</td>
<td>12</td>
<td>11</td>
<td>9</td>
<td>7</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>9</td>
<td>0.834</td>
<td>0.97</td>
<td>LinComplex</td>
</tr>
<tr>
<td>11</td>
<td>10</td>
<td>8</td>
<td>11</td>
<td>9</td>
<td>11</td>
<td>8</td>
<td>14</td>
<td>8</td>
<td>10</td>
<td>0.955</td>
<td>1</td>
<td>RanExcur</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>0.275</td>
<td>1</td>
<td>RanExcurVar</td>
</tr>
</tbody>
</table>

NIST recommends a Chi-square test by dividing the interval 0 to 1 in sub-intervals of ten producing the p-value in column 11 and column 12 is the proportion of tests passed. The minimum pass rate for each statistical test (except the random excursion variant test), is approximately 96 for a sample size of 100 binary sequences as recommended by NIST.

For greater histogram resolution the sample size sequences were set to 1000 sequences. The results for testing these sequences is shown in Table 9.7.

---

3The minimum pass rate for the random excursion variant test is undefined
9.3.4 Testing Prototype Chaos Circuits

Table 9.8 shows prototype tests results from individual Lorenz and Chua chaos circuits and delay variants of these chaos systems. It can be seen how adding a delay to each chaos circuit improves the pass rate but XORing the two circuits passes all the NIST tests.

Table 9.8: Comparing Lorenz and Chua systems.

<table>
<thead>
<tr>
<th>Test</th>
<th>Lorenz</th>
<th>DelayLorz</th>
<th>Chua</th>
<th>DelayChua</th>
<th>LorXORChua</th>
</tr>
</thead>
<tbody>
<tr>
<td>Freq</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>Bfreq</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>Runs</td>
<td>P</td>
<td>P</td>
<td>F</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>LRuns</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>P</td>
</tr>
<tr>
<td>Rank</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>DFT</td>
<td>P</td>
<td>P</td>
<td>F</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>NonOver</td>
<td>F</td>
<td>P</td>
<td>F</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>Overlap</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>Universal</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>Entropy</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>P</td>
</tr>
<tr>
<td>Serial</td>
<td>F</td>
<td>P</td>
<td>F</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>CSum</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>Lcompl</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>Rand</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>Randv</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>
Chapter 9. Testing the OTP Encoder Prototype

9.3.5 Histogram Testing Long Prototype sequences

A comprehensive assessment is to examine the distribution of the p-values by plotting the fifteen tests as histograms in Excel, as shown in Figure 9.6. Each test was done ten times for each 1000 bit sequence from the 158 Mbits available.

![Figure 9.6: Histogram of the p-values for 158 Mbit OTPs divided into 1000 bit sequences.](image)

This result shows the p-values for each test are displayed within a consistent range of values, verifying the OTP sequences are random.
9.4 Chapter Conclusions

This chapter examined NIST test results from the OTP prototype encoder. The output binary sequences from each chaos source were combined in different configurations, but the final solution was to take a binary digit from each chaos source and form dibits. Applying a VN algorithm to the dibits maximised the entropy in the final sequence which was then exported for randomness testing. Various randomness tests carried out on the prototype were discussed but the main test for randomness was the NIST suite of fifteen tests where each test had a certain length and parameter requirements, as discussed in 9.2.

A sequence of 157 Mbits from the prototype was divided into 1000 bits and tested using a NIST test executable program called ASSESS. This was repeated ten times and the data was exported to Excel, where the results were plotted to observe the uniformity of the p-values. Thus, the fifteen tests produced p-values (statistical probability bands of randomness) for short and large sequences and the uniformity of the p-values was consistent with very good randomness.

Another test was developed during the research to evaluate the entropy for the best Padé delay added to each chaos source. The delay value which maximised the entropy was found by varying the Padé potentiometer over a range of resistance values and observing the average of the analogue x-signal closest to the zero axis.
10 Conclusion and Future work

“Out of chaos comes order.” Friedrich Nietzsche

10.1 Conclusions and Future Work

The thesis aims and objectives involved methods for answering the research question, “How can stored Cloud data be made unreadable using chaos encoding?”. The research commenced with a review of methods for creating random binary sequences for encoding data locally by the client. The research question was finally answered by creating a prototype one-time pad (OTP) encoder which incorporated a novel application of an analogue Padé delay added to the two analogue chaos sources. Adding delays in this manner converted the third-order chaos sources to novel fourth-order hyper-chaos systems and produced binary sequences with greater entropy.

The VN algorithm applied to binary streams from each chaos source maximised the entropy of the final OTP. Furthermore, the Lorenz and Chua chaos oscillators were seeded by a novel sampled electronic noise from a 433 MHz data receiver to provide the necessary keys with which to start the two chaos sources. This initialising method ensured the chaos trajectories started from a random key each time they were generated and hence adversaries could not successfully cryptanalyse intercepted data. OTP encoding according to Vladimir Kotelnikov in 1941 and Claude Shannon in 1945, is a theoretically-perfect, and unbreakable system, if OTP rules are adhered to [Molotkov, 2006], [Sachkov, 2006].

Techniques using evolutionary computing for evolving noise-producing chaos-initialising functions from true noise were also investigated. Since noise is stochastic this is a highly speculative idea, nevertheless, evolved functions which, with some post-processing, produced pseudo-random binary sequences. More importantly, a prototype constructed from analogue circuits ensured the binary sequences produced had an infinite cycle length, unlike the previous digital platform method. This alternative method used a true noise source to produce true random binary sequences and was more effective at securing sensitive data and classified the encoder as a true source of randomness.
Currently, the main objection to OTP encoding for localised encryption of data before storing in the Cloud, concerns the distribution of the OTP between two parties, the so-called key-distribution problem (KDP). However, the research examined in this thesis proposed one-to-cloud specialised applications where this was not a problem because only the cloud client was involved with the encoding-decoding process.

A substantial part of the research applied the linear PSpice circuit simulator to exploit aspects of chaos for maximising the entropy of the OTP true random bit sequences (TRBS) for achieving true secrecy in the Cloud. This necessitated creating new simulation parts, simulation tools and meters, for examining phenomenon such as bifurcation to ensure the encoder operated in the chaos region. However, in this region, many PSpice convergence problems occurred because of the nonlinear nature of chaos systems but a simple novel application of PSpice parts enabled a range of system parameters to be examined. Furthermore, a novel application of PSpice VECT0R1 parts allowed multiple digital signals to be recorded and exported from PSpice to a text file. This allowed the four chaos digital binary streams to form a single OTP which was encoded using modulo-two arithmetic with the data using a JavaScript application.

Data from thirty chaos systems was exported from PSpice and processed in software downloaded from the Internet to calculate the Lyapunov dimension. This metric was used as an indicator of the source entropy to select two novel modified delay Chua and Lorenz chaos oscillator for the alpha prototype constructed on a double-sided PCB using a cloud-based facility in China. The major benefit of building a prototype was that extensive testing could be carried out on all fifteen tests of the internationally-accepted NIST tests for randomness. The encoder passed all the tests and produced excellent random OTP sequences, as is evidenced by the uniformity of the p-values histograms given in Chapter 9.

10.2 Future Work

Memristance, the new circuit element, was added to a Chua chaos oscillator and experimentation indicated it could have many potential applications in cryptography. Research on the application of memristance in cryptography will expand exponentially once memristors, which became available mid-2017, are affordable. This is a catch 22 situation because memristors will become affordable once industrial applications have been created. To become familiar with this device several memristance
models were simulated in PSpice. From a simulation, a circuit memristance emulator was constructed from standard electronic components and produced results which, by and large, agreed with the simulation data. Memristance might have a role in elliptic curve cryptography for protecting Internet of Things (IoT) devices but this is just speculation. Neuromorphic engineering is another research area for memristors which has great potential for addressing the semiconductor manufacturing Landauer limit. Within ten years Moore’s law will no longer apply and manufacturing technology will have reached this limit. However, memristor nanotechnology might extend the limit by overcoming the heat dissipation and density-packing problems.

The phase space of fractional-order chaos oscillators is theoretically infinite and should produce random signals with even greater entropy. In this research fractional-order Lorenz chaos systems with orders less than three were produced, but according to the Poincaré-Bendixson theorem, chaos cannot exist in continuous-time autonomous chaos systems for orders less than three, so maybe this theorem needs updating.

10.3 Final Comments

Two main objectives were successfully addressed in this thesis: (a) To develop new simulation paradigms, parts and tools for simulating nonlinear circuits, and, (b) To design and produce a one-time pad encoder prototype which incorporates two novel delay chaos sources mounted on a PCB for protecting sensitive data stored in the Cloud. From an initial mathematical idea, a prototype device was designed and built and produced binary OTP sequences which met all the requirements of the NIST suite of tests. Chaos technology has not developed as rapidly as it could and in this thesis new simulating parts, tools and methods for simulating chaos were developed which might encourage other researchers to create more chaos applications in cryptography. Finally, quantum computing will present problems to all cryptographic systems when the number of qubits in existing hardware is increased. The only encoding system to resist cryptanalysis by these computers is the unconditionally-secure OTP encoder [Bernstein and Lange, 2017].
Bibliography


Badger, Lee et al. (2012). “Cloud computing synopsis and recommendations”. In: NIST special publication 800, p. 146.


Bellovin, Steven M. (2016). “Vernam, mauborgne, and friedman: The one-time pad and the index of coincidence”. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 9100, pp. 40–66. ISSN: 16113349. DOI: 10.1007/978-3-662-49301-4_4.


Blackledge, Jonathan, A Al-Rawi, and P Tobin (2014). “Stegacryption of DICOM Metadata”. In:


Blackledge, Jonathan and Paul Tobin (2013). “Cryptography Using Evolutionary Computing”. In:


Carnahan, Lisa J and Miles E Smid (1994). “Security requirements for cryptographic modules”. In:


— (2011b). “The information: A history, a theory, a flood.” In:


Han, F, J Hu, and Kai Xi (2010). Highly efficient one-time pad key generation for large volume medical data protection. DOI: 10.1109/ICIEA.2010.5516860.

Hassard, B et al. (1994). “A computer proof that the Lorenz equations have “chaotic” solutions”. In: Applied Mathematics Letters 7.1, pp. 79–83.


Lorenz, EN (1972). “Does the flap of a butterfly’s wings in Brazil set off a tornado in Texas?, American Association for the Advancement of Science”. In: Global Atmospheric Research Program. 139th Meeting of the American Association for the Advancement of Science. Unpublished.


Scanlon, Mark and Nhien-An Le-Khac (2017). “16th European Conference on Cyber Warfare and Security ECCWS 2017”. In:


Tobin, P, J Blackledge, and S Bezobrasov (2016). “Personalised encryptor generation in the Cloud using Evolutionary Computing”. In:


Tobin, Paul and Jonathan Blackledge (2014). “Entropy, Information, Landauer ‘s limit and Moore’s law”. In:


Widrow, Bernard et al. (1960). Adaptive” adaline” Neuron Using Chemical” memistors.”.


Yang, Haibo and Mary Tate (2009). “Where are we at with cloud computing?: a descriptive literature review”. In: 20th Australasian conference on information systems, pp. 2–4.


A Padé Delay Test Results

A.1 Analogue Delay Tests

These are signal and spectral plots for the delay mechanism for maximum entropy using the averaging test developed. Taken’s attractor plots for delays corresponding to 200 ns to 1.2 us.

**FIGURE A.1:** Analogue delay range 200 ns to 1.2 us.

**FIGURE A.2:** Analogue delay spectrum 200 ns to 1.2 us. 

ing to 200 ns to 1.2 us.
Appendix A. Padé Delay Test Results

**Figure A.3**: Takens embedding Attractor $Z_v$ vs $Z_d$

**Figure A.4**: Taken’s Strange Attractor $Z_d$ vs $X$ for delay range 200 ns to 1.2 us

**Figure A.5**: Delay range 1 us to 6 us.
Appendix A. Padé Delay Test Results

Figure A.6: Takens attractors vs Delay.

Figure A.7: Large delay spectrum.

Figure A.8: Delay range 1 us to 6 us.
Appendix A. Padé Delay Test Results

**Figure A.9:** Delay spectra for 1 us to 6 us

**Figure A.10:** The average x plot for 0 to 0.8 us delay.

**Figure A.11:** The average x for 1 us to 1.4 us delays.
B  Additional Chaos Circuit Designs

B.1  Operational Amplifier Offset Voltages

The operational saturation and offset voltages are determined using the set-up in Figure B.1. The saturation voltage, $E_{\text{sat}}$, is required in order to calculate the breakpoints for the negative resistance. Sweeping $v_{\text{sweep}}$ from $-100 \mu\text{V}$ to $100 \mu\text{V}$, in steps of $1 \text{nV}$, produced the transfer characteristic in Figure B.2. The maximum and minimum saturation voltages, $E_{\text{sat}}$, and the offset voltage, $V_{\text{os}}$, which depends on the DC supply voltage and other factors, can be read from the characteristic.

![Figure B.1: Measuring operational amplifier bias characteristics.](image)

![Figure B.2: The operational amplifier voltage transfer function characteristic.](image)
Appendix B. Additional Chaos Circuit Designs

B.2 The AD633 4-Quadrant Multiplier

The AD633 4-quadrant multiplier implements the nonlinear chaos product terms and can be configured to achieve all four permutations of signals applied to the differential inputs (+ - + -). The device has an internal scaling factor of 0.1 to stop the output from becoming too large in the multiplication process. A 50 kΩ trim pot connected to the positive supply and attached to pin 6 (z-pin) ensures the output never saturates to the negative supply voltage.

The AD633 was used for signal multiplication in the logistic oscillator in Chapter 6 and the Lorenz and Chua circuits in Chapter 3. Other signals can be connected to pin 6, the Z-input pin, to add to the output signal at the W pin. For the Logistic map, the delayed output signal, $y_{nd}$ is applied to pin 1 and 4 to produce a negative sign on the product, but also it is applied to the potentiometer attached to the Z input pin. It is easier to analyse if the potentiometer is considered as two resistors $R_1$ and $R_2$. Let $W = 0$, then the signal applied to the input at the Z-input due to the feedback term, $y_{nd}$ is:

$$\frac{R_1}{R_1 + R_2} y_{nd} \quad \text{(B.1)}$$

The component in the $W$ output due to the Z-input (Set $y_{nd}$ to 0), is:

$$\frac{R_2}{R_1 + R_2} W \quad \text{(B.2)}$$

The output from the multiplier is the sum of the voltages from the three components, the input signal, the component from the $W$ out to the z input, and the z input from the input signal, i.e.,

$$W = - \frac{y_{nd} y_{nd}}{10} + Z = - \frac{y_{nd}^2}{10} + \frac{R_2}{R_1 + R_2} W + \frac{R_1}{R_1 + R_2} y_{nd} \quad \text{(B.3)}$$

The final output expression, $W$ for $R_1 = 10$ kΩ and $R_2 = 90$ kΩ is given as:

$$W = y_{nd}(1 - y_{nd}) \quad \text{(B.4)}$$

A small negative IC value will make the system move towards negative infinity to become unstable. Hence, it is necessary for the electronic design to include a small positive (millivolt) offset signal to encourage the trajectory at the FP to move in a positive direction.\(^1\)

\(^1\) If a trajectory moves away from an FP it becomes an unstable FP and is called a repellor
Appendix B. Additional Chaos Circuit Designs

of the FP. If the slope, $|df/dy|_{y^*} < 1$, it is an attracting FP (a sink), but is a repelling FP (a source) if the slope is greater than one. It was necessary to add a GAIN part to increase the loop gain to overcome the losses in the loop due to the transmission delay part. Setting the loop gain correctly ensures the first bifurcation occurs at $R = 3$. Increasing the loop gain moves the bifurcation point to the left of the correct location, but decreasing it moves the point past $R = 3$ to the right. The LIMIT part in the loop limits the signal from exceeding the values in the LIMIT part when $R$ is close to 4.

B.2.1 Active Integration

There are many designs for simulating electronic integration in Chaos oscillators: Integ ABM parts, active inverting integrators using a TL084 IC, and a non-inverting integrator using a constant current output from the AD633 z-pin. Figure B.3 shows an ABM INTEG model with gain of -10,000, and a second active integrator with a large resistor $R_2$ placed across the capacitor to stop the output saturating to the DC rail voltage. $R_2$ is very large and can be ignored when calculating the gain, which is calculated:

$$\text{Gain} = -\frac{1}{C_1R_1} = -\frac{1}{10^{-9}10^5} = -10^4$$

(B.5)

The non-ideal integrator:

This integrator works above the cut-off frequency, $f_{c2} = 1/(2\pi C_1R_2)$. $R_2$ is much bigger than $R_1$ and limits the DC gain to prevent small DC offsets from saturating
the output to the DC supply voltage. The circuit behaves like a DC amplifier below $f_c$, and the gain is approximately equal to $1/sC1R1$ for frequencies greater than the cut-off frequency. The gain for these components is $-(1/\tau) = 1/(1e5*2.5e-9) = 4000$. The period of the integrator should be about 0.2 of the period of the applied pulse width.

**B.2.2 Summing Integrator**

Figure B.4 is a summing integrator which sums and integrates the square and sine signals.

![Figure B.4: A summing integrator circuit.](image)

**B.3 Active Integration using the AD633**

The AD633 multiplier device can be configured as an integrator using the resistance $R1$ as shown in Figure B.5 which converts the output to a current source. Applying a square wave should be displayed as a triangle waveform at the output. A $10 \, \text{k}\Omega$

![Figure B.5: Configuring the AD633 as an integrator to implement the Lorenz chaos model.](image)
Appendix B. Additional Chaos Circuit Designs

resistance, \( R_1 \) connected between the \( w \) and \( Z \) pins makes pin 6 output a current source making the device a voltage-controlled integrator. The current out of the \( W \) pin is:

\[
i_0 = \frac{1}{R_1} \left[ \frac{(x_1 - x_2)(y_1 - y_2)}{10} \right]
\]  

(B.7)

To understand how the AD633 integrates, apply Kirchhoff’s current law to the currents at node \( y \) by summing the currents at that node and equating to zero. The input impedance at pin 6 is very high and draws negligible current, so the currents at node, \( y \) is:

\[
i_{R1} = i_{C1} + i_{R2}
\]  

(B.8)

The voltage at the output, \( w \), is the sum of the signals at the \( w \) output and the \( Z \) input, \( V_x \). Hence, the voltage across the current source resistance, \( R_{m1} \) is the difference between the \( w \) and \( X \) nodes, so the currents are:

\[
i_{Rm1} = \frac{(V_W + V_x) - V_x}{R_1} = \frac{10V_y}{R_{m1}} = \frac{V_y}{R_{m1}}, i_{C1} = C_1 \frac{dV_x}{dt}, \text{ and, } i_{R1} = \frac{V_x}{R_1}
\]  

(B.9)

For \( R_1 = R_{m1} \), substituting for the current, \( i_t \), gives:

\[
\frac{dV_x}{dt} = \frac{1}{C_1 R_{m1}} (V_y - V_x)
\]  

(B.10)

\[
\frac{dV_z}{dt} = \frac{V_x V_y}{10C R_{m1}} - \frac{V_z}{C R_{m2}}
\]  

(B.11)

For \( R_{m1} = 10 \, \text{k\Omega} \), \( R_{m2} = 100 \, \text{k\Omega} \), and \( C_1 = 1\text{nF} \).

\[
\frac{dV_z}{dt} = 10^4 (V_x V_y - V_z)
\]  

(B.12)

\[
\frac{dx}{dt} = P(x - y) = 10(x - y)
\]  

(B.13)

B.3.1 The High-Frequency Lorenz Chaos Generator

The high-frequency version of the AD633 is the AD734, multiplier which can be configured as an integrator. A design for producing high-frequency Lorenz chaos signals is shown in Figure B.6 and was designed by Jonathan Blakely [Blakely, Eskridge, and Corron, 2008].

\[
i_{Rm1} = \frac{(V_W + V_x) - V_x}{R_{m3}} = \frac{V_x V_y}{10 R_{m3}}, i_{C1} = C_1 \frac{dV_x}{dt}
\]  

(B.14)
Appendix B. Additional Chaos Circuit Designs

Figure B.6: A high-frequency Lorenz chaos oscillator.

\[ \frac{V_z V_y}{10R_{m3}} = C_1 \frac{dV_x}{dt} \Rightarrow \frac{dV_z}{dt} = \frac{V_z V_y}{10C_1 R_{m3}} \]  
(B.15)

B.4 Gyrator Design

Figure B.7 shows the Tellegen inductive gyrator emulator for greater flexibility. The inductance value is calculated:

\[ L = \frac{R_1 R_3 R_4 C_1}{R_2} \]  
(B.16)

This gyrator design is limited to application where the inductance is grounded at one end. Figure B.8 is an inductive reactance plot for a range of resistance values.
Appendix B. Additional Chaos Circuit Designs

The gyrator equivalent series resistance is very low and results in a high Q-factor. An inductance of 18 mH from the reactance plot corresponds to 6 kΩ resistance calculated in equation (B.17):

\[ R_4 = \frac{R_2 L_1}{R_1 R_3 C_1} = \frac{7.5 \times 10^3 \times 18 \times 10^{-3}}{3.10^2 \times 7.5 \times 10^3 \times 10^{-9}} \]  

(B.17)

The gyrator inductance was tested with a capacitor as shown in Figure B.9. The resonant frequency of the parallel-tuned circuit is given approximately:

\[ f_0 \approx \frac{1}{2\pi \sqrt{L_1 C_2}} = \frac{1}{2\pi \sqrt{18 \times 10^{-3}}} = 3750 \text{Hz} \]  

(B.18)

The reactance of the 18 mH inductance at this frequency is:

\[ X_L = 2\pi f_0 L = 2\pi 3750 \times 18 \times 10^{-3} = 424 \Omega \]  

(B.19)
The -3 dB bandwidth measured at 15.6 Hz has a Q-factor $Q = \frac{f_0}{BW} = \frac{3.751 \text{ kHz}}{15.6} = 240$, which can be verified $Q = \frac{X_L}{R_{coil}}$ from the frequency response. The current in the coil at resonance is $Q$ times the input current. Since the maximum current is 240 mA, and the input is 1 mA, therefore, $Q = 240$.
C Digital Maps

C.1 The Baker Map

Digital maps not considered in Chapter 6 are now examined as possible sources of OTPs in future encoders. The Baker map (BM) schematic in Figure C.1 produced the strange attractor shown in Figure C.2.

**Figure C.1**: The Baker Map schematic.

**Figure C.2**: The Baker Map attractor.
Appendix C. Digital Maps

C.2 The Bernoulli Map

The Bernoulli map schematic in Figure C.3 produced the attractor in Figure C.4.

![Bernoulli Map Schematic](image)

**Figure C.3:** The Bernoulli map schematic.

![Bernoulli Attractor](image)

**Figure C.4:** A Bernoulli attractor.
C.3 The Lozi Map

The Lozi digital map schematic in Figure C.5 plots the attractor in Figure C.6.

\[
\begin{align*}
n_{n+1} &= 1 - a |x_n| + y_n \\
y_{n+1} &= bx_n \\
\end{align*}
\]

PARAMETERS:
DELAY = 100u
a = 1.7
b = 0.4

For producing a ramp

Figure C.5: The Lozi map schematic.

Figure C.6: The Lozi attractor.
Appendix C. Digital Maps

C.4 The Gaussian Map

The Gaussian digital map schematic in Figure C.7 produces the bifurcation plot in Figure C.8.

The Gaussian map orbit is plotted in Figure C.9 using a single value for beta.
C.5 The Tinkerbell Map

The Tinkerbell chaos map is plotted using the schematic in Figure C.10.

\[
x(n) = x(n-1)^2 - y(n-1)^2 + ax(n-1) + by(n-1)
\]
\[
y(n) = 2x(n-1)y(n-1) + cx(n-1) + d\cdot y(n-1)
\]

**Figure C.10:** The Tinkerbell schematic.

**Figure C.11:** Tinkerbell noise.
D Book List and Arduino Code

Textbooks Consulted

- Acheson D. From calculus to chaos: An introduction to dynamics. Oxford University Press on Demand; 1997.
Appendix D. Book List and Arduino Code

D.1 Arduino Code

```c
#define switchPin 9
#define lorenzSetPin 10
#define lorenzResetPin 11
#define chuaSetPin 12
#define chuaResetPin 13
#define chunkSize 25
#define bufferSize 256 // Don't make this too large!

/*
At the command prompt:
  mode com3:115200
  type com3
*/

// Initial values -- have to be different than (HIGH or LOW)
int lorenzSet = -1;
int lorenzReset = -1;
int chuaSet = -1;
int chuaReset = -1;

int lorenzOutput;
int chuaOutput;
int exor[chunkSize]; // holds the XOR values
int temp; // loop var

int outLorIndex = 0;
int outLor[bufferSize];

int outChuaIndex = 0;
int outChua[bufferSize];

void setup() {  
  pinMode(switchPin, INPUT);
  digitalWrite(switchPin, HIGH);

  // make them inputs
  pinMode(lorenzSetPin, INPUT);
  pinMode(lorenzResetPin, INPUT);
  pinMode(chuaSetPin, INPUT);
  pinMode(chuaResetPin, INPUT);
  // setup serial port
  Serial.begin(115200);
}

void loop() {  
  if (digitalRead(switchPin) == 0) { // If we want chaos (switch is closed)
    // Read new values from device
    lorenzSet = readChaos(lorenzSetPin, lorenzSet);
    lorenzReset = readChaos(lorenzResetPin, lorenzReset);
    chuaSet = readChaos(chuaSetPin, chuaSet);
    chuaReset = readChaos(chuaResetPin, chuaReset);
  }
}
```

185
Appendix D. Book List and Arduino Code

// If there's at least a chunk in each output...
if (outLorIndex > (chunkSize - 1) && outChuaIndex > (chunkSize - 1)) {
    for (temp = 0; temp < chunkSize; temp++) {
        exor[temp] = outLor[temp] ^ outChua[temp];
    }
    /*
     * von Neumann corrector
     * For each input pair:
     * If the input is "00" or "11", the input is discarded (no output).
     * If the input is "10", output a "1".
     * If the input is "01", output a "0".
     * eg:
     * i/p 0111001101
     * o/p 00
     */
    for (temp = 0; temp < chunkSize - 2; temp = temp + 2) {
        if (exor[temp] != exor[temp + 1]) { // if different
            if (exor[temp] == 1) {
                Serial.write("1\n");
            } else {
                Serial.write("0\n");
            }
        }
    }
}
// Move over array values to overwrite old values used (consume the arrays)
memcpy(&outLor[0], &outLor[chunkSize - 1], (bufferSize - chunkSize) * sizeof(*outLor));
outLorIndex = outLorIndex - chunkSize;
memcpy(&outChua[0], &outChua[chunkSize - 1], (bufferSize - chunkSize) * sizeof(*outChua));
outChuaIndex = outChuaIndex - chunkSize;
}
}

int readChaos(int pin, int prevVal) {
    int val = digitalRead(pin); // Read the data from the pin
    // If value is different from previous
    if (val != prevVal) {
        if (val == HIGH) { // If it's high put a value into the output array
            switch (pin) {
            case lorenzSetPin:
                if (outLorIndex <= (bufferSize - 1)) // check don't overfill the array
                    outLor[outLorIndex++] = 1;
                break;
            case lorenzResetPin:
                if (outLorIndex <= (bufferSize - 1))
                    outLor[outLorIndex++] = 0;
                break;
            case chuaSetPin:
                break;
            } // end switch
        } // end if
    } // end if
} // end readChaos
if (outChualIndex < (bufferSize -1))
    outChua[outChualIndex++] = 1;
break;
case chuaResetPin:
    if (outChualIndex < (bufferSize -1))
        outChua[outChualIndex++] = 0;
    break;
    }
}
return val; // Return whatever the read value was