Secure Communications based on Chaotic Systems

by

Mohamed Haroun
B.Sc., Alexandria University, 1999
M.Sc., Alexandria University, 2009

A Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Electrical and Computer Engineering

Secure Communications based on Chaotic Systems

by

Mohamed Haroun
B.Sc., Alexandria University, 1999
M.Sc., Alexandria University, 2009

Supervisory Committee

Dr. T. Aaron Gulliver, Supervisor
(Department of Electrical and Computer Engineering)

Dr. Mihai Sima, Departmental Member
(Department of Electrical and Computer Engineering)

Dr. Andrew Rowe, Outside Member
(Department of Mechanical Engineering)

Supervisory Committee

Dr. T. Aaron Gulliver, Supervisor
(Department of Electrical and Computer Engineering)

Dr. Mihai Sima, Departmental Member
(Department of Electrical and Computer Engineering)

Dr. Andrew Rowe, Outside Member
(Department of Mechanical Engineering)

ABSTRACT

This dissertation provides methods to utilize chaos efficiently in secure communications. Chaos has many desirable characteristics such as ergodicity and sensitivity to initial conditions, and is considered an ideal candidate for use in cryptography and secure communications. On the other hand, it suffers from sensitivity to noise and fading if it is used for physical layer transmission, and errors due to the finite precision of the numerical algorithms in digital systems. This limits the use of chaos in cryptographic applications. Accordingly, this dissertation proposes new algorithms to enhance the security of modern communication systems using chaos. The focus is on developing chaotic cryptosystems for wireless systems that are reliable, secure, and have good performance.

Contents

List of Tables

List of Figures

List of Abbreviations

AES	Advanced encryption standard
AWGN	Additive white Gaussian noise
BPSK	Binary phase shift keying
CCDF	Complementary cumulative distribution function
CDF	Cumulative distribution function
CF	Crest factor
COOK	Chaos on-off keying
CPU	Central processing unit
CSK	Chaos shift keying
CTPNCM	Coupled two-dimensional piecewise non-linear chaotic map
DCSK	Differential chaos shift keying
DFT	Discrete Fourier transform
DS-SS	Direct sequence spread spectrum
DV	Difference vector
FFT	Fast Fourier transform
FM-DCSK	FM-differential chaos shift keying
FPGA	Field programmable gate array
GSR	Gram-Schmidt reorthonormalization
HPA	High power amplifier
ICI	Intercarrier interference
IFFT	Inverse fast Fourier transform
ISI	Intersymbol interference
KDP	Key disagreement probability
KGR	Key generation rate
MIMO	Multiple-input multiple-output
OFDM	Orthogonal frequency division multiplexing
PAPR	Peak-to-average power ratio
PRNG	Pseudo-random number generator
QAM	Quadrature amplitude modulation
QPSK	Quadrature phase shift keying
RK-4	Fourth order Runge-Kutta
RSS	Received signal strength
SER	Symbol error rate
SLM	Selected mapping
SNR	Signal-to-noise ratio
UWB	Ultra wideband
WSS	Wide-sense stationary

ACKNOWLEDGEMENTS

I am grateful to **Allah**, for good health, loving parents, and my beautiful family who were supportive and instrumental in me completing this dissertation.

I am thankful to many sources that have contributed to this work, from direct advisement on the research, to financial support. First, I wish to express my sincere thanks to Dr. T. Aaron Gulliver whose expertise, understanding, and patience added considerably to my graduate experience. Second, I am also indebted to my supervisory committee: Dr. Mihai Sima, and Dr. Andrew Rowe for their insightful comments and encouragement. Finally, I would like to thank the government of Egypt for scholarship funding.

Chapter 1

Introduction

1.1 Chaos Theory

Chaos is a natural phenomenon that provides the very interesting property of sensitivity to initial conditions [1]. Chaos has been found to occur in a great number of non-linear dynamic systems, and in frequency ranges from baseband to optical. Chaos is the irregular motion of a dynamical system; it is deterministic, sensitive to initial conditions, and impossible to predict in the long term. It is neither harmonic nor random. Chaos is characterized by the way a dynamical system does not repeat itself, even though the system is governed by deterministic equations.

In the same way that time and the frequency are used to identify chaotic signals, phase-plane and correlation are used to identify the attractor and randomness of the chaotic system. The attractor is a region of the state space from which there are no exit paths. That is, points that get close enough to an attractor remain close even if they are slightly disturbed. Attractors can consist of a single state called an equilibrium state, or a cycle of states called a limit cycle. For chaotic systems, the attractor does not settle to one of these but explores all of the state space around the attractor for all time without ever repeating. That is, it does not return to some previously visited point in the state space, this describes the stretching and folding properties [2], which can be seen when plotting the states of the system against each other. Figure 1.1 plots the trajectory of the Rossler attractor in the phase space, depicting the stretching and folding properties.

In addition, chaotic signals have a broadband continuous frequency spectrum, which explains the noise-like behavior of chaotic systems [3], which can be illustrated

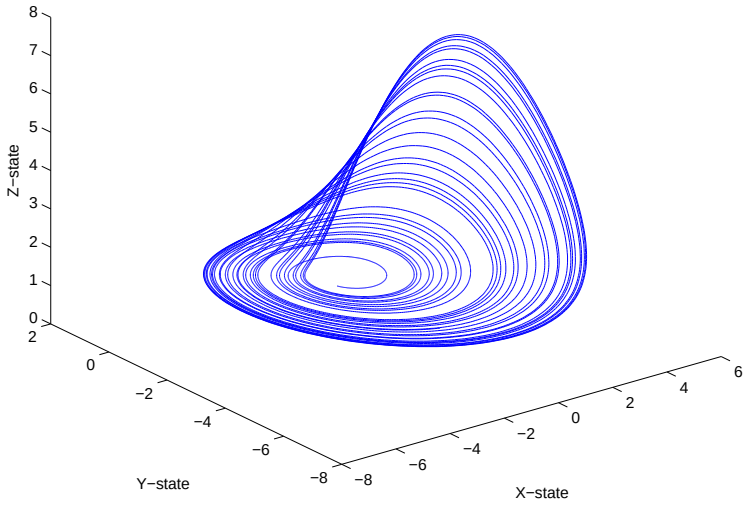


Figure 1.1: The Rossler attractor.

using the correlation function. The word attractor is used to identify the chaotic dynamical system, while generator is used when referring to the role of the chaotic attractor in the algorithm.

In general, chaotic systems can be classified into two main categories: continuous and discrete. For continuous time systems, for the chaotic trajectory to be non-repetitive the system must have a minimum of three differential equations, or two differential equations plus a forcing function. The most famous continuous chaotic systems include Chua's circuit [4], and the Lorenz [1] and Rossler [5] attractors. All these systems are three-dimensional (3D). For discrete systems, one difference equation is sufficient to achieve chaotic behavior. The Logistic map is a one-dimensional (1D) system [6], while the Henon map [7] is a two-dimensional (2D) system. Higher dimensional continuous and discrete systems have been developed based on these systems.

The synchronization of chaotic systems has attracted considerable interest since the pioneering work of Pecora and Carroll [8]. They showed that the dynamics of a drive system and of a driven subsystem (response system), become synchronized if the Lyapunov exponents of the response system are less than zero. This drive-response concept has led to the development of several methods for obtaining synchronized

dynamics in both continuous and discrete chaotic systems. Approaches proposed to achieve synchronization include observer-based [9], linear and non-linear feedback control [10, 11], adaptive control [12], backstepping design [13], active sliding mode [14], projective synchronization [15], anti-synchronization [16], and lag synchronization [17].

Once the problem of synchronization was solved, chaotic systems were proposed for use in wireless communications. Two methods have been used to create broadband signals using chaos. One uses the chaotic signal as the carrier signal [18], while the second approach employs the chaotic signal for spreading in direct sequence spread spectrum (DS-SS) system [19], where systems use continuous pseudo-random time series to spread the spectrum of message signal, and the spread signal is then directly sent through a channel to the receiver. Results have been presented concerning chaotic modulation and demodulation [20] and channel coding [21].

1.2 Chaos in Secure Communications

Chaotic cryptosystems are an important application of chaos in communication systems. Based on Shannon's theory of secrecy, confusion and diffusion are two properties used to make ciphers robust against statistical analysis [22]. For non chaos-based cryptosystems (such as AES which is now used worldwide), the cipher goes through a number of rounds of substitution and transposition operations to achieve these two properties. On the other hand, the complex behaviour of the chaotic system provides these properties directly [23]. The ergodicity of the chaotic signal corresponds to the confusion, as all chaotic signals generated using different initial values have the same statistics. In addition, the diffusion is achieved by the sensitivity to the initial conditions and the mixing property [24]. As a result, chaos-based cryptosystems can be used to develop strong ciphers with a simple structure, and the confusion and diffusion properties are realized through the dynamics of the system. Thus, chaos is an ideal candidate for use in cryptographic systems. This dissertation proposes chaotic cryptosystems with simple structures which provide security comparable to that of non chaos-based cryptosystems, as well as fast encryption rates.

Analog chaos-based communications is the first chaotic cryptosystem. It can be achieved by synchronizing chaotic systems at the transmitter and receiver driven by one or more analog chaotic signals which are transmitted through the physical channel. The outputs of these systems can be used for both analog and digital

communications.

Several types of analog chaotic cryptosystems have been developed. Both the masking and chaos shift keying are considered the first generation of the analog chaos-based communications. With chaotic masking, the message is added to the output of the chaotic generator at the transmitter [25]. The message signal is typically 20 to 30 dB weaker than the chaotic signal in order to hide the message and achieve synchronization at the receiver [20]. At the receiver, the chaotic signal is subtracted to recover the message. Several attacks on this cryptosystem have been developed [26], which make chaotic masking insecure.

Communication systems have been developed to transmit digital data using chaos shift keying (CSK), chaos on-off keying (COOK), differential chaos shift keying (DCSK), and FM-differential chaos shift keying (FM-DCSK). With these modulation techniques, the message (typically binary) is used to select the signal to be transmitted from two or more chaotic systems [20]. At the receiver, the received signal is used to drive chaotic subsystems identical to those at the transmitter. Each subsystem is identical to one of the transmitter systems, and so one will be synchronized with the transmitted binary symbol while the other will remain unsynchronized. The message is recovered by low-pass filtering and then using a threshold on the synchronization error signal. Successful attacks on this approach have been developed [27].

Chaotic modulation is considered to be the second generation of analog chaotic cryptosystems. Two methods have been proposed to modulate the messages. The first is called chaotic parameter modulation [28], and is based on using a message to modulate one or more parameters of the chaotic system. The second method is called chaotic non-autonomous modulation [29], where the message is injected into the dynamics of the chaotic system. Techniques have been suggested to break chaotic parameter modulation [30].

The third generation of analog chaotic cryptosystems provides a much higher level of security than the first two generations [31]. The first approach combines a traditional cryptographic technique with a chaotic system for synchronization. The message is encrypted using a conventional cipher with a key signal generated by a state variable of the chaotic system. The resulting signal is used to drive the chaotic system such that the chaotic dynamics are changed continuously in a very complex way. Another state variable of the chaotic system is used as the transmitted signal. The second approach uses higher order chaotic systems (called hyper chaotic), to increase the complexity and the key space.

Digital chaotic cryptosystems use one or more discrete chaotic systems (chaotic maps) directly to provide security rather than via chaotic synchronization as in analog cryptosystems. Digital processors are employed with the chaotic maps implemented using finite precision arithmetic to encrypt the messages. Most of these systems are based on chaos-based pseudo-random number generators (PRNGs). These numbers can be generated using floating-point (e.g. double-precision), then a binary key is extracted using quantization function. The initial conditions and control parameters play the role of the secret key, which exploit the large parameter space, strong sensitivity to initial conditions, and the random-like behavior of the resulting chaotic signals. Chaos has also been used for image encryption using the permuting mechanism of the chaotic generators [32, 33]. Generally, the pixels of the image are consider as elements of a matrix. The image is encrypted by permuting the pixels in non-predictable manner. At the receiver, the image is retrieved by applying inverse permutation on the ciphered image.

1.3 Contributions

As the behavior of chaotic system is sensitive to the initial conditions, any disturbance however small will grow exponentially, and leads to a different trajectory over time. In communication systems, the signal is transferred from the transmitter to the receiver through a channel. Synchronization is the key to using chaos in communications and cryptography applications. However, the channel and the receiver noise affect this synchronization, which makes it hard to establish reliable communications between users using chaos. As well, implementing chaos using discrete maps either for communications or encryption purposes is subject to errors due to finite precision arithmetic [34]. This dissertation explores how to overcome these barriers, and develops reliable algorithms that can offer security for communications with good performance. These algorithms suggest new solutions in the physical layer and higher layers such as the presentation and data link layers, and contribute to the use of chaotic cryptosystems as an alternative for effective and dependable security. This dissertation consists of two parts that are outlined below:

I The first part focuses on chaotic cryptosystems in the higher layers of digital systems. Ciphers are developed to encrypt digital data using high-dimensional chaotic systems. The problem of finite precision arithmetic is overcome and the

computational complexity is low. Chapters two, three and four present the first and second contributions.

II The second part provides security in the physical layer of the wireless communications. Based on the characteristics of the wireless channel and the frequency of the chaotic signal, the third and fourth contributions are presented in Chapter five and six respectively.

The contributions of this dissertation are as follows:

1. A new class of discrete chaotic systems based on 3D continuous systems is developed. These discrete systems provide Lower computational complexity compared with existing methods when implemented in digital hardware/software.

2. Two new chaotic ciphers are developed. These ciphers depend on the complex dynamic behavior of chaotic systems to provide fast and simple encryption. Further, the problem of finite precision arithmetic in numerical computations is overcome.

3. A new algorithm to extract shared key between two users is developed. This algorithm benefits from the frequency characteristics of the chaotic signal and the fading channel to generate random sequences of bits for secure communications. The use of the frequency characteristics makes the algorithm superior to time-domain based algorithms in terms of noise sensitivity and key generation rate.

4. A secure transmission technique for orthogonal frequency division multiplexing (OFDM) is developed. The phase of the chaotic signal is used to manipulate the signal constellation of the transmitted signal. In addition to providing security, the randomness of the chaotic phase signals is used to reduce the peak to average power ratio (PAPR) with full spectrum efficiency.

1.4 Thesis Organization

Chapter 1 briefly introduces the concept of chaos theory, and the motivation of using chaos in secure communications for analog and digital communications. In addition, it gives a quick look at the problems of using chaos in secure communications. The chapter ends with the dissertation contributions and organization.

Chapter 2 presents a new class of high-dimensional discrete chaotic systems. The transformation from continuous to discrete form results in new control parameters. The chaotic behavior of the systems is verified, and the range of each new parameter to preserve the chaotic behavior is defined. The low computational complexity is verified by comparing with the computational complexity of the corresponding continuous chaotic systems.

Chapter 3 provides a new scheme to encrypt two different digital data streams. The cipher is based on a 3D discrete Lorenz generator. The chapter introduces the cipher, verifies the randomness of the transmitted encrypted signal, the security, and performance of the cipher.

Chapter 4 presents an image cipher. Similarly to the cipher in Chapter 3, the proposed cipher uses the 3D discrete Lorenz generator which has a complex chaotic signal and low computational complexity. The cipher offers high speed encryption with good security, and overcomes the problem of finite precision arithmetic of the digital hardware and software. The performance and security are analyzed, and a comparison with previous results in the literature is performed.

Chapter 5 introduces a new technique to achieve secure wireless communication using physical layer security. A shared key between two legitimate users is generated exploiting the reciprocity of the fading channel between two points in free space. The performance according to the key generation rate (KGR), the key disagreement probability (KDP), and the key randomness is examined. The robustness against timing error and the signal-to-noise ratio (SNR) is verified. This shows the superiority of using frequency characteristics of the probing signal over other types of signal characteristics employed in the literature.

Chapter 6 introduces a secure OFDM system with PAPR reduction. The chaotic phase randomness of the chaotic signal is used to provide PAPR reduction as well as security. There is no need for side-information to be sent to the receiver as in the literature, which preserves the bandwidth efficiency and increases the security. The performance with different modulation techniques and different SNRs is illustrated.

Chapter 7 presents some conclusions and suggestion for future work.

Chapter 2

Low Complexity Discrete Multi-dimensional Chaotic Generators

2.1 Introduction

For secure communications using chaos, it has been recommended that high-dimensional systems be used rather than those of low-dimensional [35]. Since continuous chaotic systems have a complex dynamic behavior, many chaotic communication systems have been proposed based on analog circuits [36, 37, 38]. One deficiency of these systems is that both the transmitter and receiver must be constructed using very accurate components to ensure synchronization and data recovery. In practice, component accuracy can be insufficient due to effects such as aging, temperature and manufacturing variations. Thus analog solutions can be very difficult to implement [39], even for short periods of time under controlled laboratory conditions.

In modern digital communications, discrete chaotic systems are used for encryption purposes. Digital chaotic cryptosystems use one or more chaotic maps to provide security directly rather than via chaotic synchronization as in analog cryptosystems. These digital chaotic systems are 1D and 2D systems. Even though the logistic map is only a 1D system, it has been widely used to encrypt images and data in digital communication systems due to its simplicity. To enhance the security, continuous chaotic systems are implemented in digital hardware and/or software using approximation methods. Runge-Kutta is the most commonly used approximation method,

such as in [40], where it is used to approximate the 3D Lorenz attractor for real-time image encryption. However, the computational complexity is a drawback for implementing continuous chaotic systems, and hence the applicability of using it in chaotic cryptosystems.

In this chapter, new 3D discrete systems based on 3D continuous systems are developed. The chaotic behavior and synchronization of the resulting systems are verified. This is done using both Lyapunov exponents and randomness. The objective of the proposed approach is to develop chaotic systems that can be implemented simply and accurately. While the differential equations for dynamic system are solved using approximation methods, the proposed approach employs discrete expression for the integration to obtain new discrete systems. Additionally, the difference equations of the new systems have additional parameters which enhance the security level for the chaotic cryptosystems by increasing the key length. This provides reliable and secure communications.

2.2 Difference Equations of the New Class of Discrete Chaotic Generators

The well-known fourth order Runge-Kutta numerical integration method RK-4 [41], is widely used to simulate first order differential equations. It is an extension of the Euler method which provides greater accuracy [42]. RK-4 is frequently used to simulate continuous dynamic systems using digital hardware and is given by

$$Y_{i+1} = Y_i + h(a_1K_1 + a_2K_2 + a_3K_3 + a_4K_4) \tag{2.1}$$

where h is the step size, and K_1 to K_4 are parameters which depend on the previous one, This method is computationally expensive, as K_1 to K_4 must be calculated each iteration. In addition, these computations are performed sequentially. This results in an increase in the execution time. The objective here is to develop discrete systems based on continuous chaotic systems, resulting in fewer computations, lower execution times and smaller circuits than existing solutions in the literature [43].

The finite difference approximation for derivative is

$$\dot{x} = \frac{dx}{dt} \approx \frac{X_{n+1} - X_n}{\triangle t} \tag{2.2}$$

where $\triangle t$ is the step time and n is an integer. Equation (2.2) is called a forward difference approximation. This time step is used in numerical approximation methods to simulate the integration process in software and digital hardware. In contrast, the proposed systems are discrete and thus do not employ a time step. Thus the solution of (2.2) is

$$X_{n+1} = \dot{x} \times g + X_n \tag{2.3}$$

where g is the gain. This transformation from a continuous to a discrete system results in the new state X_{n+1} being the sum of the previous state X_n and the present transition $\dot{x}$ multiplied by gain g. This can be implemented as shown in Figure 2.1, where D is a delay by one sample.

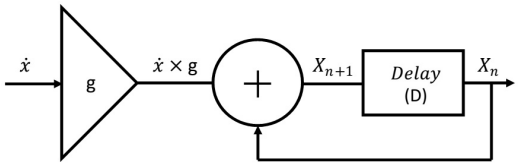


Figure 2.1: Discrete implementation of (2.3).

Equation (2.3) and its implementation in Figure 2.1 can be used to transform continuous dynamical systems to obtain new discrete systems. For chaotic systems, it must be determined if these new discrete dynamical systems have chaotic behavior. This will be determined in the next section. Well-known continuous chaotic systems such as those by Rossler [5] and Lorenz [1] can be converted to discrete systems by substituting (2.3) in the corresponding state equations. The Rossler state equations are

$$\begin{aligned} \dot{x} &= -y - z \\ \dot{y} &= x + Ay \\ \dot{z} &= B + Z(x - C) \end{aligned} \tag{2.4}$$

where $A = 0.398$, $B = 2$ and $C = 4$. Substituting (2.3) in the differential equations in (2.4) gives the difference equations

$$\begin{aligned} X_{n+1} &= g_1(-Y_n - Z_n) + X_n \\ Y_{n+1} &= g_2(X_n + AY_n) + Y_n \\ Z_{n+1} &= g_3(B + Z_n(X_n - C)) + Z_n \end{aligned} \tag{2.5}$$

The Lorenz state equations are

$$\begin{aligned} \dot{u} &= (v-u)A \\ \dot{v} &= Bu - v - 20uw \\ \dot{w} &= 5uv - Cw \end{aligned} \tag{2.6}$$

where $A = 10$, $B = 28$ and $C = 8/3$. Substituting (2.3) in (2.6), the resulting difference equations are

$$\begin{aligned} U_{n+1} &= g_1(V_n - U_n)A + U_n \\ V_{n+1} &= g_2(BU_n - V_n - 20U_nW_n) + V_n \\ W_{n+1} &= g_3(5U_nV_n - CW_n) + W_n \end{aligned} \tag{2.7}$$

These discrete models are investigated in the following sections to verify the chaotic behavior and synchronization.

The advantage of the proposed discrete systems can be clearly seen by comparing them with the corresponding Runge-Kutta based approximations of the continuous systems. For example, using the RK-4 method given in (2.1) with the Lorenz state equations in (2.6) gives

$$\begin{aligned} K_{11} &= h(V_i - U_i)A \\ K_{12} &= h(BU_i - V_i - U_iW_i) \\ K_{13} &= h(U_iV_i - CW_i) \\ K_{21} &= h[(V_i + \tfrac{1}{2}K_{12}) - (U_i + \tfrac{1}{2}K_{11})]A \\ K_{22} &= h[B(U_i + \tfrac{1}{2}K_{11}) - (V_i + \tfrac{1}{2}K_{12}) - (U_i + \tfrac{1}{2}K_{11})(W_i + \tfrac{1}{2}K_{13})] \\ K_{23} &= h[(U_i + \tfrac{1}{2}K_{11})(V_i + \tfrac{1}{2}K_{12}) - C(W_i + \tfrac{1}{2}K_{13})] \\ K_{31} &= h[(V_i + \tfrac{1}{2}K_{22}) - (U_i + \tfrac{1}{2}K_{21})]A \\ K_{32} &= h[B(U_i + \tfrac{1}{2}K_{21}) - (V_i + \tfrac{1}{2}K_{22}) - (U_i + \tfrac{1}{2}K_{21})(W_i + \tfrac{1}{2}K_{23})] \\ K_{33} &= h[(U_i + \tfrac{1}{2}K_{21})(V_i + \tfrac{1}{2}K_{22}) - C(W_i + \tfrac{1}{2}K_{23})] \\ K_{41} &= h[(V_i + K_{32}) - (U_i + K_{31})]A \\ K_{42} &= h[B(U_i + K_{31}) - (V_i + K_{32}) - (U_i + K_{31})(W_i + K_{33})] \\ K_{43} &= h[(U_i + K_{31})(V_i + K_{32}) - C(W_i + K_{33})] \\ U_{i+1} &= U_i + \tfrac{1}{6}(K_{11} + 2K_{21} + 2K_{31} + K_{41}) \\ V_{i+1} &= V_i + \tfrac{1}{6}(K_{12} + 2K_{22} + 2K_{32} + K_{42}) \\ W_{i+1} &= W_i + \tfrac{1}{6}(K_{13} + 2K_{23} + 2K_{33} + K_{43}) \end{aligned} \tag{2.8}$$

Comparing (2.7) with (2.8) shows the advantage of the proposed method in terms of complexity. In addition, the proposed method provides new discrete chaotic systems rather than approximations of continuous systems.

2.3 Performance and Synchronization

Simulink is a simulation tool based on MATLAB which can be used to simulate linear or non-linear, continuous or discrete, dynamic systems. The Simulink models for the proposed discrete Rossler and Lorenz attractors are established based on Equations (2.5) and (2.7), respectively. Each of these models has six parameters which will affect the behavior in addition to the parameters associated with the original attractor. These new parameters are the gains g_1, g_2 and g_3, and the initial values of the three delay units representing the initial state of the system.

Simulating continuous chaotic systems requires that an appropriate step size be chosen. This is a hidden parameter within the system model (typically set to 0.01 s), used to obtain the desired behavior. Conversely, the gain parameters in a discrete system are explicitly present as parameters, and can be chosen to vary the system behavior in a controlled way. Thus, the ranges of these values which result in chaotic behavior are investigated here.

Synchronization can be achieved even if the initial values differ at the transmitter and receiver, so the choice of the initial values is not critical. However, in applications such as chaos-based cryptography the initial conditions can play a critical role. In this case, the discrete models are preferable as these values can be defined precisely.

Lyapunov exponents are a mean of checking the stability of dynamic systems, and to determine if they are chaotic. They provide the average exponential rate of divergence or convergence of nearby orbits in the phase space. There are two means of determining the chaotic behavior of a dynamic system. The first employs the Jacobian matrix to determine the Lyapunov spectrum, while the second calculates the largest Lyapunov exponent to establish if the system behavior is chaotic. The latter approach depends on the state variables and not the Jacobian matrix. In each iteration, the deviation is determined between two orbits obtained using the same initial conditions but with a small permutation. The first approach considers the growth and change of an orthogonal set of vectors over the system iterations. For a linear system, the Lyapunov spectrum can be calculated directly because the Jacobian matrix is constant (i.e., independent of the state variables). Conversely,

the Lyapunov spectrum for non-linear continuous systems can be determined using approximate numerical methods. Several techniques have been employed to determine this spectrum.

The accuracy of the solution obtained depends on the method employed, the dimension of the system, the system parameters, the size of the data set, the number of output values discarded at the start, the step size, and the technique used for integration [44, 45, 46].

For 3D chaotic systems, the Lyapunov spectrum consists of three exponents. For a dissipative chaotic system, the sum of these exponents must be negative. In addition, the system is chaotic if at least one of these exponents is positive. A positive Lyapunov exponent indicates that the system is sensitive to the initial conditions. For the other two exponents, one should be approximately zero and the other should be negative. These positive and negative exponents determine the stretching and folding properties of the chaotic dynamic system. In this case the stationary points are neither attractors nor repellers, and so are called strange attractors [47].

The most commonly employed method to determine the Lyapunov spectrum is the Wolf Lyapunov exponent [48]. This is based on Gram-Schmidt reorthonormalization (GSR). Table 2.1 shows the Lyapunov exponents for the Rossler and Lorenz continuous attractors, as well as the exponents obtained using an implementation of the Wolf method in MATLAB. The Lyapunov exponents are arranged in order from largest to smallest. These results indicate that the exponents using these approaches can vary, but the differences are not substantial.

For the proposed discrete systems, the Wolf method was used with (2.5) and (2.7) to determine the Lyapunov exponents. The average of multiple runs (10 or 50) of the proposed discrete system with different numbers of discarded initial values and data set sizes for the Rossler and Lorenz based discrete systems are given in Tables 2.2 and 2.3, respectively. These results indicate that the first Lyapunov exponent is positive, the second is approximately zero, and the third is negative, as with the continuous systems.

In addition, these discrete systems are dissipative as the sum of the three exponents is negative. This means they are chaotic and bounded, so that folding and stretching occurs. Comparing Table 2.1 with Tables 2.2 and 2.3, it can be seen that the values for the discrete systems differ (smaller) from those of the continuous systems. This is due to the fact that the discrete systems are not approximations of the continuous systems, but rather new chaotic systems. As discussed previously,

Table 2.1: The Lyapunov Spectrum of Some Continuous Attractors

Method	Rossler system	Lorenz system
[44]	0.0900 0.0000 -9.8000	1.5070 0.0000 -22.4600
[45]	-	0.9057 0.00001 -14.5724
[46]	-	1.4504 -0.0057 -13.9990
[48]	0.1300 0.0000 -14.1000	2.1600 0.0000 -32.4000

Table 2.2: Discrete Rossler Lyapunov Spectrum (averaged over 10 or 50 Runs)

g	No. of initial values discarded	No. of values	Lyapunov spectrum
0.01	20	10000	0.0006 0.0001 -0.0334
0.01	20	50000	0.0007 0.0000 -0.0333
0.01	200	10000	0.0006 0.0001 -0.0335
0.01	200	50000	0.0006 0.0000 -0.0332
0.01	2000	10000	0.0006 0.0001 -0.0334
0.01	2000	50000	0.0007 0.0000 -0.0333

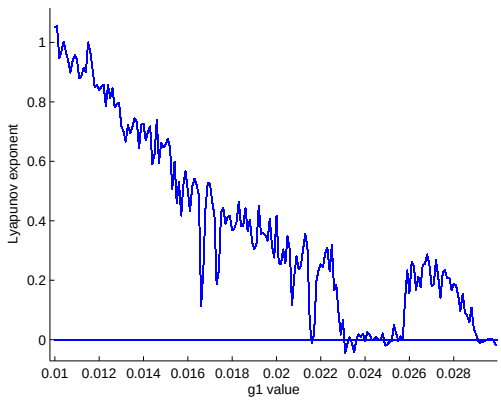
they are transformations of the continuous systems to discrete systems. However, the results in these tables verify the chaotic behavior of the discrete systems.

The second approach to determine if the systems have chaotic behavior is to calculate the largest Lyapunov exponent. This method was applied to the proposed discrete Rossler and Lorenz attractors for different initial values and data set sizes. The corresponding results are shown in Tables 2.4 and 2.5. From these tables, the largest Lyapunov exponents for the proposed discrete systems are near the largest Lyapunov exponents for the continuous systems. This indicates that the proposed discrete Rossler and Lorenz attractors will behave as chaotic systems, as the corresponding continuous systems are chaotic.

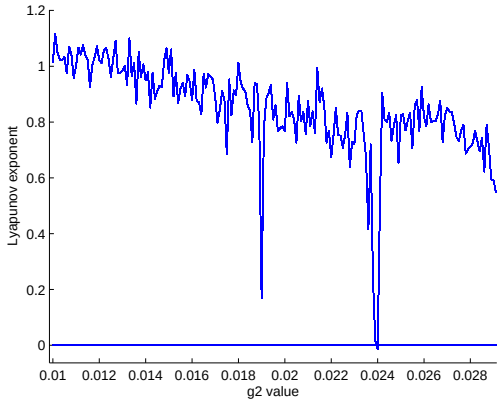
The largest Lyapunov exponent method was used to define the range of the model gain parameters g which result in chaotic behavior, and this is shown in Figures 2.2 and 2.3 for the Lorenz and Rossler attractors, respectively. To check the reliability of the proposed discrete systems, they were run for a very long time span (approximately one month).This was done using MATLAB for 2.592 X 10^8 iterations. The state space vectors for the last 10,000 output values are shown in Figures 2.4 and 2.5 for the Lorenz and Rossler attractors, respectively. These shapes are the same as those for the corresponding continuous systems [1] and [5], which further confirms that the chaotic behavior of the discrete systems is stable.

Table 2.3: Discrete Lorenz Lyapunov Spectrum (averaged over 10 or 50 Runs)

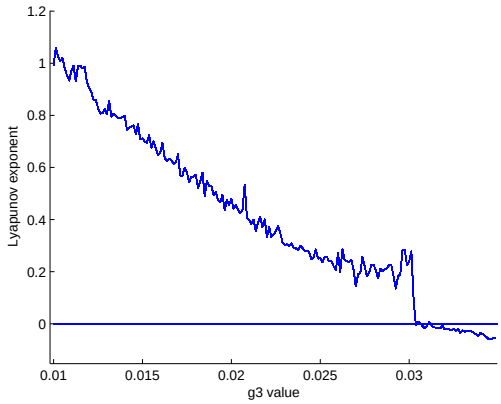
g	No. of initial values discarded	No. of values	Lyapunov spectrum
0.01	20	10000	0.0100 0.0001 -0.1497
0.01	20	50000	0.0103 0.0000 -0.1498
0.01	200	10000	0.0102 -0.0000 -0.1496
0.01	200	50000	0.0104 -0.0000 -0.1499
0.01	2000	10000	0.0104 -0.0001 -0.1499
0.01	2000	50000	0.0104 -0.0000 -0.1500



(a) Lyapunov exponent versus g_1 for the u variable



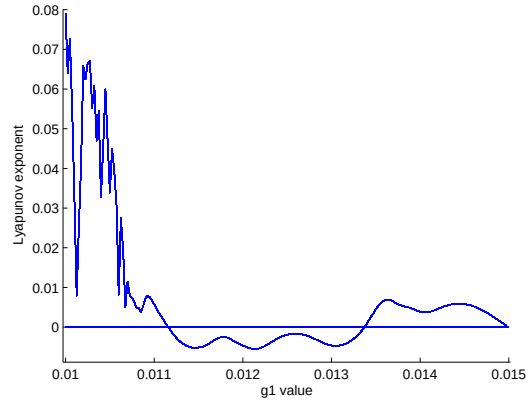
(b) Lyapunov exponent versus g_2 for the v variable



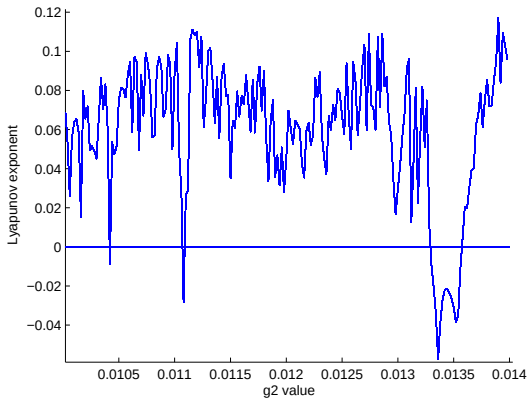
(c) Lyapunov exponent versus g_3 for the w variable

Figure 2.2: Discrete Lorenz parameter g ranges based on the Lyapunov exponents.

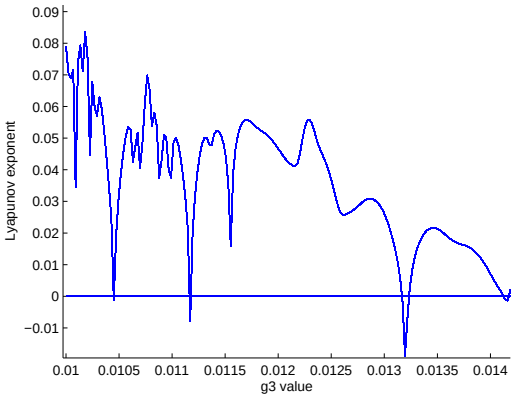
Randomness is a distinguishing feature of chaotic systems. Measuring the randomness is important if a chaotic system is to be used in cryptographic and spread spectrum communications applications.



(a) Lyapunov exponent versus g_1 for the x variable



(b) Lyapunov exponent versus g_1 for the y variable



(c) Lyapunov exponent versus g_1 for the z variable

Figure 2.3: Discrete Rossler parameter g ranges based on the Lyapunov exponents.

One method commonly used to measure randomness is the autocorrelation function. An ideal random sequence should be uncorrelated regardless of the shift. To evaluate the randomness of the proposed discrete systems, their autocorrelations were compared with those of the continuous systems. The results for the continuous Lorenz attractor with a time step of 0.01 s and a run time of 100 s and the proposed discrete Lorenz attractor with $10,000$ iterations are depicted in Figure 2.6. This shows that the autocorrelations have similar values, but the discrete system results in more zero crossings.

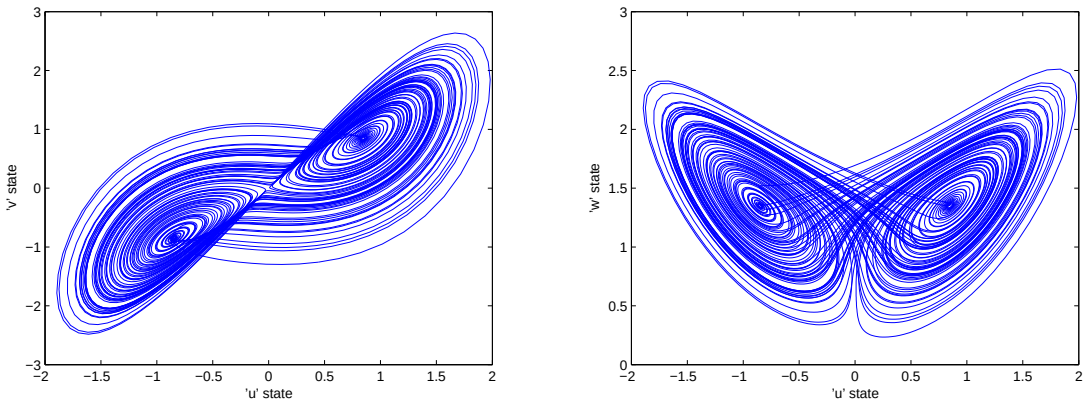
Next, the limits on the gains g for the 3D discrete chaotic system are determined. These parameters correspond to the three state equations (using 2.7). It is important to define upper and lower limits on these parameters that will ensure chaotic behavior.

Table 2.4: Largest Lyapunov Exponent for the Discrete Rossler Generator

g	No. of values	No. of initial values discarded	Largest Lyapunov
0.01	10000	500	0.0618
0.01	10000	1000	0.0618

Table 2.5: Largest Lyapunov Exponent for the Discrete Lorenz Generator

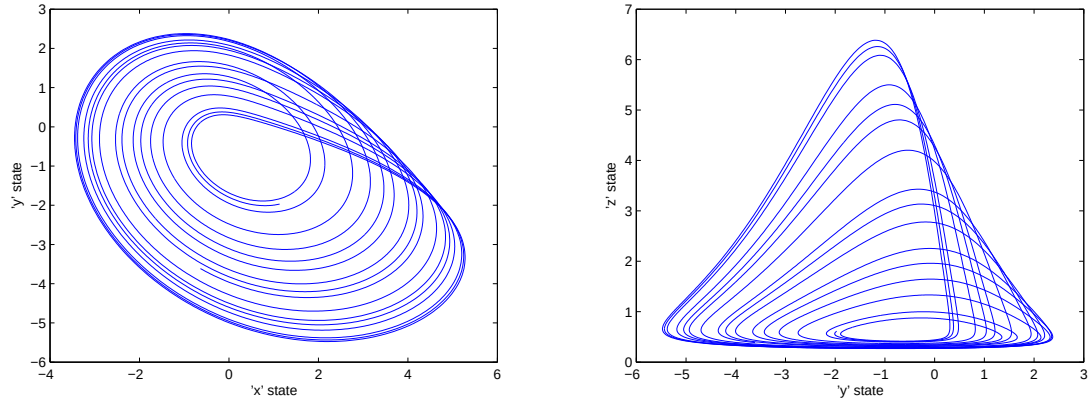
g	No. of values	No. of initial values discarded	Largest Lyapunov
0.01	10000	500	1.0414
0.01	10000	1000	1.0427



(a) (u,v) space (b) (v,w) space

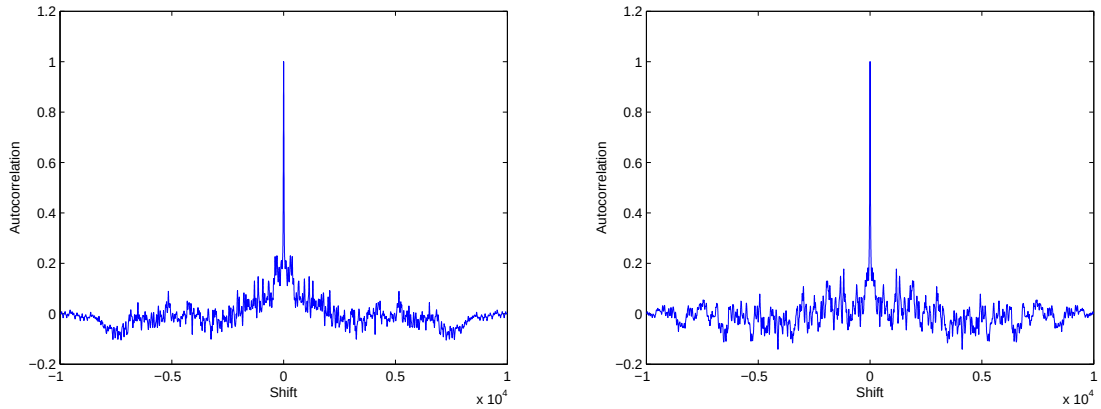
Figure 2.4: The discrete Lorenz attractor state space vectors.

Considering the Lyapunov exponents, the upper limit is $g = 0.024$, as a value of 0.025 results in all exponents being positive. Alternatively, randomness is considered in determining the lower limit. With a very small value of g, the signal will change very slowly, which produces highly correlated output values. Figure 2.7 shows the autocorrelation for the proposed discrete Lorenz attractor with $g = 0.024$, 0.01 and 0.001. The data set size used was $10,000$ values. The autocorrelation for $g = 0.001$ shows a very high correlation between values, while $g = 0.024$ results a very small correlation values. Thus the correlation increases as g decreases, and a value of $g = 0.01$ was found to provide sufficiently small correlation values. Therefore, an acceptable range for g is 0.01 to 0.024.



(a) (x,y) space (b) (y,z) space

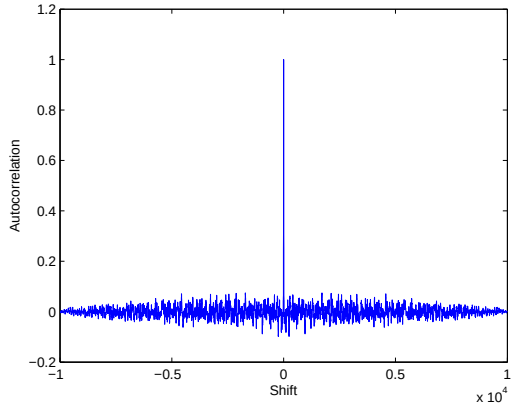
Figure 2.5: The discrete Rossler attractor state space vectors.



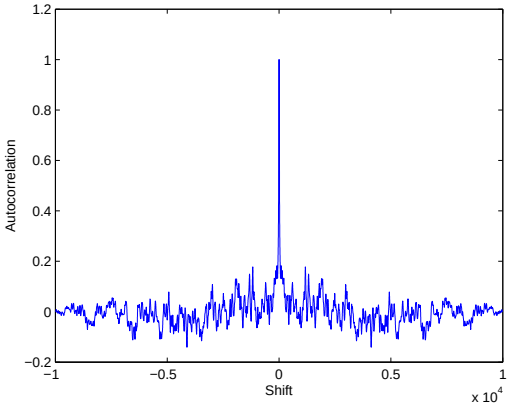
(a) Autocorrelation of the continuous Lorenz attractor output (b) Autocorrelation of the discrete Lorenz attractor output

Figure 2.6: Autocorrelation of the (a) continuous Lorenz attractor, and (b) discrete Lorenz attractor.

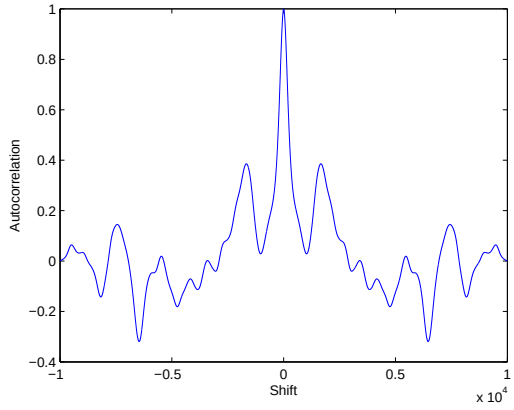
The synchronization of chaotic systems was first achieved by Pecora and Carroll [8]. This was done using one state variable as a drive signal to synchronize the remaining state variables (subsystem). The only requirement is that the subsystem be stable, so the corresponding Lyapunov exponents must all be negative. Table 2.6 shows the conditional Lyapunov exponents of the discrete Lorenz and Rossler driven subsystems using different driving signals. The conditional Lyapunov exponents are the Lyapunov exponents of a driven subsystem. From this table, the state variables u and v can be used to synchronize two Lorenz attractors, while the state variable y



(a) $g = 0.024$



(b) $g = 0.01$



(c) $g = 0.001$

Figure 2.7: Autocorrelation for different values of g.

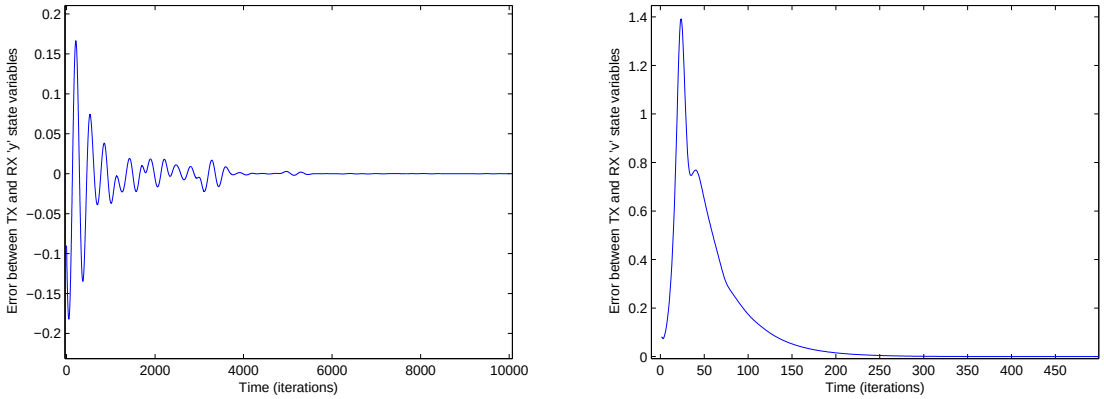
is the only driving signal that can be used for Rossler attractors. These results are similar to the corresponding results for continuous Lorenz and Rossler attractors [8].

To confirm the synchronization of the proposed discrete systems, the Simulink models (for 2.5 and 2.7) were used to synchronize two identical chaotic systems but with different initial conditions. The results for the discrete Rossler attractor are given in Figure 2.8-a using state y as the drive signal with initial state variable values [0.1 0.01 0.2] at the transmitter and [0.4 0.1 0.3] at the receiver. As the Rossler attractor is very sensitive to the initial conditions, the synchronization was achieved using feedback control [49]. The results for the discrete Lorenz attractor are given in Figure 2.8-b using state v as the drive signal with initial state variable values [0.1 0.01 0.2] at the transmitter and [0.4 0.1 0.3] at the receiver. The Lorenz attractor

Table 2.6: Conditional Lyapunov Exponents for Different Drive-Response Subsystems for the Discrete Lorenz and Rossler Systems

(drive-response) subsystem	Conditional Lyapunov exponent
Lorenz	
u driving signal; (v,w) response	-0.0147 -0.0153
v driving signal; (u,w) response	-0.0270 -0.1054
w driving signal; (u,v) response	0.0000 -0.1187
Rossler	
x driving signal; (y,z) response	0.0040 -0.0367
y driving signal; (x,z) response	-0.0021 -0.0345
z driving signal; (x,y) response	0.0020 0.0020

synchronization was achieved using the drive-response system in [8]. These results show that synchronization can be achieved when the initial conditions at the receiver and transmitter differ.



(a) Rossler synchronization

(b) Lorenz synchronization

Figure 2.8: Synchronization with different initial conditions at the transmitter and receiver using the (a) Rossler discrete attractor state y as the drive signal, and (b) Lorenz discrete attractor using state v as the drive signal.

2.4 Conclusion

In this chapter, discrete chaotic systems were developed based on continuous 3D systems. These systems have several advantages over continuous systems. Continuous systems implemented using analog circuits suffer from accuracy problems associated

with analog components. Approximations based on techniques such as Runge-Kutta methods lead to complex digital implementations. Conversely, the proposed discrete models are much simpler and have additional parameters that can be used to control system behavior. This is an advantage when these discrete systems are used in important applications such as real-time chaotic communications and cryptography.

Since these new discrete systems are transformations of continuous systems and not approximations, the chaotic behavior of these systems was examined and confirmed for a range of system parameters. This has been done using both Lyapunov exponents and randomness. A model was developed using Simulink to verify the theoretical results. The proposed transformation can be applied to any continuous system, and is not limited to the two systems examined in this chapter.

Chapter 3

A New 3D Chaotic Cipher for Encrypting Two Data Streams Simultaneously

3.1 Introduction

In this chapter, a new chaotic cipher based on the discrete Lorenz generator is presented to encrypt two digital data streams. The non-autonomous modulation technique is employed to encrypt the data samples, which enhances the security of the cipher. Each data sample is injected into the dynamics of the Lorenz generator through one of the difference equations. Accordingly, the transmitted signal is a complex mixture of the encrypted data. Thus the data are encrypted and mixed together through the generator dynamics. The data streams affect the dynamics of the chaotic generators independently. This technique improves the chaotic non-periodic property. In addition, the cipher has a simple structure and so is suitable for practical applications. Moreover, using one discrete Lorenz generator for encrypting/decrypting two data streams saves the hardware resources by approximately 50%. The cryptographic properties of the cipher are analyzed. The results obtained show that it provides excellent security and is resistant to existing attacks such as those based on Lorenz synchronization.

3.2 Literature Review

As mentioned in the Introduction, chaotic cryptosystems have evolved from analog encryption of analog and digital information, to the digital chaotic cryptosystems used in modern communications.

Digital chaotic cryptosystems avoid the problems of analog systems by encrypting data in higher-layers using discrete chaotic generators, rather than in the physical layer using analog systems. Analog systems are designed considering the effects of receiver noise and channel fading. On the other hand, the finite precision of the digital hardware makes the chaotic orbits periodic. Generally, the length of an orbit is different according to the complexity of the chaotic attractor, and the precision of the digital hardware [50]. In addition, the approximation process of the continuous chaotic systems in digital hardware is complex in terms of the computations. Thus, most of digital chaotic cryptosystems are based on low-dimensional chaotic attractors. The use of discrete chaotic generators in cryptosystems can be categorized as either block or stream ciphers, or public key ciphers.

In block ciphers, a string of bits is permuted to another string with the same length via a discrete chaotic map. [51] Is the first work presents chaotic permutations (called Bernoulli permutation) using the Baker map, followed by several ciphers such as in [52] where multi-chaotic systems are used.

For chaos-based stream ciphers, a chaotic generator is used as a PRNG. In these techniques, a chaotic map is used to generate a larger set of random-like numbers from a small set (seed), which in this case consists of the initial values and the control parameters of the chaotic generator [53]. The observations in [34] show that the average orbit length grows exponentially with the precision of the digital hardware. Based on this fact, using high precision arithmetic keeps the non-periodicity for longer orbit lengths, and avoid key repetition if used with low precision arithmetic.

Chaotic generators in public key ciphers are used to establish secure communications without exchanging secret keys [54]. The first use of chaos for public key encryption was presented in [55], where the Chebyshev polynomial map is used. Most subsequent work is based on the Chebyshev map.

Hashing is another technique that benefits from chaos. It is a one-way function where a variable length input is transformed to a shorter fixed length output. Different approaches have been investigated for chaos-based hashing such as simple chaotic-based hash function [56], and chaotic neural network-based hash function [57].

Digital chaotic cryptosystem have been used in many applications especially image encryption, which is a very active research area with many ciphers [58, 59].

3.3 The New Cipher

In chaotic cryptography, a complex signal is desirable to make the encryption cipher more robust to attacks, in particular statistical attacks. Therefore, high-dimensional dynamical systems are preferable to low-dimensional systems. The proposed cipher employs a 3D discrete Lorenz map. This chaotic attractor is based on a 3D continuous Lorenz attractor [1], which has been shown to have very complex dynamics. The state variables of the continuous Lorenz attractor are described by the following differential equations

$$\begin{aligned} \dot{u} &= A(v-u) \\ \dot{v} &= Bu - v - 20uw \\ \dot{w} &= 5uv - Cw \end{aligned} \tag{3.1}$$

Although a continuous Lorenz attractor can be implemented using numerical techniques such as Runge-Kutta methods, a discrete attractor is employed here. This is because a discrete attractor provides greater signal complexity and can be implemented simply in hardware.

As mentioned previously, with analog chaotic systems, and in particular non-autonomous modulation, the data signal is injected into the dynamics of the chaotic system. Then one or more state variables are sent to the receiver, which must perform an inverse operation to retrieve the data. This requires full knowledge of the parameter values (i.e., the key). These systems also require synchronization between the transmitter and receiver, which is difficult to achieve using analog circuits, and create constraints on the data and the way it are injected into the dynamics of the generator. The proposed discrete cipher benefits from the non-autonomous modulation used in analog systems, but synchronization is easily maintained. The discrete Lorenz attractor employed here is given by the following difference equations

$$\begin{aligned} U_{n+1} &= g_1(A(V_n - U_n)) + U_n \\ V_{n+1} &= g_2(BU_n - V_n - 20U_nW_n) + V_n \\ W_{n+1} &= g_3(5U_nV_n - CW_n) + W_n \end{aligned} \tag{3.2}$$

where $A = 10$, $B = 28$, $C = 8/3$, and $g_i = 0.01$, $where\, i = 1, 2, 3$. There are three

state variables U, V and W. Two data samples m_1 and m_2 are inserted into U and V, respectively, which gives

$$\begin{aligned} U_{n+1} &= g_1(A(V_n - U_n) + m_{1_n}) + U_n \\ V_{n+1} &= g_2(BU_n - V_n - 20U_nW_n + m_{2_n}) + V_n \\ W_{n+1} &= g_3(5U_nV_n - CW_n) + W_n \end{aligned} \tag{3.3}$$

The transmitted signal is the U state variable, and the objective is to retrieve m_1 and m_2 from this signal at the receiver. Feedback is used to update the state variables at the receiver to synchronize the system and allow decryption of subsequent data values. To illustrate this, the cipher is analyzed for the first two iterations. Two iterations are required because the transmitted encrypted signal is a single state variable, but it conveys two data values. This gives one equation with two variables which has an infinite number of solutions. Therefore, the generator at the transmitter is used to encrypt m_1 and m_2 twice, which gives two equations with two unknowns, which has a unique solution if the other parameters are known.

Iteration 1:

$$U_1 = g_1(A(V_0 - U_0) + m_{1_0}) + U_0$$

$$V_1 = g_2(BU_0 - V_0 - 20U_0W_0 + m_{2_0}) + V_0$$

$$W_1 = g_3(5U_0V_0 - CW_0) + W_0$$

where U_0, V_0 and W_0 are the initial values which are known at both the transmitter and receiver as part of the secret key. In this case, U_1 conveys m_{1_0} only. At the receiver, to calculate m_{1_0} all variables in the first state equation should be known, including U_1. Since U_1 is the first received signal, m_{1_0} can be calculated as

$$\widetilde{m}_{1_0} = round[\frac{1}{g_1}(U_1 - U_0) - A(V_0 - U_0)] \tag{3.4}$$

where round denotes rounding to the nearest integer. However, to update the receiver state variables, both $\widetilde{m}_{1_0}$ and $\widetilde{m}_{2_0}$ must be known. Consider the set of equations

$$U_1 = g_1(A(V_0 - U_0) + \widetilde{m}_{1_0}) + U_0$$

$$\widetilde{V}_1 = g_2(BU_0 - V_0 - 20U_0W_0 + \widetilde{m}_{2_0}) + V_0$$

$$\widetilde{W}_1 = g_3(5U_0V_0 - CW_0) + W_0$$

To calculate $\widetilde{m}_{2_0}$ all other variables in the second state equation should be known at the receiver. Since $\widetilde{V}_1$ is unknown, a solution cannot be obtained. However, this problem can be solved by estimating $\widetilde{V}_1$ using the received value of U and their relationship with V in the first state equation in (3.3).

Iteration 2:

The updated state equations at the receiver are

$$U_2 = g_1(A(\widetilde{V}_1 - U_1) + \widetilde{m}_{1_1}) + U_1$$

$$\widetilde{V}_2 = g_2(BU_1 - \widetilde{V}_1 - 20U_1\widetilde{W}_1 + \widetilde{m}_{2_1}) + \widetilde{V}_1$$

$$\widetilde{W}_2 = g_3(5U_1\widetilde{V}_1 - C\widetilde{W}_1) + \widetilde{W}_1$$

In the first state equation, $\widetilde{V}_1$ is the only unknown variable once U_2 is received. This requires that the receiver store the previous value of U_1 to obtain

$$\widetilde{V}_1 = \frac{1}{A}(\frac{(U_2 - U_1)}{g_1} - \widetilde{m}_{1_1}) + U_1 \tag{3.5}$$

Note that the value of $\widetilde{V}_1$ depends on $\widetilde{m}_{1_1}$ which gives an infinite number of solutions. However, $\widetilde{m}_{1_1}$=$\widetilde{m}_{1_0}$ and $\widetilde{m}_{2_1}$=$\widetilde{m}_{2_0}$, so the solution is obtained from the second state equation as

$$\widetilde{m}_{2_0} = round[\frac{1}{g_2}(\widetilde{V}_1 - V_0) - (BU_0 - V_0 - 20U_0W_0)] \tag{3.6}$$

Therefore, the receiver must wait until U_1 and U_2 have been received before $\widetilde{m}_2$ can be calculated.

After $\widetilde{m}_1$ and $\widetilde{m}_2$ are obtained, the receiver state equations can be updated (thus achieving synchronization with the transmitter), and the next two data values can be recovered. For even n, the data are recovered at the receiver using the equations

$$\widetilde{m}_{1_n} = round[\frac{1}{g_1}(U_{n+1} - U_n) - A(\widetilde{V}_n - U_n)]$$

$$\widetilde{V}_{n+1} = \frac{1}{A}(\frac{(U_{n+2} - U_{n+1})}{g_1} - \widetilde{m}_{1_{n+1}}) + U_{n+1}$$

where $\widetilde{m}_{1_{n+1}} = \widetilde{m}_{1_n}$, and

$$\widetilde{m}_{2_n} = round[\frac{1}{g_2}(\widetilde{V}_{n+1} - \widetilde{V}_n) - (BU_n - \widetilde{V}_n - 20U_n\widetilde{W}_n)]$$

The corresponding state equation updates at the receiver are

$$\begin{aligned} U_{n+1} &= g_1(A(\widetilde{V}_n - U_n) + \widetilde{m}_{1_n}) + U_n \\ \widetilde{V}_{n+1} &= g_2(BU_n - \widetilde{V}_n - 20U_n\widetilde{W}_n + \widetilde{m}_{2_n}) + \widetilde{V}_n \\ \widetilde{W}_{n+1} &= g_3(5U_n\widetilde{V}_n - C\widetilde{W}_n) + \widetilde{W}_n \end{aligned} \tag{3.7}$$

The proposed cipher is able to encrypt and decrypt two sets of data values m_1 and m_2 simultaneously. This has implications on the security, throughput, modulation, demodulation, and implementation of the system. The system security will be discussed in detail in Section 3.5. Note that the throughput is the same as using a cipher to encrypt and decrypt the two data streams individually as the proposed cipher encrypts each value twice.

The modulation and demodulation of m_1 and m_2 must be done carefully since the cipher is chaotic. According to the ranges of m_1 and m_2, they may need to be scaled to preserve the chaotic behavior. From a security perspective, m_1 and m_2 can be functions of the data to be encrypted rather than the actual data values. This can increase the robustness against some types of attacks.

Although the proposed cipher is used to encrypt two data values simultaneously, the hardware implementation is simple. Note that encrypting two data streams using only one chaotic generator reduces the resources required by approximately 50%. In addition, the proposed discrete Lorenz generator has a simple structure compared with hyper chaotic systems or solutions which combine a conventional cryptographic cipher with a chaotic generator. This reduces the computational complexity, processing time, and power consumption. The proposed implementation exploits the fact that the discrete Lorenz map has two state variables in the first difference equation. Further, one of these variables can be used to recover the other if the initial conditions are known.

As mentioned previously, the proposed cipher can be used to encrypt two data files simultaneously, or encrypt just a single file. In the latter case, the file can be divided into two parts with one sent as m_1 and the other sent as m_2. In addition,

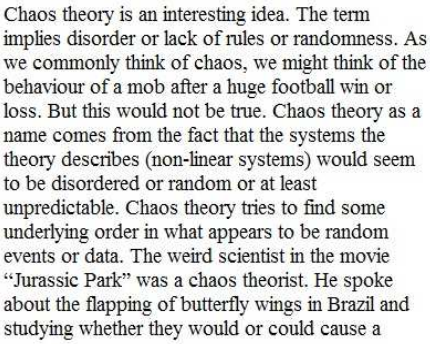
the signals corresponding to the data files (or two file halves), can be combined to increase the security. In this proposal, simple addition and subtraction are used for illustration purposes.

3.4 Cipher Implementation

One of the major disadvantages of using chaos in digital chaotic systems is the finite precision arithmetic with either software or hardware implementations. Since chaotic systems are sensitive to very small signal deviations due to the limited precision of the digital hardware, errors due to noise and quantization will propagate and multiply. Once the sensitivity threshold of the system is reached, synchronization between the transmitter and receiver will be lost, leading to system failure.

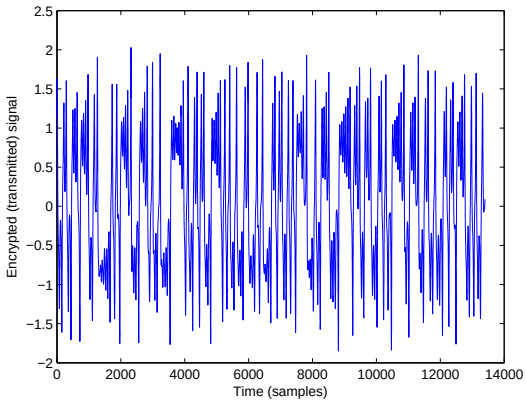
Although the proposed cipher employs real numbers using floating- or fixed-point arithmetic, the digital data have only integer values. At the transmitter, these values are converted to floating point numbers and scaled to preserve the chaotic behavior of the system. At the receiver, the estimated data values are rounded to integers and then used to synchronize the receiver. This approach removes any errors in recovering the data from the received signals and thus eliminates error propagation in the system. This results in at cipher which is robust to errors.

The proposed cipher was first investigated via simulation using MATLAB. All simulations were run for 200 iterations before starting the encryption/decryption process to eliminate the effects of the initial conditions. Two files were considered, a text file of 1120 words and a JPEG color image with dimensions 284×177 (both 7 kB). As the values are in the range of hundreds, m_1 and m_2 were multiplied by 0.0001 to preserve the chaotic behavior. Figure 3.1 shows a portion of the transmitted and recovered files, as well as the encrypted signal transmitted and its autocorrelation. Note that if a single file is being transmitted, the assignment of values to m_1 and m_2 can be done using another encryption cipher to increase the complexity (and thus the security), of the cipher. The proposed cipher was also used to encrypt a single 3.22 MB JPEG image with dimensions 3072×2304. As before, m_1 and m_2 were multiplied by 0.0001 to preserve the chaotic behavior. Note that most image encryption techniques process the image pixels as blocks of bits (16 for gray-scale images and 24 for color images). Conversely, the proposed cipher first converts the data values to floating-point. After decryption at the receiver, these values are converted back to integers to recover the original file. The results for this image file are shown in Figure 3.2.

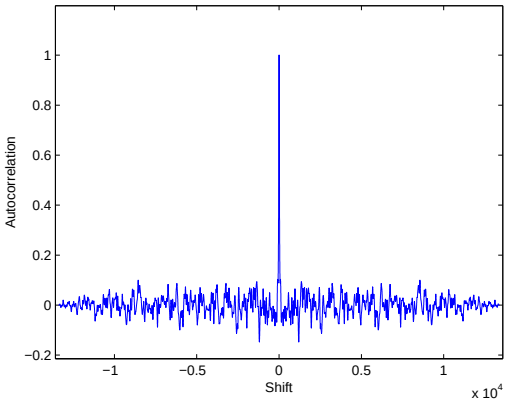

Chaos theory is an interesting idea. The term implies disorder or lack of rules or randomness. As we commonly think of chaos, we might think of the behaviour of a mob after a huge football win or loss. But this would not be true. Chaos theory as a name comes from the fact that the systems the theory describes (non-linear systems) would seem to be disordered or random or at least unpredictable. Chaos theory tries to find some underlying order in what appears to be random events or data. The weird scientist in the movie "Jurassic Park" was a chaos theorist. He spoke about the flapping of butterfly wings in Brazil and studying whether they would or could cause a

(a) Original text file

(b) Original bird image



(c) Transmitted signal



(d) Autocorrelation of the transmitted signal

Chaos theory is an interesting idea. The term implies disorder or lack of rules or randomness. As we commonly think of chaos, we might think of the behaviour of a mob after a huge football win or loss. But this would not be true. Chaos theory as a name comes from the fact that the systems the theory describes (non-linear systems) would seem to be disordered or random or at least unpredictable. Chaos theory tries to find some underlying order in what appears to be random events or data. The weird scientist in the movie "Jurassic Park" was a chaos theorist. He spoke about the flapping of butterfly wings in Brazil and studying whether they would or could cause a

(e) Recovered text file

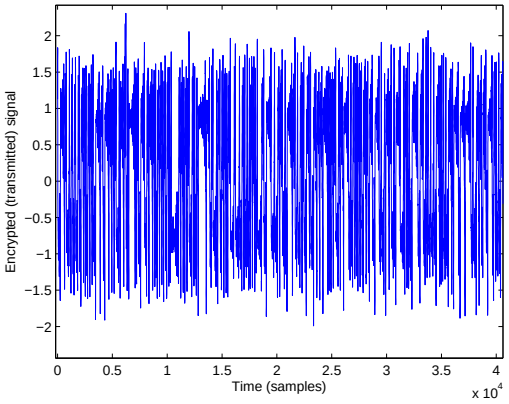
(f) Recovered bird image

Figure 3.1: (a) and (b) The original text file and image, (c) the transmitted signal, (d) the autocorrelation of the transmitted signal, and (e) and (f) the recovered text file and image.

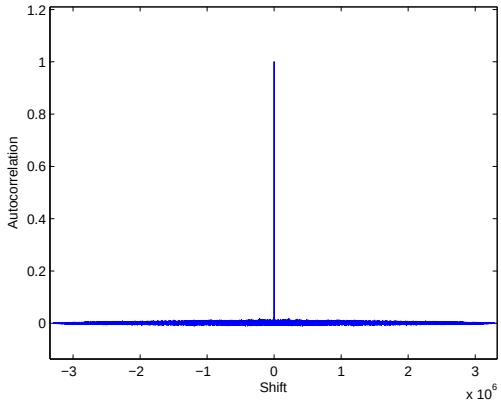
3.5 Security and Performance Analysis

Security is a major consideration with any cryptosystem. In this section, a security analysis of the proposed cipher is presented based on a brute-force attack, statistical

(a) Original image



(b) Transmitted signal



(c) Autocorrelation of the transmitted signal

(d) Recovered image

Figure 3.2: (a) The original image, (b) the transmitted signal, (c) the autocorrelation of the transmitted signal, and (d) the recovered image.

analysis, differential attack, key sensitivity attack, and known Lorenz attacks.

3.5.1 Statistical analysis

A good cipher should be robust to attacks based on a statistical analysis. Therefore, such an analysis is essential for any encryption cipher. Figures 3.1-d and 3.2-c show that the autocorrelation of the transmitted signals is low, and they are similar to the autocorrelation of the Lorenz attractor output. This illustrates the randomness of the generated signals, and the difficulty in exploiting them via correlation techniques.

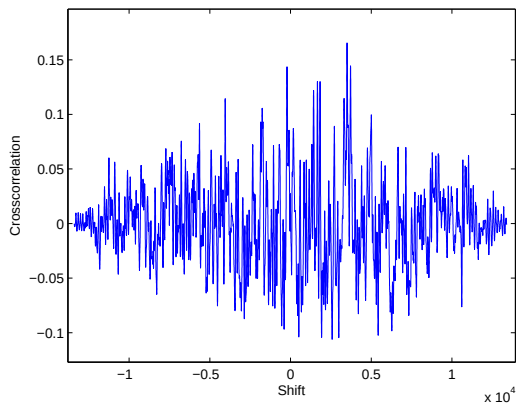
3.5.2 Differential analysis

Differential analysis is a powerful means of attacking cryptosystems and thus can be used to evaluate the security of an encryption cipher. In this attack, the cryptanalyst is assumed to have the capability of modifying single values of the plaintext (data) and observing the resulting encrypted signal. If such a change results in a significant change in this signal, then the attack is considered to be inefficient and impractical. To illustrate such an attack on the proposed cryptosystem, consider the text file in Section 3.4 with the first character changed from C to B, and the image file unchanged. The difference between the new encrypted signal and the original encrypted signal should be random. The cross-correlation of these signals and the autocorrelation of their difference are shown in Figure 3.3-a and 3.3-b, respectively. These results show that the correlation between the signals is minimal. This test was repeated several times with different symbols, and similar results were obtained. Next, the image file from the second test in Section 3.4 was employed with the first byte changed from 255 to 254 (a change in a single bit). The cross-correlation of the two signals and the autocorrelation of their difference are shown in Figure 3.3-c and 3.3-d, respectively. Again the correlation between the signals is minimal.

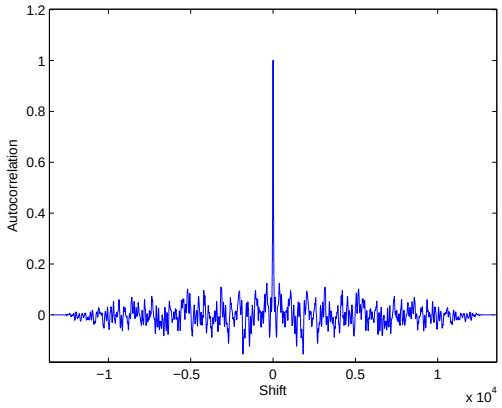
3.5.3 Attacks on the system

An attack based on having samples of both the plaintext and the corresponding ciphertext is called a known-plaintext attack. When an attacker has the capability to choose an arbitrary plaintext and obtain the corresponding ciphertext, it is called a chosen-plaintext attack. A chosen-ciphertext attack is thus when the attacker can insert ciphertext into the system and obtain the resulting plaintext. The goal of these attacks is to obtain information about the secret key.

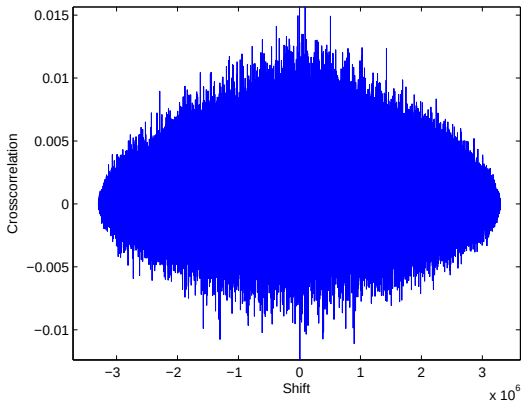
When the encryption cipher is based on non-autonomous modulation, the complex dynamics of the chaotic cipher are modified by the data being encrypted. Thus, the permutation and diffusion behavior of the chaotic system are directly affected by the data. In addition, this modulation is done over two of the three state equations of the discrete Lorenz generator with two plaintext streams encrypted. At the receiver, the decrypted plaintext values are used to update the chaotic generator. Thus, the generator dynamics are related to both the plaintext being encrypted, and the ciphertext being decrypted [32]. Therefore, without knowledge of the particular plaintext being encrypted, an attacker will not be able to reproduce the particular system dynamics,



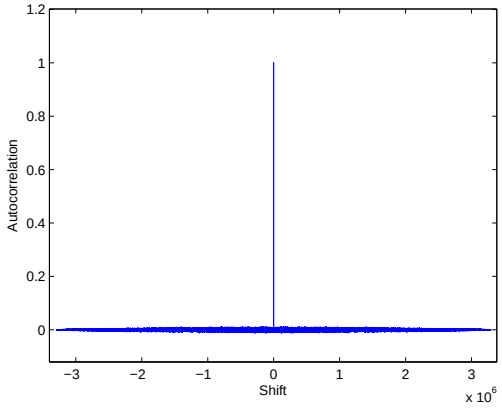
(a) Cross-correlation between two text files



(b) Autocorrelation of the difference



(c) Cross-correlation between two images



(d) Autocorrelation of the difference

Figure 3.3: (a) The cross-correlation of the encrypted signals generated from two text files with a small difference between them, (b) the autocorrelation of the difference between these encrypted signals, (c) the cross-correlation of the encrypted signals generated from two image files with a small difference between them, and (d) the autocorrelation of the difference between the encrypted signals.

so the proposed cipher is resistant to these attacks.

One situation that should be considered is when the attacker has access to the encryption system and can encrypt null (all-zero) data. In this case, the transmitted signal will reflect the dynamics of the chaotic generator without any variations due to the data. The attacker can then perform a geometric attack on the Lorenz generator. Despite this opportunity, these attacks will not be effective because the discrete Lorenz generator used in the proposed approach has an additional parameter g which appears in all three difference equations. Since three different g values

can be employed, they represent three additional key parameters beyond those of the differential equations of the continuous Lorenz generator and so provide an increased level of security. This greatly decreases the probability of a geometric attack on the proposed cipher being successful as will be shown later when attacks on the Lorenz generator are considered.

3.5.4 Key space analysis and brute-force attack

The key space of an encryption cipher is the set of keys that can be used for encryption. From a security perspective, the size of the key space should not be smaller than 2^{100} [60]. The key for the proposed cipher includes the initial values for the three state variables U_0, V_0 and W_0, the parameters A, B and C, and g for each of the state equations. From extensive experiments using this cipher for text and image files where the data values are integers, the key length is 39 decimal digits. Thus the key space is $10^{39} >> 2^{100}$, which provides significant robustness against a brute-force attack. Table 3.1 gives the sensitivity of each parameter. Any change in a parameter greater than its sensitivity will prevent an eavesdropper from decrypting data. In addition, the initial values should be known within the sensitivity of the variables, and even a small change will cause a loss of data. Note that the sensitivity is affected by the software and/or hardware used in the implementation, and thus an attack using a different implementation may fail solely because of this.

Table 3.1: Parameter Sensitivity, Key Length and Keyspace Size

Parameter	Sensitivity	Number of digits
Initial values		
U_0	10^{-6}	6
V_0	10^{-6}	6
W_0	10^{-6}	6
Lorenz parameters		
A	10^{-3}	3
B	10^{-3}	3
C	10^{-3}	3
g parameter for		
U	10^{-4}	4
V	10^{-4}	4
W	10^{-4}	4
Keyspace size		10^{39}

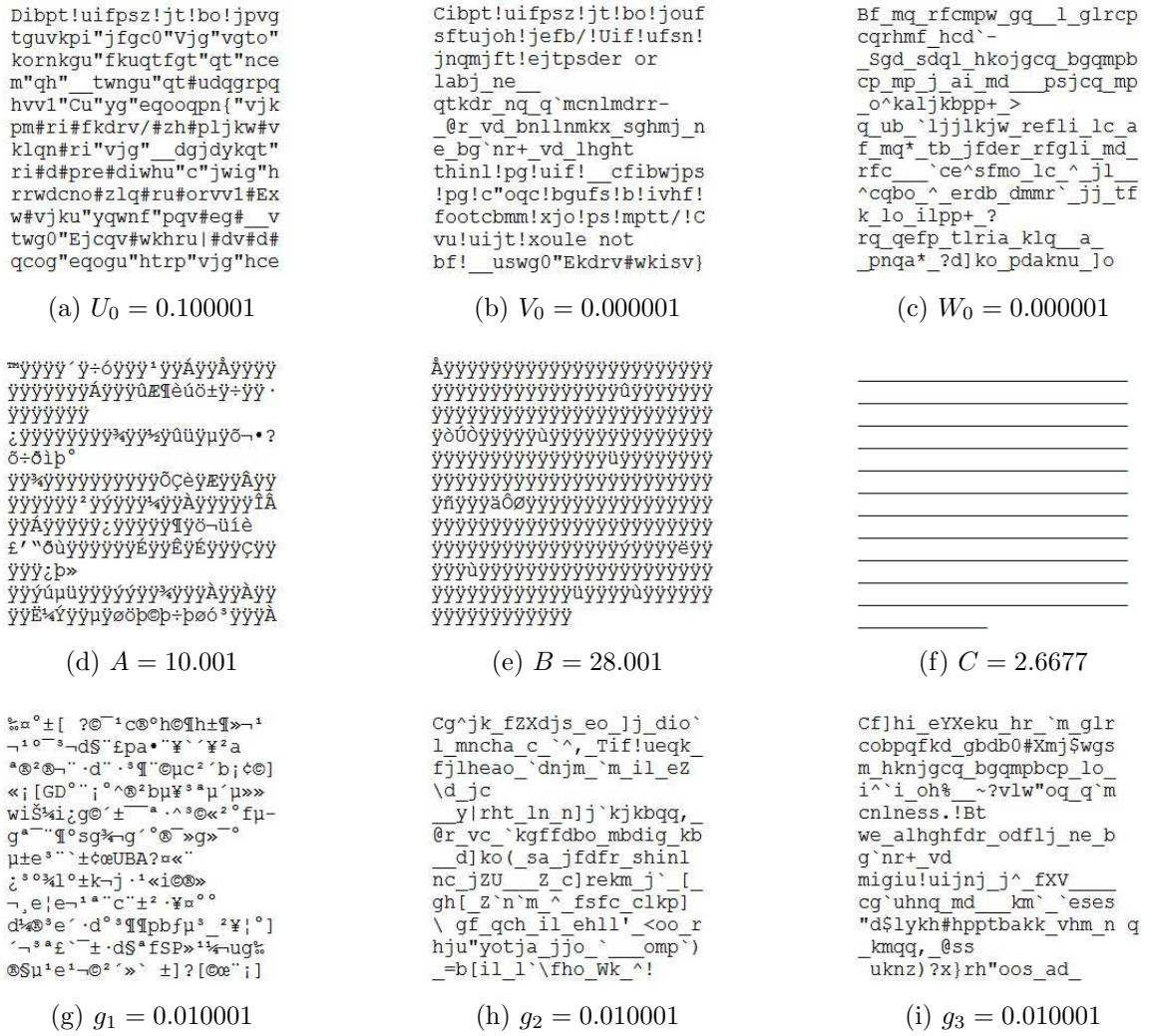
3.5.5 Key sensitivity

The cipher should be very sensitive to changes in the system parameters to resist cryptanalysis attacks. This property is illustrated in Figure 3.4 using the text file from Section 3.4. First the file was encrypted with initial conditions $(0.1, 0, 0)$, and parameters $A = 10$, $B = 28$, $C = 2.6667$, and $g_i = 0.01$, $i = 1, 2, 3$. Subfigures (a) to (i) show the decrypted text file with the parameters changed slightly to $U_0 = 0.100001$, $V_0 = 0.000001$, $W_0 = 0.000001$, $A = 10.001$, $B = 28.001$, $C = 2.6677$, $g_1 = 0.010001$, $g_2 = 0.010001$, and $g_3 = 0.010001$, respectively. In all cases, the original file is not obtained, which shows that the cipher has significant sensitivity to small changes in the key.

3.5.6 Lorenz attacks

Since the proposed cipher is based on the discrete Lorenz system, it is important to consider the attacks used against the continuous Lorenz system, which are primarily the return map and geometry attacks. In [61], the authors showed that for non-autonomous modulation, the return map shape is not changed but only shifted. Their attack was based on a simple cosine function as the message data, and they were only able to extract the periodicity of the function, not the parameters of the generator. However, the proposed cipher is a digital system, and the data values are not periodic, which make this attack useless. Figure 3.5 shows that the return map of the proposed cipher is blurry as defined in [61].

In [62] a cosine function was used to blur the return map to make it more difficult to extract the data from this map. The attack proposed in [63] predicts the periodicity of the cosine using the power spectrum of the encrypted signal, and then a geometry attack is employed to obtain the parameter values. Figure 3.5-a and 3.5-b present the return map for the encrypted image signal using the proposed technique. This shows that the map is very blurry. The red dots represent the return map without the data, while the blue dots represent the return map with the data injected into the system. In the proposed cipher, the complexity (confusion) of the return map depends on the data rather than a periodic function. Since the data values are not periodic, the attack in [63] will fail, so attacks based on the return map will not break the proposed cipher. Power spectrum-based approaches will also not work if the modulation is not periodic. In addition, any spectrum estimation will be a combination of the values representing the two data, and no further information can



(a) $U_0 = 0.100001$ (b) $V_0 = 0.000001$ (c) $W_0 = 0.000001$

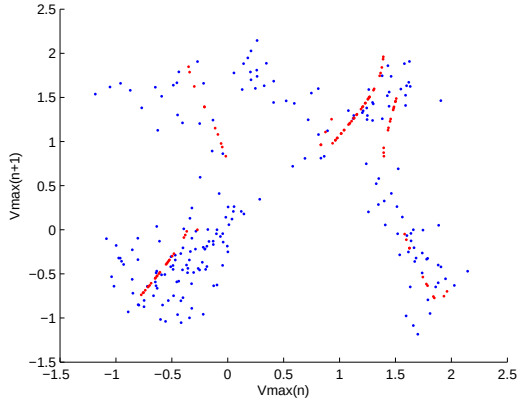
(d) $A = 10.001$ (e) $B = 28.001$ (f) $C = 2.6677$

(g) $g_1 = 0.010001$ (h) $g_2 = 0.010001$ (i) $g_3 = 0.010001$

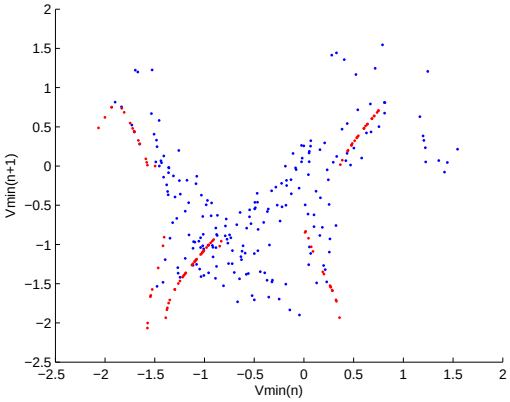
Figure 3.4: Decrypted text files using the chaotic cryptosystem with the parameters changed slightly to (a) $U_0 = 0.100001$, (b) $V_0 = 0.000001$, (c) $W_0 = 0.000001$, (d) $A = 10.001$, (e) $B = 28.001$, (f) $C = 2.6677$, (g) $g_1 = 0.010001$, (h) $g_2 = 0.010001$, and (i) $g_3 = 0.010001$.

be acquired without knowing the exact parameter values to solve the equations. As an example, Figure 3.5-c presents the power spectrum for the encrypted bird image from the previous section. This shows that the spectrum has no discernible features.

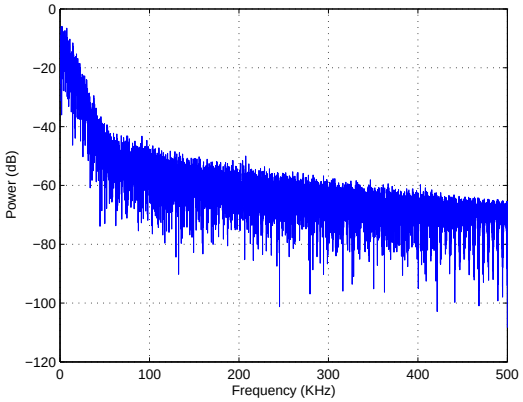
Another approach to breaking Lorenz attractor-based cryptosystems is via a geometry attack. This reduces the key space and the time required to predict the remaining parameters. In [64], the drive response synchronization method was used to construct the geometry of the Lorenz attractor. This reduced the search space and allowed the parameters to be determined directly from the ciphertext. In [65],



(a) V_{max} return map



(b) V_{min} return map



(c) Power spectrum

(d) Bird image

Figure 3.5: (a) and (b) The return maps, (c) the power spectrum, and (d) the bird image.

a simple method was presented to simultaneously obtain values for the three Lorenz parameters. This is based on the synchronization of the subsystems (v, w) or (u, w) with u or v used as the driving signal, respectively. In [66], an adaptive method to synchronize with a partially known Lorenz chaotic system was presented. This approach exploits a particular characterization of the Lorenz system to identify system parameters. Note that all of the methods in [64, 65, 66] are based on the system synchronization. The attacker uses the transmitted signal as the driving signal to synchronize his receiver to obtain the Lorenz attractor parameters. An advantage of the proposed cipher is that synchronization cannot be achieved without knowing all the parameter values, including the initial values, exactly. The reason is that it injects

two signals into the first two difference equations of the non-autonomous transmitter, and since the transmitted signal is only the state variable U, synchronization at the receiver cannot be achieved without estimating the second data signal and injecting it into the second difference equation. In addition, the geometry analysis is more complex than for the continuous Lorenz system due to the additional parameters in the difference equations. Thus it can be concluded that the proposed discrete cipher is secure against techniques used to attack the continuous Lorenz system.

3.6 Execution Time

Apart from security, an important issue for encryption ciphers is the time taken for execution. A comparison of several image encryption ciphers that employ 1D, 2D or 3D chaotic system and the proposed cipher is given in Table 3.2. The proposed cipher was used to simultaneously encrypt two images of size 1,400,676 bytes, and two text files of size 670,934 bytes using a Pentium dual-core 1.6 GHz processor. The tests were run 1000 times and the average times are shown in the table. Considering the differences in the hardware used for each cipher, the performance of the proposed cipher is comparable. The 1D ciphers are simpler and thus faster than many higher dimensional ciphers, but they provide much lower security. The ciphers in [67] and [68] have speeds similar to that of the proposed cipher, while the cipher in [69] is slower by a factor of almost 6. Note that the system characteristics for these four cases are almost identical. In addition, despite the fact that the processor used to obtain the results in [70] is faster than the processor used in this proposal, the proposed cipher is faster by more than a factor of 2. For 2D ciphers, although the processor in [71] is better, the proposed cipher is faster by a factor of 2 or more. In [72] the coupled two-dimensional piecewise non-linear chaotic map (CTPNCM) cipher was presented and compared with a cipher based on the advanced encryption standard (AES) using an Intel Core 2 Duo 3.0 GHz processor, which is faster than the processor used in this proposal. However, the proposed 3D cipher is faster than the AES-based ciphers and is slower than the CTPNC cipher by less than 20%, but it provides better security. On the other hand, the proposed cipher is superior to the 3D ciphers as it is much faster than the cipher in [73], and more than a factor of two faster than the cipher in [74]. This is despite the fact that the processor employed here is less powerful. The results in this table indicate that the speed of the proposed cipher is superior to many chaos-based ciphers (particularly the 3D ciphers), and is suitable for most real-time

applications.

Table 3.2: Encryption/Decryption Speeds for Various Ciphers

Cipher	System characteristics	Dimension	Speed
Ref. [67]	Pentium IV 1.6 GHz	1D	59.2 Mbps
Ref. [68]	Pentium IV 2.1 GHz	1D	74.4 Mbps
Ref. [69]	Pentium 4 1.7 GHz	1D	512×512 (768 kB) 1.14 s (5.2 Mbps) 256×256 (65 kB) 0.078 s (6.5 Mbps)
Ref. [70]	Pentium dual-core 2.7 GHz	1D	15.6 Mbps
Ref. [71]	Intel core 2 duo 2.1 GHz	2D	From 6.69 to 22.55 Mbps
Ref. [72]	Intel core 2 duo 3.0 GHz	2D	AES [128 key] 11.2 Mbps AES [192 key] 9.25 Mbps AES [256 key] 9.19 Mbps CTPNCM cipher 44.9 Mbps
Ref. [73]	Pentium IV 1 GHz	3D	8 Mbps
Ref. [74]	Intel core i5 2.27 GHz	3D	15.44 Mbps
Proposed cipher	Pentium dual core 1.6 GHz	3D	Two images (each 1,400,676 bytes) Encryption 0.5662 s (37.7 Mbps) Decryption 0.5795 s (36.8 Mbps) Two text files (each 670,934 bytes) Encryption 0.2722 s (37.6 Mbps) Decryption 0.2784 s (36.7 Mbps)

3.7 Effect of Noise

The secure transfer of data requires two operations, encryption and decryption, with the latter reversing the encryption operation in order to recover the data. In the proposed cipher, encryption is achieved by modifying the Lorenz generator states. This is done by inserting weighted data samples into the dynamics of the chaotic generator. At the receiver, the decryption process consists of two phases. First, the data samples are extracted, and then the Lorenz generator states are updated using these samples. Although this updating is identical to the encryption process at the transmitter, the extraction phase requires additional arithmetic operations. These operations create quantization noise due to the finite precision arithmetic of digital hardware and/or software. This noise can result in the failure of the chaotic cipher, but can be overcome by exploiting the fact that the data to be encrypted have integer values. During the extraction phase, the floating point valued data samples

are rounded to the nearest integer, which effectively removes the noise and prevents it from propagating within the chaotic dynamics, which would cause decryption failure.

To test the immunity of the proposed cipher against quantization noise, small errors were added to each arithmetic operation in (3.4), (3.5) and (3.6) during the extraction phase. This is in addition to the existing quantization noise from the hardware and software used. These additional errors simulate the effect of using different computing platforms at the transmitter and receiver and show the sensitivity to noise. The data streams in Figures 3.1 and 3.2 were used with double-precision (64-bit) floating-point arithmetic to generate the transmitted chaotic signal. The length of the mantissa is 52 bits. The results obtained indicate that decryption is successful if the errors introduced in the arithmetic operations in (3.4) and (3.6) are less than 10^{-5} (the least significant 36 bits of the mantissa are changed), and for (3.5) are less than 10^{-6} (the least significant 32 bits of the mantissa are changed). This shows that the rounding employed in the cipher provides significant robustness to noise as decryption of the data can be achieved in the presence of quantization noise which is 62% to 69% of the mantissa.

3.8 Conclusion

In this chapter, a new chaos-based cryptosystem has been proposed. It is based on the non-autonomous modulation technique used in continuous chaotic cryptosystems. The proposed cipher provides significant security by encrypting two data streams simultaneously, which effectively prevents an attacker from achieving synchronization to obtain the system parameters. The values assigned to these streams can be done using another cryptographic cipher to increase the key length and thus the security. An extensive evaluation of the proposed cipher was performed which showed that the proposed cipher is more secure and faster than other chaotic cryptosystems proposed in the literature. It is therefore very suitable for most real-time encryption applications.

Chapter 4

Real-time Image Encryption using a Three-Dimensional Discrete Dual Chaotic Cipher

4.1 Introduction

In this chapter, a cipher is proposed for real-time encryption applications, in particular image encryption. This scheme employs a dual chaotic generator based on the 3D discrete Lorenz attractor. Extending the work in the previous chapter, the encryption is achieved using non-autonomous modulation where the image data are injected into the dynamics of a master chaotic generator. The second generator is used to permute the dynamics of the master generator using the same approach. In the literature, dual chaotic cryptographic systems were developed with one generator used to drive the other to retrieve the image. Conversely, the proposed approach uses one (permutation) generator to permute the other (master) generator to increase the security, the orbit length, and to preserve the randomness. Decryption cannot be achieved without the signal from the permutation generator as it is required for synchronization with the master generator. This prevents an eavesdropper from decrypting the ciphered signal by synchronizing their master generator using only this signal. Since the image data can be regarded as a random source, it results in a random permutation of the dynamics of the master generator. In addition, a technique is proposed to mitigate the error propagation due to the finite precision arithmetic of digital hardware. In particular, truncation and rounding errors are eliminated by em-

ploying an integer representation of the image data which can be easily implemented. The simple hardware architecture of the cipher makes it suitable for secure real-time applications.

This cipher is similar to the cipher in Chapter 3. The difference is that the cipher in Chapter 3 encrypts two data streams at the same time, by involving the data in the dynamics of the Lorenz generator, while in this cipher the second data stream is replaced by a permutation signal generated from another chaotic generator. In this chapter a second Lorenz generator is used. Moreover, in Chapter 3 the cipher encrypts the samples of the two data streams twice and only one state variable is transmitted. Conversely, the cipher in this chapter encrypts each sample only once and only one image is encrypted.

4.2 The Proposed Lorenz Dual Chaotic Cipher

The proposed system consists of two 3D chaotic generators based on the discrete Lorenz attractor. This provides greater signal complexity than other generators and can be implemented simply in hardware. A master generator is used to create the encrypted signal, while a permutation generator is used to permute the master generator dynamics. This permutation enhances the security of the system. According to (2.7) the state equations of the discrete Lorenz attractor are

$$\begin{aligned} U_{n+1} &= g_1(V_n - U_n)A + U_n \\ V_{n+1} &= g_2(BU_n - V_n - 20U_nW_n) + V_n \\ W_{n+1} &= g_3(5U_nV_n - CW_n) + W_n \end{aligned} \tag{4.1}$$

where A,B,C and g_i, $i = 1, 2, 3$ are control parameters. Different control parameters are used for the master and permutation generators. The image data are encrypted using non-autonomous modulation via insertion into a state equation of the master generator, while the permutation signal is injected into another state equation of this generator. The state equations of the master generator for encryption are

$$\begin{aligned} U_{n+1} &= g_1((V_n - U_n)A + m_n) + U_n \\ V_{n+1} &= g_2(BU_n - V_n - 20U_nW_n + p_n) + V_n \\ W_{n+1} &= g_3(5U_nV_n - CW_n) + W_n \end{aligned} \tag{4.2}$$

where m_n is a scaled image data value and p_n is the permutation signal which is one of

the state variables of the permutation generator. Since the image to be encrypted is digital, it consists of blocks of bits (bytes or words). These blocks can be represented as integers and then converted to floating-point numbers. Scaling of the resulting values is used to preserve the chaotic behavior of the system [20]. As mentioned previously, these values are injected into the dynamics of the master generator to obtain the encrypted signal. For decryption, the encrypted signal is used with the signal generated by the local permutation generator to synchronize the master generator and retrieve the image. As the U state is the encrypted state variable, this signal is used to update the difference equations for decryption as follows

$$\begin{aligned} \widetilde{U}_{n+1} &= U_{n+1} \\ \widetilde{V}_{n+1} &= g_2(BU_n - \widetilde{V}_n - 20U_n\widetilde{W}_n + p_n) + \widetilde{V}_n \\ \widetilde{W}_{n+1} &= g_3(5U_n\widetilde{V}_n - C\widetilde{W}_n) + \widetilde{W}_n \end{aligned} \tag{4.3}$$

where U_{n+1} is the encrypted signal and $\widetilde{U}$, $\widetilde{V}$, and $\widetilde{W}$ are the state variables for decryption. The retrieved image data are given by

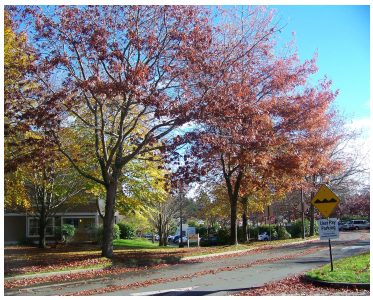
$$\widetilde{m}_n = \text{round}\left[\frac{1}{g_1}(U_{n+1} - U_n) - A(\widetilde{V}_n - U_n)\right] \tag{4.4}$$

where round denotes rounding to the nearest integer. This is used to eliminate noise due to the finite precision arithmetic in the software and/or digital hardware.

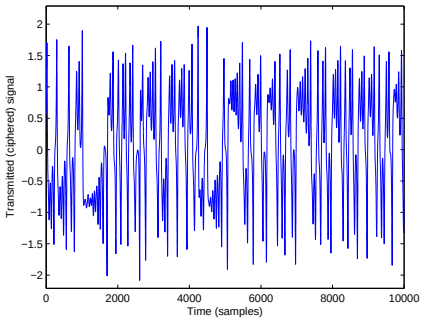
4.3 Cipher Implementation

The proposed system was simulated using MATLAB with double-precision floating-point numbers. The control parameters for the master generator are $A_M = 10$, $B_M = 28$, $C_M = 8/3$, and $g_{M_i} = 0.024$, $i = 1, 2, 3$, while the parameters for the permutation generator are $A_P = 9.8$, $B_P = 27$, $C_P = 2.8$, and $g_{P_i} = 0.024$, $i = 1, 2, 3$. The state variable V of the permutation generator is used as the permutation signal p_n. A JPEG color image of size $5,067,717$ bytes (3072×2304 pixels) was used to investigate the system properties. This size was chosen to test the ability of the proposed encryption cipher to mitigate the noise due to finite precision arithmetic. The integer image pixels m_n were scaled by a factor of 0.00001 to preserve the chaotic behavior, and the permutation values p_n were scaled by a factor of 0.01 to blur the return map as will be shown in Section 6.4. Figure 4.1 presents the original image,

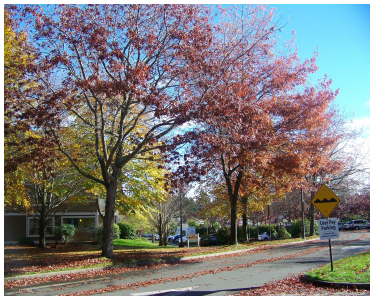
part of the encrypted signal, and the recovered image. This shows that the range of the encrypted signal differs significantly from the range of the image data (0 to 255).



(a) Original image



(b) Encrypted signal



(c) Recovered image

Figure 4.1: An example of image encryption: (a) the original image, (b) the encrypted signal, and (c) the recovered image.

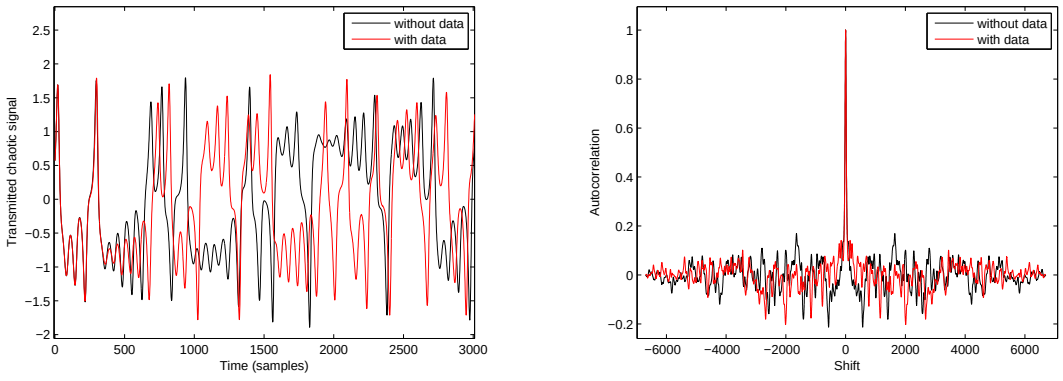
Compared to the cipher in [75], the proposed cipher is significantly faster and thus is more suitable for real-time image applications. With the approach in [75], two data streams are injected into two state variable equations of the generator, which requires encrypting each sample twice. Instead of inserting two data streams, the proposed cipher inserts the image data into one state equation and a permutation signal into another state equation. Therefore, the image data are encrypted only once, and consequently the proposed cipher is faster.

4.4 Security Analysis

Security is a major consideration with any cryptosystem. In this section, the security of the proposed cipher is analyzed based on several well-known attacks. Although the chaotic behavior of the continuous Lorenz attractor is very complex, there have

been numerous attempts to break ciphers based on this attractor, and these attacks will be considered for the proposed cipher.

Known-plaintext, chosen-plaintext, and chosen-ciphertext attacks aim to obtain information about the encryption scheme employed. In a chosen-plaintext attack, the attacker has many plaintext-ciphertext pairs, so it is considered more powerful than a known-plaintext attack which typically has few pairs. Therefore, any cipher that prevents a chosen-plaintext attack should also be secure against known-plaintext and ciphertext-only attacks. The proposed cipher is based on non-autonomous modulation, so the complex dynamics of the chaotic cipher are modified by the image being encrypted and a second generator. Thus, the permutation and diffusion in the chaotic system are directly affected by external signals. In addition, this modulation is done over two of the three state equations of the discrete Lorenz generator. Since the dynamics of this generator evolve over time, a given image data value will affect the resulting chaotic signal differently at different times. Figure 4.2 shows the U state variable of the master generator in the time-domain with and without the injected image data and the autocorrelation of each signal. This indicates that although the use of the image data results in a different chaotic trajectory, it does not alter the properties of the chaotic signal such as the autocorrelation. For decryption, the master generator dynamics are related to the image being encrypted and the permutation signal. Therefore, without knowledge of the permutation signal, an attacker will not be able to reproduce the system dynamics, so the proposed cipher is resistant to these attacks.



(a) Chaotic signal (b) Autocorrelation

Figure 4.2: The effect of the image (a) on the time-domain chaotic signal, and (b) the autocorrelation of the chaotic signal.

4.4.1 Key space analysis and brute-force attack

At present, a key length of 2^{100} is sufficient to protect a cipher against a brute-force attack [60]. Table 4.1 shows the sensitivity of the control parameters of the master and permutation discrete Lorenz generators using double-precision floating-point numbers. This sensitivity is the value below which the dynamics of the generators for encryption and decryption converge. Conversely, the dynamics differ for larger parameter differences. The product of the sensitivities (or equivalently the sum of their digits) gives the key length for the cipher. The resulting key length is 60 decimal digits and $10^{60} >> 2^{100}$. Note that this does not include the initial values of the master generator. This is because synchronization of this generator can be achieved even if the initial values are not known exactly. On the other hand, the initial values of the permutation generator are very important since the output is used to permute the master generator. This greatly increases the robustness against a brute-force attack, as indicated by the values on the right of Table 4.1.

Table 4.1: The Key Length based on the Master and Permutation Generator Sensitivities

Master generator parameter	Sensitivity	Number of digits	Permutation generator parameter	Sensitivity	Number of digits
Initial values			Initial values		
U_0	-	-	U_0	10^{-4}	4
V_0	-	-	V_0	10^{-4}	4
W_0	-	-	W_0	10^{-4}	4
Lorenz parameters			Lorenz parameters		
A	10^{-3}	3	A	10^{-4}	4
B	10^{-3}	3	B	10^{-4}	4
C	10^{-3}	3	C	10^{-4}	4
g parameter for			g parameter for		
U	10^{-4}	4	U	10^{-5}	5
V	10^{-4}	4	V	10^{-5}	5
W	10^{-4}	4	W	10^{-5}	5
Combination	10^{-21}	21	Combination	10^{-39}	39

4.4.2 Statistical analysis

The encrypted signal was subjected to a statistical analysis to determine whether information can be obtained about the dynamics of the chaotic system. Figures 4.1-b and 4.3 show the time-domain encrypted signal and the autocorrelation of this signal,

respectively. The nearly flat autocorrelation illustrates the randomness of the signal and thus the difficulty in exploiting it via correlation techniques. Further, Figure 4.4 shows the low cross-correlation between the U state variable with and without the injected image data. This clearly indicates the randomness of the generated signals and the difficulty in exploiting them via correlation techniques.

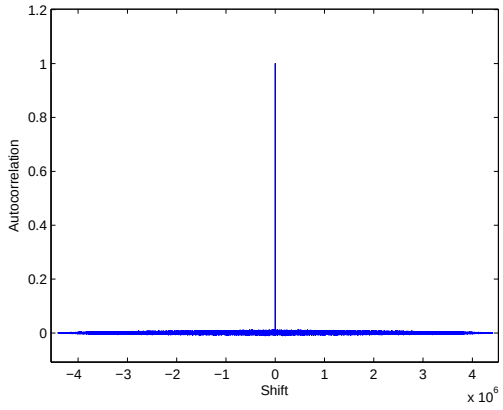


Figure 4.3: The autocorrelation of the signal in Fig. 4.1-b.

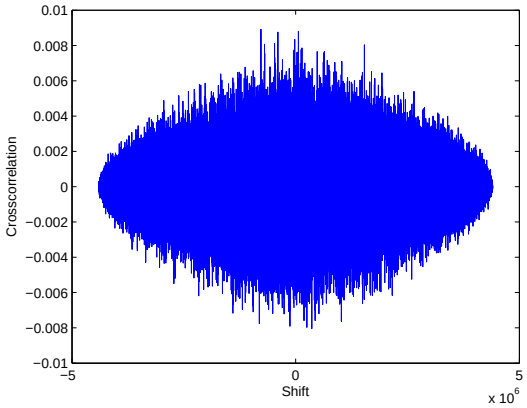
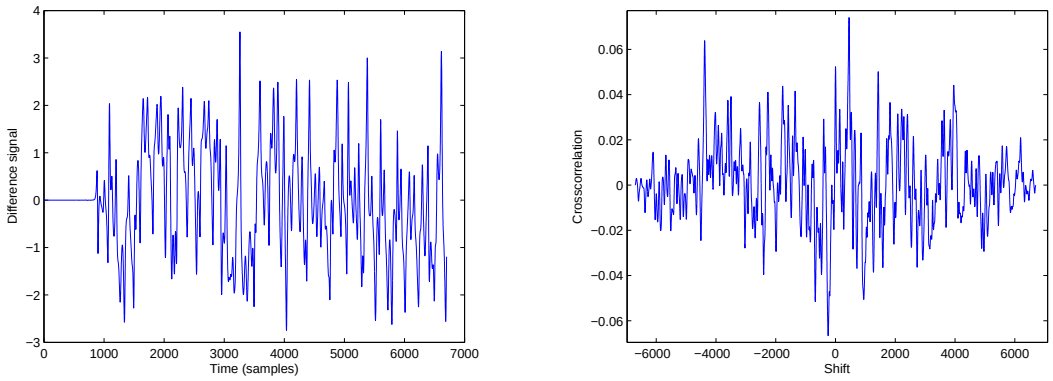


Figure 4.4: The cross-correlation of the encrypted signals with and without an injected image.

4.4.3 Differential analysis

In a differential attack, the cryptanalyst is assumed to have the capability of modifying individual values of the plaintext and observing the resulting encrypted signal. If such

a change results in a significant change in the signal, then the attack is considered to be inefficient and impractical. The proposed cipher was used to encrypt a JPEG color image with dimensions 284×177 and the same image with only one bit in the first byte changed. This change results in a different signal injected into the Lorenz generator which will propagate through the dynamics and result in a different trajectory. Figure 4.5-a presents the difference between the signal trajectories, and their cross-correlation is shown in Figure 4.5-b. This shows that a small variation in the plaintext results in significant changes in the encrypted signal.



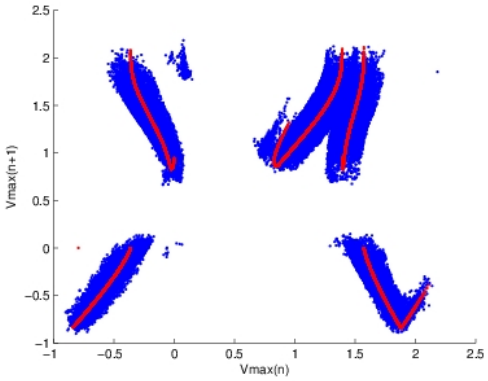
(a) Difference signal (b) Cross-correlation

Figure 4.5: (a) The difference between the signals for two encrypted images which differ in one bit, and (b) the cross-correlation of the two signals.

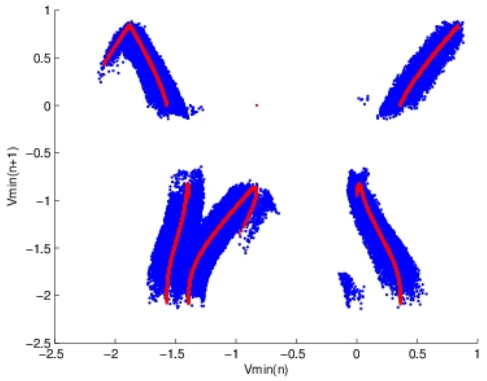
4.4.4 Lorenz attacks

Since the proposed cipher is based on a discrete Lorenz system, it is important to consider attacks against the corresponding continuous Lorenz system. These attacks are based on the system synchronization and are primarily return map and geometry attacks [61, 62, 63, 64, 76]. The attacker uses the encrypted signal as the driving signal to synchronize their master generator in an attempt to obtain the Lorenz generator parameters. However, these methods cannot be used against the proposed system because a signal from the permutation generator is injected into the second state equation of the driven subsystem. To illustrate this, Figure 4.6 shows the return map of the proposed cipher obtained using the approach in [61]. In Figure 4.6-a and 4.6-b, the red dots represent the return map of the Lorenz generator, and the blue dots represent the return map of the proposed system without the image data injected.

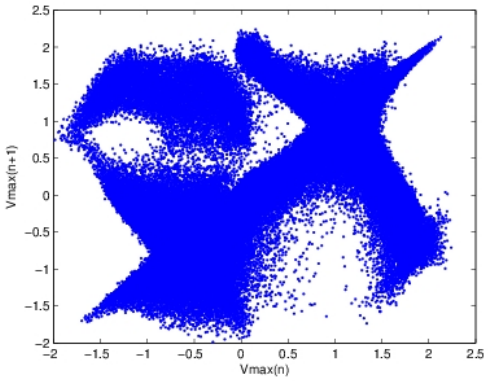
This indicates that the return map of the master Lorenz generator is blurred by the signal from the permutation generator. Figure 4.6-c and 4.6-d shows the effect of the injected image data on the return map. Clearly, the image data increase the blurriness of the return map, which is desirable. The importance of a blurry return map is that it prevents an attacker from extracting any useful information about the generator control parameters and hence breaking the cipher [61]. The return map of the proposed cipher is blurred using random signals from the permutation generator and the image data. This is significantly better than using a periodic signal to blur the return map as in [62], which was broken in [63].



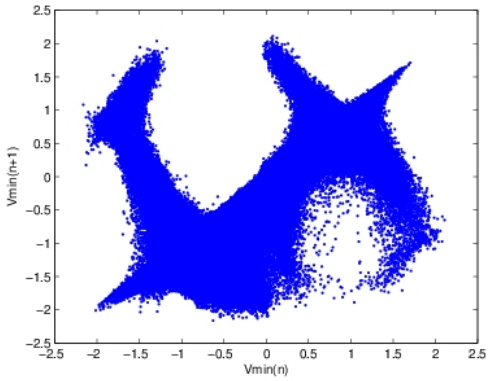
(a) V_{max} return map (without image data)



(b) V_{min} return map (without image data)



(c) V_{max} return map (with image data)



(d) V_{min} return map (with image data)

Figure 4.6: The return map of the proposed dual chaotic cryptosystem based on the Lorenz generator: (a) and (b) without an injected image, and (c) and (d) with an injected image.

Geometry attacks [64, 76] will also be ineffective because the discrete Lorenz generator used in the proposed technique has additional parameters g_i, one in each of the three difference equations. Since the g_i can be assigned different values, they

represent three additional key parameters beyond those of the differential equations of the continuous Lorenz generator, and so provide an additional level of security. This greatly decreases the probability of a geometric attack on the proposed cipher being successful.

4.4.5 Execution time

The time to execute a cipher is an important factor in many applications. The time required for the proposed cipher is compared in Table 4.2 with several ciphers in the literature. A 256-level gray-scale Lena image with dimensions 512×512 ($262,144$ bytes) was encrypted and decrypted using a Pentium dual core 1.6 GHz processor. The proposed cipher was executed 1000 times, and the average times for the two operations are shown in the table. The encryption process has one phase (update phase) where produces the encrypted image. Conversely, the decryption process has two phases, extraction where the image data are retrieved using the encrypted signal and the state variables of the decryption master generator, and update which is identical to that for encryption. This explains why decryption is slower than encryption, as indicated in Table 4.2. Considering the differences in the hardware used for each cipher, the performance of the proposed cipher is comparable. From the table, the proposed cipher is faster than the other 3D chaotic systems and the 4D hyper-chaotic system except for the approach in [77], which is faster by about a factor two. However, the CPU used in [77] is more powerful than that employed here. The proposed cipher is faster than the technique in [75] which uses a similar approach for encryption. Further, it is better than most of the 1D and 2D systems which are much simpler. Consequently, the proposed cipher is superior to most existing chaotic ciphers from a speed perspective, and thus, it is well suited for real-time applications.

4.5 FPGA Implementation

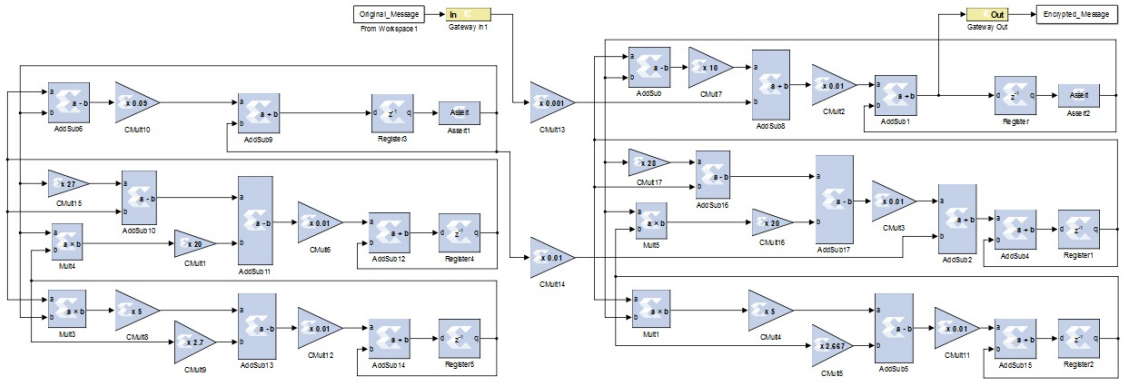
The proposed encryption system was implemented using a field-programmable gate array (FPGA). An FPGA is a high-performance data processing device. Because no CPU governs the entire chip and no sequential instructions have to be processed, typically thousands of operations can be performed in parallel during every clock cycle. The Xilinx ISE software is used to convert the developed models into bit stream files which can be downloaded to the FPGA. Further, a Xilinx tool is used to extend

Table 4.2: Encryption and Decryption Execution Times for Several Ciphers

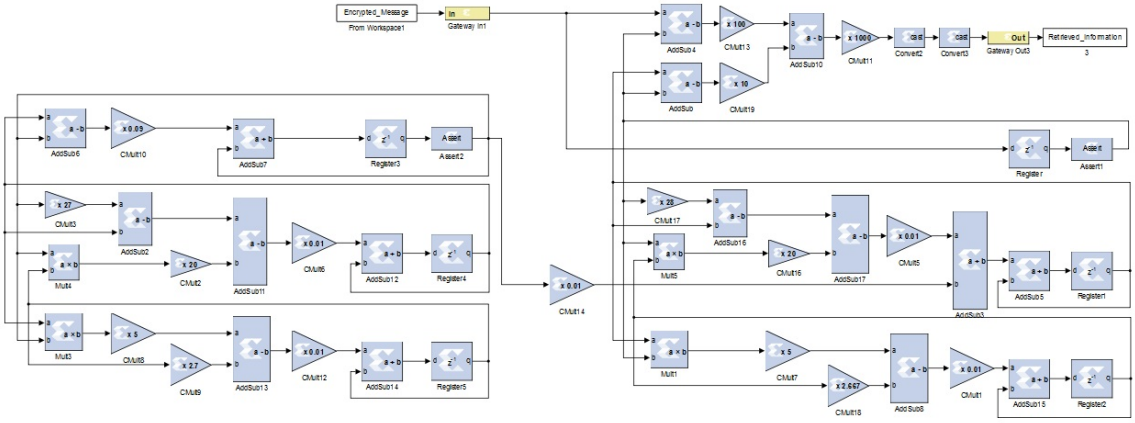
Cipher	System characteristics	Dimension of the chaotic generator	Execution Time
Ref. [59]	P. Dual Core 2.0 GHz	1D	256-level 512×512 (210 ms) 9.56 Mbps
Ref. [68]	Pentium IV 2.1 GHz	1D	74.4 Mbps
Ref. [70]	P. Dual Core 2.7 GHz	1D	15.6 Mbps
Ref. [78]	Intel Core i7-2600 3.4 GHz	1D	256-level 256×256 (0.1789 s) 2.8 Mbps
Ref. [71]	Intel Core 2 Duo 2.1 GHz	2D	22.6 Mbps
Ref. [72]	Intel Core 2 Duo 3.0 GHz	2D	44.9 Mbps
Ref. [79]	Intel Core i32350 2.3 GHz	2D	256-level 512×512 (0.156 s) 13 Mbps
Ref. [80]	Intel Core i7-2600	2D	max. speed < 3.2 Mbps
Ref. [73]	P. IV 1 GHz	3D	8 Mbps
Ref. [74]	Intel Core i5 2.27 GHz	3D	15.4 Mbps
Ref. [75]	P. Dual Core 1.6 GHz	3D	37.7 Mbps
Ref. [77]	Intel Core i5-2400 3.1 GHz	3D	256-level 512×512 (4.79208 ms) 421 Mbps
Ref. [81]	Intel P. Dual Core 3.2 GHz	3D	20.6 Mbps
Ref. [82]	CPU 2.4 GHz	4D (hyper-chaotic)	256-level 256×256 (0.4 s) 1.27 Mbps
Proposed cipher	P. Dual Core 1.6 GHz	3D	256-level image 512×512 (262,144 bytes) Encryption 0.0133 s (150 Mbps) Decryption 0.0252 s (79 Mbps)

Simulink models in MATLAB to provide a modeling environment that is well suited to hardware design. This tool provides high-level abstractions that are automatically compiled into the FPGA. It also provides access to the underlying FPGA resources through low-level abstractions, allowing for highly efficient FPGA designs. Using the Xilinx tool in MATLAB, the encryption and decryption blocks were constructed and are shown in Figure 4.7. Figure 4.7-a shows the implementation of the permutation generator on the left and the implementation of the master generator on the right. The image data (integers) are converted to floating-point numbers and injected after scaling into the master generator. For decryption, (4.3) and (4.4) are used, and their implementation is shown in Figure 4.7-b.

The FPGA implementation was used to encrypt and decrypt a 256×256 gray-scale image of size $65,536$ bytes. The original image, encrypted signal, and recovered



(a) Encryption implementation



(b) Decryption implementation

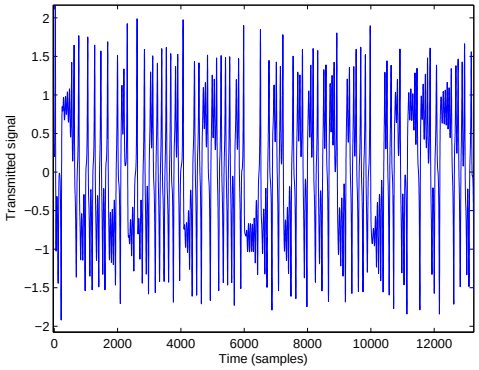
Figure 4.7: FPGA hardware implementation using the Xilinx tool in MATLAB: (a) encryption, and (b) decryption.

image are presented in Figure 4.8. The autocorrelation is also shown to indicate the randomness of the encrypted signal. These results verify the performance of the FPGA implementation.

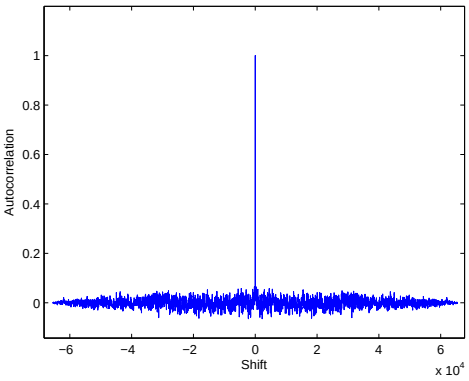
4.6 Conclusion

In this chapter, an encryption cipher was proposed based on a dual chaotic system for secure real-time image applications. A 3D discrete Lorenz attractor was employed with non-autonomous modulation. The dynamics of the master chaotic generator are permuted by both the image data and the output of a second (permutation) chaotic generator. This prevents an eavesdropper from synchronizing their master generator with the encrypted signal since they have no information about the signal from the

(a) Original image



(b) Encrypted signal



(c) Autocorrelation

(d) Recovered image

Figure 4.8: FPGA implementation performance: (a) the original image, (b) the encrypted signal, (c) the autocorrelation of the encrypted signal, and (d) the recovered image.

second generator. The effects of using finite precision arithmetic were mitigated using integers to represent the image pixels and rounding the floating-point values to integers during decryption process. Using both analysis and simulation, it was shown that the security of the proposed scheme is excellent, and the execution speed is very suitable for secure real-time image applications.

Chapter 5

Secret Key Generation using Chaotic Signals over Frequency Selective Fading Channels

5.1 Introduction

This chapter presents a practical key generation algorithm based on the reciprocity of wireless fading channels. A broadband chaotic signal is employed for transmission so that the fading is frequency selective. In this case, signal components in the frequency-domain spaced greater than the coherence bandwidth of the channel can be considered uncorrelated. The proposed algorithm exploits this property to generate a unique shared key between two parties. The non-periodicity of the chaotic signal provides a unique signal for key generation which can be used even with static fading channels, so that probing signals can be sent through the channel several times during the channel coherence time. The proposed approach is robust to timing differences between the parties because the frequency spectrum of the signals is employed. A technique for information reconciliation is presented which does not reveal any information about the values used to generate the key. The randomness of the key is confirmed, and the effects of additive white Gaussian noise (AWGN) and timing differences on the performance of the algorithm are examined.

5.2 Literature Review

Key establishment via the exchange of secret keys is essential in many applications for encryption to preserve data confidentiality and integrity. The challenge is for two parties to exchange keys through an insecure channel without prior knowledge of each other and without revealing them to an eavesdropper. Diffie-Hellman key exchange [54] provides a solution, but it suffers from computational complexity and it lacks authentication (however, an algorithm for authentication was proposed in [54]). Although a trusted third party can be employed to solve the problem of authentication for fixed infrastructure networks, it is not suitable for wireless communication networks which have a dynamic topology. Further, many mobile devices have limited computational power.

The reciprocity of wireless fading channels means that the multipath fading statistics such as the signal gains, delays, and phase-shifts are approximately the same between two parties. These statistics are highly correlated within the coherence time of the channel [83]. The randomness of the channel due to the geometry, user movements and/or the motion of nearby objects creates an environment which can be exploited to achieve information-theoretic secrecy [84]. Spatial variations in the channels result in statistics which are unique for the two parties, and an eavesdropper located a sufficient distance from them will have channel characteristics which are uncorrelated with theirs [85]. As a consequence, there has been much research on physical layer techniques to establish key agreement protocols in wireless communication systems [86].

There are three steps to generating a shared key using a wireless channel. First, a probing signal is transmitted which results in data at the receivers of the legitimate parties which are highly correlated but not necessarily identical. Differences can occur because of the timing of the signals due to half duplex transmission, and the independence of the noise at the receivers. Quantization is typically used to convert the received signals into a sequence of key bits. The quantization threshold can be determined in several ways. In [87], the mean and standard deviation of the samples were used to determine the threshold, while in [88] the fade duration of the received signal strength (RSS) was employed. An adaptive threshold was presented in [89]. In this case, the received signal is divided into groups and a threshold calculated for each group. This has the additional effect of increasing the randomness of the generated key. In [90], the cumulative distribution function (CDF) of the RSS was

used to obtain quantization levels such that each group contains the same number of values. While this improves the randomness of the key and the number of bits generated, increasing the number of levels can increase the key bit mismatch rate.

Discrepancies in the bits obtained by the two parties are corrected in the next two steps, key reconciliation and privacy amplification. In key reconciliation, messages are exchanged so they can agree on the key bit strings. These messages reveal a certain amount of information about the keys. In [91], a protocol was presented which provides a lower bound on the information leakage. The problem of information leakage during reconciliation can be solved using privacy amplification. In this last step, common bits from both key bit strings are discarded to reduce the amount of information disclosed [92]. In [93], an efficient protocol was used to extract the key bits without revealing information to an eavesdropper. In particular, only one bit of information is revealed in each communication when the parties agree, and this bit is discarded to maintain the secrecy of the remaining bits (which are used for the key).

Key generation using a wireless channel can be done in several ways. The RSS was the first parameter used to extract secret keys [87, 88, 89]. Recently, this approach was used for vehicular communication networks [94]. The advantage of using the RSS is the ease of measuring the signal amplitude and the availability of this statistic in most communication systems. However, key generation may be limited as many consecutive measurements can yield only a single bit due to correlated values. This is a significant issue when the channel is static or slowly time-varying. In [95], a multiple-antenna system was used to increase the key rate based on the fact that multiple antennas can provide channel diversity, but this increases the system complexity. Fractional interpolation was used in [90] to extract uncorrelated bits which are not affected by the non-simultaneous measurements at the receivers. However, this approach results in a low key rate compared to other methods.

The channel fading parameters can also be used for key generation. For example, an algorithm was proposed in [88] to generate a secret key using channel fade durations and level crossings. In [96, 97, 98], an ultra-wideband (UWB) signal was used for key generation. The large bandwidth and the corresponding fine-time resolution of these signals provides significant information about the channel impulse response. Other approaches are based on the phase reciprocity of the channel [99, 100, 101]. An advantage over using the RSS is that a signal received through a fading channel typically has a uniform phase distribution [102]. In [103], key generation based on the phase of an OFDM signal was proposed. A cooperative key generation protocol was

developed in [104] where two users extract the phase of the fading channel with the aid of relay node(s). The drawbacks of using phase reciprocity are the implementation complexity compared to RSS and channel parameter techniques, and the increased sensitivity to noise [105]. The channel impulse response can also be used to establish a key between users [90, 106, 107, 108]. Although the resulting key rate may be less than with techniques based on phase randomness, this approach is more robust to noise. Most systems use an omnidirectional antenna for key extraction, but an electronically steerable antenna can also be employed [109]. Techniques for key agreement in a cooperative wireless communication system were developed in [104, 110].

The performance of key generation algorithms is typically evaluated using three parameters. The key disagreement probability (KDP) is the probability of bit mismatch between the keys generated by the two parties, and thus characterizes the robustness of the algorithm. The key generation rate (KGR) is the rate at which key bits are generated, and so is a measure of the algorithm efficiency. The third parameter is the randomness of the key bits, and this can be determined using standard techniques for evaluating randomness such as the autocorrelation. This provides a measure of the resistance of the key against attacks.

In next section, a new algorithm to generate a secret key between two parties is presented which is based on the characteristics of a wireless fading channel. Instead of using the time-domain characteristics of the received signal such as the RSS, phase or fading parameters, the proposed technique employs the frequency-domain characteristics to generate the key. Based on the coherence bandwidth of the channel, the effect of a frequency selective channel on the spectrum of a broadband signal is exploited to generate a shared key. The randomness of this key is inherited from the characteristics of the fading channel.

Chaotic signals are noise-like broadband signals which are suitable for secure applications because they are non-periodic, ergodic and sensitive to initial conditions. The non-periodic property allows the channel to be probed at a rate greater than the coherence time of the channel without compromising the key randomness. Further, initial values are required for synchronization which allows the users to choose a unique probing signal for transmission. This will be discussed in Section 5.6. For these reasons, and without loss of generality, this proposal considers a chaotic signal as the broadband probing signal for generating a shared key.

5.3 The Proposed Algorithm

Chaos is a phenomenon describing the behavior of many non-linear dynamic systems. Chaotic signals are non-periodic, so they have a spectrum which has significant energy over a wide range of frequencies, and thus are broadband signals. Further, chaotic signals are sensitive to the system initial conditions, so any deviation will result in differences in the output that grow exponentially [1]. This makes it intractable to predict these signals without prior knowledge of the initial conditions. Since they are non-periodic and irregular, chaotic signals are uncorrelated and have characteristics similar to those of a random signal.

When the signal bandwidth is greater than the coherence bandwidth of the channel, the fading is frequency selective, and the channel effects on signal components separated by this bandwidth are uncorrelated. The proposed algorithm exploits this uncorrelated effect on the frequency spectrum of a chaotic signal to generate a shared key. The magnitudes of the frequency spectra of the transmitted and received signals are obtained using a discrete Fourier transform (DFT). These magnitudes are used to generate the keys bits, as described below. The uncorrelated effect of the channel on the chaotic signal results in a random sequence of zeros and ones, as will be illustrated in Section 5.7.

5.3.1 Probing the channel

Both legitimate parties probe the channel using a broadband chaotic signal with a time difference that is within the coherence time of the channel. The received signal is sampled and the frequency spectrum determined using an N-point DFT which has a frequency resolution greater than the coherence bandwidth of the channel. According to the frequency spectrum of the chaotic signal used, the algorithm uses the first M of the N DFT points to generate the key. These M DFT points represent the frequencies for which the chaotic signal has a significant magnitude. This truncation is used to ensure the randomness of the resulting key bits and an acceptable KDP.

The channel effect on the transmitted signal is determined by the differences between the magnitudes of the M DFT values of the received signal and the corresponding values of the transmitted signal. As the received signal is subject to large scale fading, the M values are normalized so that the maximum value is 1 for both the transmitted and received signals. The M differences provide a difference vector of positive and negative values that represents the effect of the channel on the signal

spectrum

$$DV = DFT_{\text{received}} - DFT_{\text{transmitted}}. \tag{5.1}$$

Assuming the channels are identical and ignoring any sources of error, if both parties are synchronized (i.e., they have the same chaotic generator with the same initial conditions), their difference vectors (and in particular the signs of the vector elements), will be identical. In a real system, there are discrepancies between the difference vectors due to the independent noise at the receivers, the half duplex communications, and the timing errors in sampling the received signals by the two parties. This can cause a mismatch in the keys generated. To mitigate this mismatch, vector elements with a small magnitude are discarded and not used for key generation. Conversely, vector elements with a large magnitude indicate that the channel has had a significant effect on the probing signal at the corresponding frequencies. Therefore, these elements are considered as candidates in the key generation process.

5.3.2 Threshold selection

To avoid mismatches in the key bits generated due to small difference vector elements, a threshold is used so that only elements with a large magnitude are used to generate the keys. The standard deviation of the M difference vector elements can be used as an adaptive (dynamic) threshold. Alternatively, since the difference vectors are obtained from normalized DFT values, a fixed threshold can be employed. From a security perspective, a dynamic threshold is preferred as it will be similar for the two parties, but can be very different for an eavesdropper due to the effects of the channel. The performance with both dynamic and fixed thresholds will be presented in Section 5.5.

Difference vector elements with a magnitude greater than the threshold are considered as "candidate elements" for key generation, while those with a magnitude below the threshold are discarded. To generate the key bits, positive candidate elements are assigned a one, while negative values are assigned a zero, so that a vector of bits is obtained. Due to noise and other effects, candidate elements with values near the threshold may be chosen by one party but not the other, which will create a mismatch in the bits generated. Thus, the next step is for the two parties to agree on the shared candidate elements and discard the others. This process is called information or key

reconciliation and is described in the next section.

5.3.3 Information reconciliation

The objective of information reconciliation is for the legitimate parties to agree on the shared key by discarding mismatched bits, and this is done by exchanging messages between users. The proposed algorithm uses the M difference vector elements to construct a reconciliation message m. Opposed to the bit generation process, which depends only on the signs of the candidate elements, information reconciliation employs the indices of the elements in the difference vectors to construct m. In particular, ones are placed at the indices of the candidate elements and zeros otherwise to obtain a binary vector m of length M. These messages are exchanged between the legitimate users to agree on the shared candidate elements for key generation.

From the security perspective, m does not reveal any information about the value of the candidate elements used to generate the key, as only their locations in the difference vector are used. However, sending these reconciliation messages through an insecure channel allows an eavesdropper to obtain information about the positions (indices) of the candidate elements. Although this information does not disclose any information about the values of these elements (which is required by an eavesdropper to obtain the key), the messages are first masked to make it more difficult for an eavesdropper to determine even the positions. This procedure is outlined below.

Each message m is divided into $K = \lceil M/L \rceil$ sub-blocks g_i, $i = 1, \ldots, K$, of length L bits (corresponding to L elements), so that

$$m = [g_1 g_2 \ldots g_K] .$$

If necessary, the last sub-block is padded with zeros so it has length L. Each sub-block is then masked separately as will be shown later. The masks should be reliable, so they are created using the indices of the two elements in the difference vectors with the largest magnitude i.e., those most affected by the channel. These indices are denoted A and B in order of magnitude, and can take values from 0 to $M - 1$. Since these values depend on the channel between the two parties and the chaotic signal transmitted, they can be considered random and typically change with each transmission. Each party, according to the received probing signal, has a pair of values (A, B). One party (called Alice) has (A_1, B_1), and XORs the L least significant bits of the binary representation of A_1 with her message m_A. The other party (called

Bob) has (A_2, B_2), and XORs the L least significant bits of the binary representation of B_2 with his message m_B. The result of these operations is

$$\begin{array}{ll} \text{Alice:} & C_A = [C_{a_1} C_{a_2} \dots C_{a_K}], \quad \text{where } C_{a_i} = A_1 \oplus g_{a_i} \\ \text{Bob:} & C_B = [C_{b_1} C_{b_2} \dots C_{b_K}], \quad \text{where } C_{b_i} = B_2 \oplus g_{b_i} \end{array} \tag{5.2}$$

Then the syndrome of an (L, n) binary linear block code is calculated for each sub-block using a parity check matrix H for the code, where L is the length of a sub-block which corresponds to the codeword length, so the syndrome length is $L - n$. The resulting messages

$$\begin{array}{ll} \text{Alice:} & T_A = [T_{a_1} T_{a_2} \dots T_{a_K}], \quad \text{where } T_{a_i} = C_{a_i} H^T \\ \text{Bob:} & T_B = [T_{b_1} T_{b_2} \dots T_{b_K}], \quad \text{where } T_{b_i} = C_{b_i} H^T \end{array} \tag{5.3}$$

are exchanged between the two parties. Alice also masks her message m_A using B_1 and Bob also masks his message m_B using A_2 to obtain

$$\begin{array}{ll} \text{Alice:} & S_A = [S_{a_1} S_{a_2} \dots S_{a_k}], \quad \text{where } S_{a_i} = (B_1 \oplus g_{a_i}) H^T \\ \text{Bob:} & S_B = [S_{b_1} S_{b_2} \dots S_{b_k}], \quad \text{where } S_{b_i} = (A_2 \oplus g_{b_i}) H^T \end{array} \tag{5.4}$$

Then Alice compares S_A with the received T_B, and Bob compares S_B with the received T_A

$$\begin{array}{ll} \text{Alice:} & \text{if } (B_1 \oplus g_{a_i}) H^T = (B_2 \oplus g_{b_i}) H^T, \text{ sub-block } g_{a_i} \text{ is retained} \\ \text{Bob:} & \text{if } (A_2 \oplus g_{b_i}) H^T = (A_1 \oplus g_{a_i}) H^T, \text{ sub-block } g_{b_i} \text{ is retained} \end{array} \tag{5.5}$$

Thus, the candidate elements corresponding to the sub-blocks that match are used to generate the key, while the others are discarded.

In summary, the uncorrelated DFT elements ensure the randomness of the bits obtained. It will be shown that this algorithm is robust to timing errors between the parties and also noise at the receivers. Moreover, sending the syndromes has two advantages. First, it disguises the reconciliation messages, and second, it shortens the length of the reconciliation messages to $(L - n)K$ bits compared to M bits.

5.4 Chaotic Signals

In this section, the transmission of chaotic signals over a wireless fading channel is examined. Due to fading, the impulse response of this channel can be considered a random process [83]. The channel coherence time is the time during which it can be considered static (time-invariant). Within this time, channel reciprocity ensures that the channels between the two users are approximately equal. This property is exploited to generate a shared key between the parties using chaotic signals. First, communication with a wide-sense stationary (WSS) signal is considered, and then an ergodic chaotic signal is employed which is WSS.

5.4.1 WSS signals over a fading channel

Assume that the channel impulse response is $h(t)$ and the transmitted signal $X(t)$ is WSS. The received signal $Y(t)$ is then

$$Y(t) = X(t) * h(t) = \int_{-\infty}^{\infty} X(t-\tau)h(t)d\tau, \tag{5.6}$$

where $*$ denotes convolution. The corresponding autocorrelation is

$$\begin{aligned} R_y(t_1,t_2) &= E[Y(t_1)Y(t_2)] \\ &= E\left[\int_{-\infty}^{\infty} X(t_1-\beta)h(\beta)d\beta \int_{-\infty}^{\infty} X(t_2-\gamma)h(\gamma)d\gamma\right] \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} E[X(t_1-\beta)X(t_2-\gamma)]h(\beta)d\beta h(\gamma)d\gamma \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} R_x(t_1-\beta,t_2-\gamma)h(\beta)d\beta h(\gamma)d\gamma. \end{aligned}$$

Let $\tau = t_2 - t_1$, so that

$$\begin{aligned} R_y(t_1,t_2) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} R_x(t_1-\beta,t_1+\tau-\gamma)h(\beta)d\beta h(\gamma)d\gamma \\ &= \int_{-\infty}^{\infty}\left\{\int_{-\infty}^{\infty} R_x(t_1,t_1+\tau+\beta-\gamma)h(\beta)d\beta\right\}h(\gamma)d\gamma \\ &= \int_{-\infty}^{\infty}\left\{\int_{-\infty}^{\infty} R_x(\tau+\beta-\gamma)h(\beta)d\beta\right\}h(\gamma)d\gamma. \end{aligned} \tag{5.7}$$

$R_x(\tau)$ is an even function so that $R_x(\tau) = R_x(-\tau)$ which gives

$$\begin{aligned} R_y(t_1,t_2) &= \int_{-\infty}^{\infty}\left\{\int_{-\infty}^{\infty} R_x(-\tau+\gamma-\beta)h(\beta)d\beta\right\}h(\gamma)d\gamma \\ &= \int_{-\infty}^{\infty}\left\{R_x(-\tau+\gamma)*h(-\tau+\gamma)\right\}h(\gamma)d\gamma \\ &= \int_{-\infty}^{\infty} R_x(\tau-\gamma)*h(-(\tau-\gamma))h(\gamma)d\gamma, \end{aligned}$$

and therefore

$$R_y(\tau) = R_x(\tau) * h(-\tau) * h(\tau). \tag{5.8}$$

Equation (5.8) indicates that the autocorrelation of the received signal depends only on the time difference τ. Thus, the received signal is WSS since the transmitted signal is WSS.

5.4.2 Chaotic signals over a fading channel

Now consider the case that the transmitted signal $X(t)$ is an ergodic chaotic signal which is WSS, and the corresponding received signal is $Y(t)$. The correlation between the transmitted and received signals at times t_1 and t_2, is

$$r_{xy}(t_1, t_2) = \frac{C_{xy}(t_1, t_2)}{\sigma_x(t_1)\sigma_y(t_2)},$$

where $\sigma_x(t_1)$ and $\sigma_y(t_2)$ are the corresponding variances and $C_{xy}(t_1, t_2)$ is the autocovariance given by

$$C_{xy}(t_1, t_2) = E\left[\{X(t_1) - \mu_x(t_1)\}\{Y(t_2) - \mu_y(t_2)\}\right],$$

where $\mu_x(t_1)$ and $\mu_y(t_2)$ are the corresponding statistical means. As the chaotic signal is ergodic

$$C_{xy}(\tau) = R_{xy}(\tau) - \mu_x\mu_y, \tag{5.9}$$

where $R_{xy}(\tau)$ is the cross-correlation between $X(t)$ and $Y(t)$. The correlation coefficient is defined as

$$r_{xy}(t_1, t_2) = \frac{R_{xy}(\tau) - \mu_x\mu_y}{\sigma_x\sigma_y}. \tag{5.10}$$

The two parties (Alice and Bob) share the same channel, but an eavesdropper (called Eve) has different channels with Alice and Bob. If Alice and Bob transmit the same chaotic signal $X(t)$, the received signals are

$$\begin{aligned} Y_{ba}(t) &= X(t) * h_1(t) + n_1(t), \\ Y_{ab}(t) &= X(t) * h_1(t) + n_2(t), \end{aligned}$$

where the subscript ab denotes the transmission from Alice to Bob. The signals

received by Eve from Alice and Bob are

$$\begin{aligned} Y_{be}(t) &= X(t) * h_3(t) + n_3(t), \\ Y_{ae}(t) &= X(t) * h_4(t) + n_4(t). \end{aligned}$$

The cross-correlation between Y_{ba} and Y_{ab} is

$$\begin{aligned} R_{y_{ba},y_{ab}}(\tau) = E[Y_{ba}(t)Y_{ab}(t+\tau)] \\ = E\left[\left(n_1(t) + \int_{-\infty}^{\infty} X(t-\beta)h_1(\beta)d\beta\right)\left(n_2(t+\tau) + \int_{-\infty}^{\infty} X(t+\tau-\gamma)h_1(\gamma)d\gamma\right)\right] \\ = E\left[\left(\int_{-\infty}^{\infty} X(t-\beta)h_1(\beta)d\beta\right)\left(\int_{-\infty}^{\infty} X(t+\tau-\gamma)h_1(\gamma)d\gamma\right)\right. \\ + \left(n_1(t)\int_{-\infty}^{\infty} X(t+\tau-\gamma)h_1(\gamma)d\gamma\right) \\ + \left(n_2(t+\tau)\int_{-\infty}^{\infty} X(t-\beta)h_1(\beta)d\beta\right) \\ \left. + (n_1(t)n_2(t+\tau))\right] \end{aligned} \tag{5.11}$$

Since $n(t)$ and $X(t)$ are independent, $E[n(t)X(t)] = E[n(t)]E[X(t)] = 0$, so that

$$R_{y_{ba},y_{ab}}(\tau) = R_x(\tau) * h_1(-\tau) * h_1(\tau) + R_{n_1,n_2}(\tau), \tag{5.12}$$

where $R_{n_1,n_2}(\tau)$ is the cross-correlation of the noise. The cross-correlation in (5.12) is the same as the autocorrelation given in (5.8) for both Alice and Bob except for the noise term, which can be assumed to be small. Clearly, from (5.10) and (5.12) the correlation between the two received signals at Alice and Bob is maximum if the signals are synchronized, so that the sampled sequences are the same. In a real communication system, there will be timing errors in sampling the signals at the two locations. As the transmitted signal is non-periodic and the number of samples is finite, this will affect the frequency spectrum of the signal. The effect of timing errors on the performance of the proposed algorithm will be examined in Section 5.7.

In general, for a non-linear dynamic system the mixing property causes the autocorrelation to decay to zero for large τ, which is known as correlation splitting. This indicates that the system states become statistically independent if they are separated by a sufficiently large time interval [111]. For a chaotic system, the correlation splitting is due to the instability of the chaotic trajectories [112], and as a consequence the autocorrelation of the output decays rapidly. For most chaotic systems this decay is exponential, with the rate of decay a function of the system [23, 113]. If there is

a small timing difference between Alice and Bob, most of the sampled values will be the same, so the frequency spectra will be similar. This will be examined in Section 5.7.

From a security perspective, the relationship between the received signals at Eve and those at Alice and Bob is important as this determines the ability of Eve to extract key bits. The cross-correlation between the signals sent from Alice to Bob and received by Eve from Bob is

$$\begin{gathered} R_{y_{be},y_{ab}}(\tau) = E[Y_{be}(t)Y_{ab}(t+\tau)] \\ = E\left[\left(n_3(t) + \int_{-\infty}^{\infty} X(t-\beta)h_3(\beta)d\beta\right)\left(n_2(t+\tau) + \int_{-\infty}^{\infty} X(t+\tau-\gamma)h_1(\gamma)d\gamma\right)\right] \\ = R_x(\tau) * h_3(-\tau) * h_1(\tau) + R_{n_2,n_3}(\tau) \end{gathered} \tag{5.13}$$

The convolution of the two impulse responses $h_3(-\tau)$ and $h_1(\tau)$ in (5.13) can be reformulated as

$$h_3(-\tau) * h_1(\tau) = \int_{-\infty}^{\infty} h_3(-\tau+t)h_1(t)dt = \int_{-\infty}^{\infty} h_3(t-\tau)h_1(t)dt = R_{1,3}(\tau) \tag{5.14}$$

Since two channels geometrically separated by more than half a wavelength of the carrier frequency (or the same channel used at time intervals separated by more than the coherence time of the channel [85]), can be considered independent, $R_{1,3}(\tau)$ is near zero (i.e., the channel impulse responses are uncorrelated). Thus from (5.13) the correlation between the signals received by Eve and those received by Alice and Bob is small, which indicates that Alice and Bob can generate a shared secret key.

5.5 Key Bit Generation

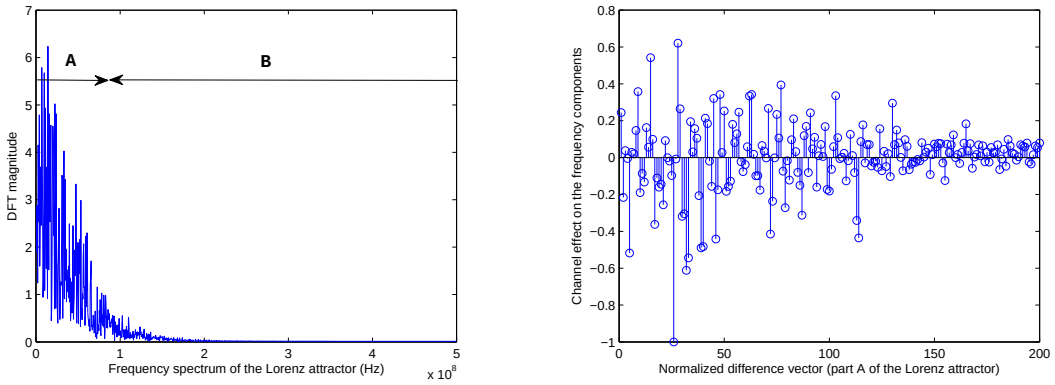
In this section, the proposed algorithm is evaluated to show the effectiveness in generating key bits. The six tap fading channel from [114] is employed where each tap has a delay and amplitude which are Rayleigh random variables. For the static fading channel case, the delays and normalized powers are given in Table 5.1. The carrier frequency is 900 MHz. A 2048-point DFT is employed with a frequency resolution of 484 kHz, which is greater than the coherence bandwidth of the channel. Thus the frequency components used to generate the key are uncorrelated. AWGN is assumed

with an average signal to noise ratio (SNR) of 24.5 dB.

Table 5.1: Six Tap Static Channel Parameters

Tap Number	1	2	3	4	5	6
Delay [μs]	0	0.3	1	1.6	5	6.6
Power [normalized]	0.164	0.293	0.147	0.094	0.185	0.117

The Lorenz chaotic attractor is employed to generate the chaotic signal used as the baseband probing signal, but many other attractors are also suitable. This attractor was chosen because of the complexity of the chaotic signal generated and its dimensionality [1]. It has three state variables denoted x, y and z. The initial values used are $x = -0.635879435665815$, $y = 0.604576974430834$ and $z = 1.248872077476289$. The frequency spectrum of the y state variable obtained using a 2048-point DFT is shown in Figure 5.1-a. This spectrum can be divided into part A which has large values, and part B which has small values. Clearly noise will have a greater effect on part B, so only frequency band A is used to extract the key bits, which corresponds to $M = 196$. The resulting difference vector between the transmitted and received signal spectra is shown in Figure 5.1-b.



(a) Frequency spectrum (b) Difference vector

Figure 5.1: (a) The frequency spectrum of the y state variable of the Lorenz attractor, and (b) the difference vector for the $M = 196$ normalized DFT values in frequency band A.

The standard deviation of the difference vector elements is used to provide a dynamic threshold. As mentioned previously, any vector elements with a magnitude less than the threshold are discarded, and the remaining elements are candidates for

generating the key bits. Here, the difference vectors have length $M = 196$, and $L = 7$ is chosen for the reconciliation messages so there are $K = \lceil M/L \rceil = 28$ sub-blocks of 7 bits. Therefore, seven bit masks are used for each sub-block, and these masks are obtained from the indices (in binary) of the two components in the difference vectors with the largest magnitude, denoted A and B in order of magnitude. According to (5.2), Alice uses A_1 as a mask and Bob uses B_2. Then the syndromes of the $(7, 3)$ simplex code are calculated as in (5.3) for each sub-block. The messages T_A and T_B of length 112 bits are sent by Alice and Bob, respectively.

Each party also generates a version of the message sent by the other party using their own difference vectors as in (5.4). The difference vector elements corresponding to the sub-blocks that match are used to generate the key, and the other elements are discarded. To check the randomness of the keys obtained, one million key bits were generated using the proposed algorithm with the static fading channel given in Table 5.1. Table 5.2 shows the percentage of ones and zeros when using dynamic and fixed thresholds, as well as the average number of bits per measurement with an average SNR of 32 dB. The fixed threshold used was 0.2, as it was found this provides sufficient robustness against noise. These results show that using a dynamic threshold results in a higher key rate compared to using a fixed threshold. Table 5.2 also shows that there is a bias in the bits obtained. This is due to the effect of the channel on the transmitted signal and the fact that the difference vectors are normalized by the maximum values.

Table 5.2: 10^6 Key Bits Generated Using the Proposed Algorithm over a Static Fading Channel

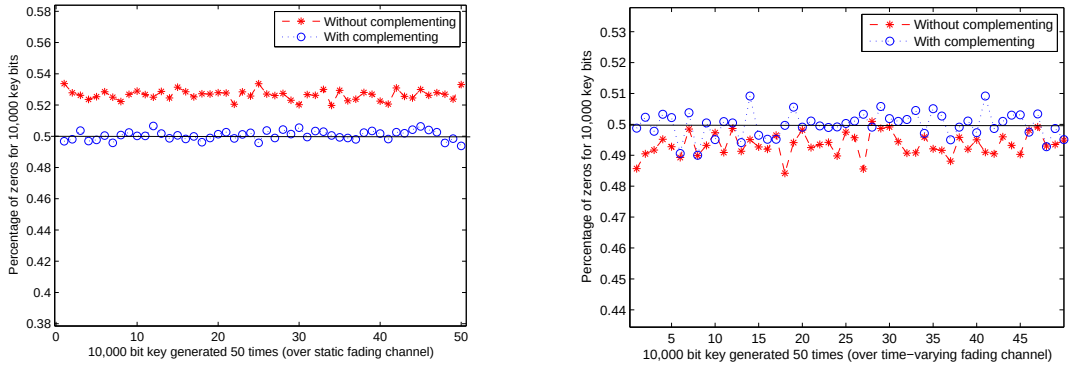
Threshold	Number of 1s	Number of 0s	Percentage of 1s	Number of Iterations	Bits/Iteration
Without Complementing Bits					
Dynamic	473,996	526,004	47.4%	15691	63.70
Fixed	449,306	550,694	45%	22385	44.67
With Complementing Bits					
Dynamic	499,946	500,054	50%	15691	63.70
Fixed	499,677	500,323	50%	22385	44.67

A simple solution to eliminate the bias is to complement the bits every second measurement, so that in these cases the difference vector would be

$$DV = DFT_{\text{transmitted}} - DFT_{\text{received}}. \tag{5.15}$$

The result of generating 1 million key bits using this technique is also given in Table 5.2. Figure 5.2-a shows the percentage of zeros after 50 trials of generating 10,000 key bits with and without complementing the bits every second measurement. This confirms that this approach is effective in removing the bias.

Next, one million key bits were generated using the proposed algorithm with a time-varying fading channel. The 6 channel taps were considered as Rayleigh random variables [83] and changed independently each iteration of the algorithm (i.e., each time a probing signal was transmitted). The results for 50 trials, each generating 10,000 key bits, are shown in Figure 5.2-b, and the results for 1 million key bits are given in Table 5.3. The bias is lower because the channel is not static, but complementing the bits every second measurement removes this bias.



(a) Static fading channel (b) Time-varying fading channel

Figure 5.2: The effect of complementing bits on the bias of 10,000 key bits for 50 trials, (a) static fading channel, and (b) time-varying fading channel.

Table 5.3: 10^6 Key Bits Generated Using the Proposed Algorithm over a Time-Varying Fading Channel

Threshold	Number of 1s	Number of 0s	Percentage of 1s	Number of Iterations	Bits/Iteration
Without Complementing Bits					
Dynamic	505,350	494,650	50.5%	14196	70.44
Fixed	463,622	536,378	46.3%	19286	51.85
With Complementing Bits					
Dynamic	500,197	499,803	50%	14196	70.44
Fixed	500,063	499,937	50%	19286	51.85

The results presented in this section show that the proposed algorithm can be

employed with static or time-varying fading channels. In a communication system, variations from the ideal results given in this section will occur due to noise, channel differences, frequency offset, timing offset (because of half-duplex transmission), and timing errors. The effects of noise and timing errors are important factors that will be considered in Section 5.7 as the others have been studied extensively in the literature and so techniques exist to mitigate their effects.

5.6 Synchronization between legitimate users

Synchronization between the two legitimate users is essential for key generation. Most key generation algorithms in the literature use modulation such as BPSK or QPSK for the probing signals based on standards such as IEEE 802.11/802.15.4. In some cases non-standard but easily created signals are employed. Conversely, a chaotic probing signal is very complex, and requires that the initial values of the chaotic signal generators at the two parties be the same. These values can be transmitted over an insecure channel once the connection is established without detracting from the security of the algorithm, as is the case with other techniques in the literature. This is because the security of the key is based on the channel, and only requires that users be sufficiently far from an eavesdropper to ensure that the channels are independent [85]. As will be shown in Section 5.9, even if an eavesdropper Eve has full knowledge of the initial values, the mismatch between the key for Alice/Bob and the key generated by Eve is almost 50%. To provide additional security, the initial values can be kept secret, but this is not essential.

5.7 Performance analysis

In this section, the key generation rate, key disagreement probability, and key randomness of the proposed algorithm are investigated.

5.7.1 Key generation rate

The key generation rate (KGR) is defined here as the number of key bits obtained per measurement. This is a relevant parameter as it determines the number of probing signals that must be sent to generate the required number of key bits.

The effect of noise and timing errors on the KGR of the proposed algorithm are investigated in this section. The timing error is the time difference in sampling the received signals by the legitimate parties. As discussed in Section 5.4 and from (5.12), the correlation between the signals received by Alice and Bob over the shared channel is not reduced to zero with a non-zero timing error. The correlation coefficient (5.10) was calculated for different timing errors with the static fading channel given in Table 5.1 and an average SNR of 24.5 dB. The results are given in Figure 5.3. A lower correlation coefficient indicates a reduced KGR, as shown in Figure 5.4. This also shows that the KGR is always higher with a dynamic threshold. These results confirm the discussion after (5.12).

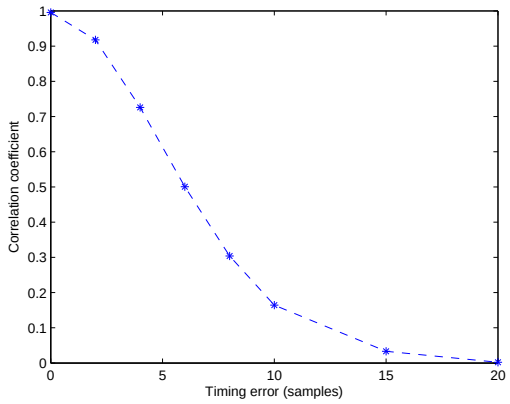


Figure 5.3: Correlation between the signals received by Alice and Bob for a given timing error.

The effect of AWGN on the KGR is presented in Figure 5.5. Figure 5.5-a shows the number of measurements needed to generate 250,000 key bits. As expected, the number of measurements decreases as the SNR increases. At 0 dB, it is not possible to generate key bits, as will be shown later in this section when the KDP is discussed. Figure 5.5-b indicates the effect of the average SNR on the key generation rate. Again a dynamic threshold provides better performance than a fixed threshold, particularly with a high SNR.

5.7.2 Key randomness

The randomness of the key generated is very important from a security perspective. To confirm the randomness, one million key bits were generated using the proposed algorithm with the bits complemented every second measurement. A static fading

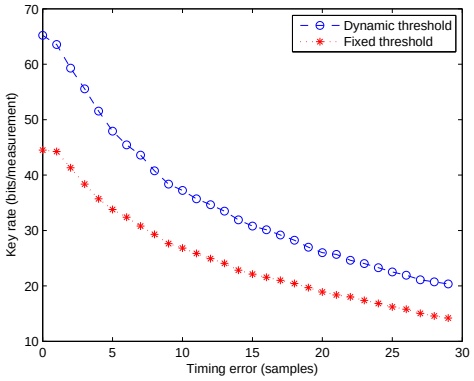
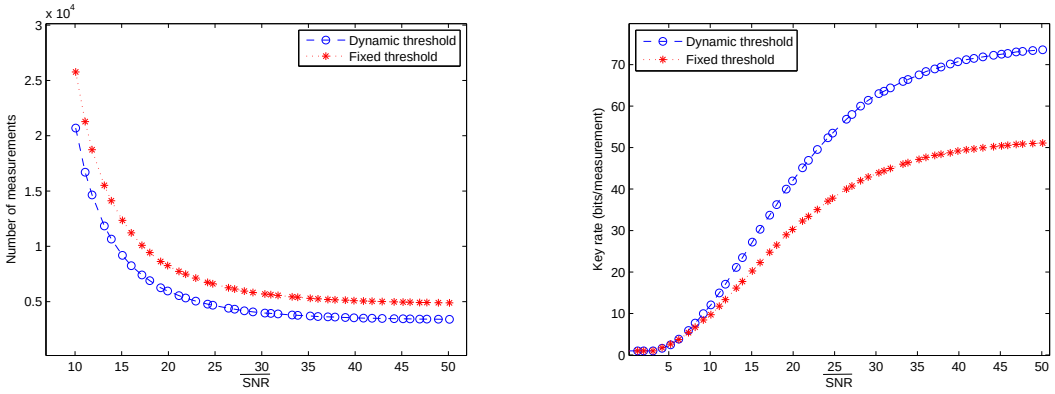


Figure 5.4: The key generation rate (KGR) versus the timing error.

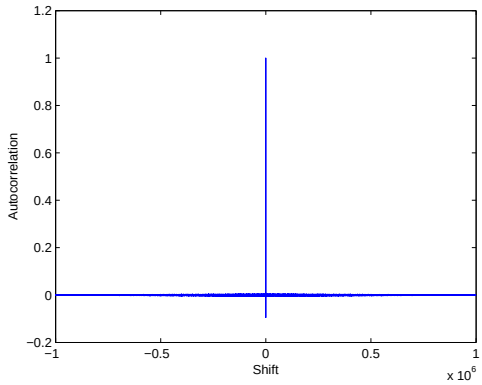


(a) Number of measurements

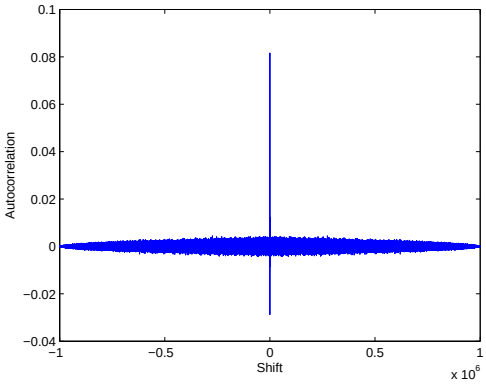
(b) Key generation rate

Figure 5.5: The effect of noise on the proposed algorithm, (a) the number of measurements needed to generate 250,000 key bits, and (b) the key generation rate in bits per measurement.

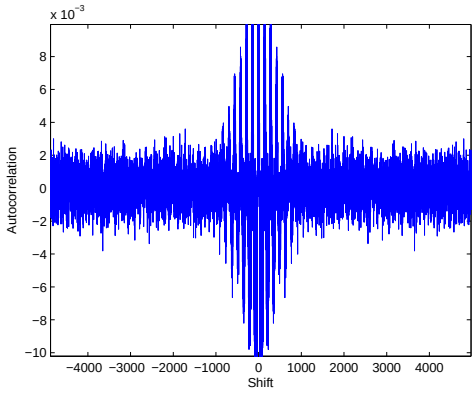
channel can be considered worst case, as a time-varying fading channel has a random effect on the frequency response. The autocorrelation of the key bits was obtained by mapping the bits 0 and 1 to -1 and +1, respectively, and the result is shown in Figure 5.6-a. This indicates that the generated key has a near perfect autocorrelation. Figure 5.6-b shows the sidelobes of the autocorrelation (with the peak value at zero shift removed). The maximum value is 0.08 compared to 1 for the peak at zero shift. Figure 5.6-c shows the sidelobes of the autocorrelation near the centre to illustrate how fast it falls with a non-zero shift. For large shifts, it is within a range of approximately 0.003. Figure 5.7 shows the correlation coefficient for the first 19 sidelobes of the autocorrelation, which is very low after a shift of 4 samples. These results confirm the randomness of the key generated.



(a) Autocorrelation



(b) Sidelobes



(c) Sidelobes near the center

Figure 5.6: The autocorrelation of one million key bits, (a) static fading channel, (b) sidelobes of the autocorrelation, and (c) sidelobes near the center.

5.7.3 Key disagreement probability

The key disagreement probability (KDP) is determined by the similarity of the key bits generated by Alice and Bob, which is affected by timing errors and noise. From Section 5.3, a threshold is used to discard the difference vector elements which have a small magnitude. This reduces the probability of using candidate elements which produce different key bits. Key bit mismatch occurs when different candidate elements are selected by the two parties due to values above and below the threshold. The reconciliation messages in (5.3) are used to agree on the shared sub-blocks. This reduces the probability of key bit mismatch without revealing information about the signs of the shared candidate elements.

Two values A and B are used as reconciliation message masks, with Alice using A_1 and Bob using B_2. Alice recovers Bob's message using B_1, and Bob recovers Alice's

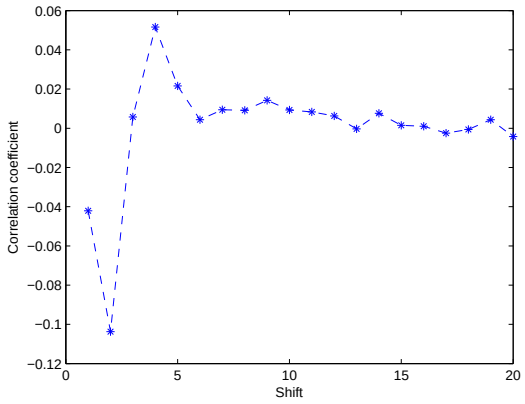


Figure 5.7: The correlation coefficient for the first 19 sidelobes of the autocorrelation.

message using A_2. If the pairs of values (A, B) at Alice and Bob are identical, it is expected that the messages for many of the sub-blocks will match. If the masks are not identical, most of the messages will not match. In this case, a disagreement message is sent to the other party to terminate the algorithm for this measurement. The condition for sending this message is when 90% or more of the sub-blocks do not match. Figure 5.8 shows the KDP and mask disagreement probability (Mask-DP) versus the average SNR. This indicates that the KDP is zero for all SNR values from

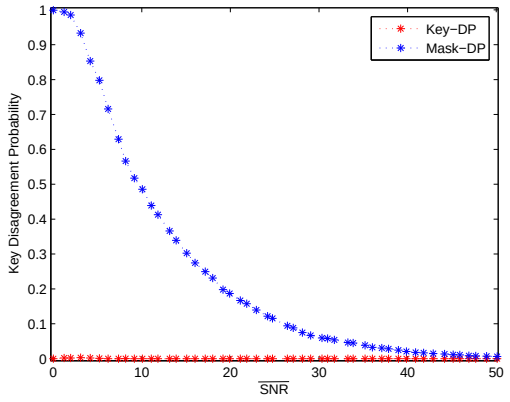


Figure 5.8: Key and mask disagreement probabilities for 10,000 measurements versus the average SNR.

0 to 50 dB. Although the KDP is constant, the KGR changes with SNR. For very low SNRs, the noise results in many variations in the difference vectors, so that the Mask-DP is almost one and nearly all sub-blocks are discarded. This explains why the KGR is almost zero although the KDP is zero. For higher SNRs, the influence of the

noise and the number of mismatches are decreased. At an SNR of 50 dB, the effect of the noise is negligible, and from Figure 5.5-b the KGR is 74 bits/measurement.

5.8 KGR comparison

The key generation rate (KGR) of an algorithm is an important consideration, particularly for secure wireless communication systems. Table 5.4 shows the key generation rate in bits/s (bps) or bits/measurement for different algorithms employing narrowband to UWB fading channels. Both single antenna and MIMO systems are considered that use fading parameters, received signal strength, channel impulse response, or phase measurements for key generation. From this table, the deep fade method [88] has the lowest key rate. This is due to the limited number of deep fades that occur. In addition, randomness is an issue as there is redundancy in the generated key. Compared to the RSS based methods [94, 95, 96], the proposed algorithm has a higher key rate. Further, the techniques which use the channel impulse response have a lower key rate than with the proposed algorithm. Although the key rate in [108] is 95 bits/measurement, the proposed algorithm can generate more bits per second. This is because the channel can be probed at a higher rate with the proposed approach since there is no limitation due to the channel coherence time. In [100, 104], key generation is based on the phase randomness in a fading channel. This technique has a higher key rate than other approaches because the phase in a fading channel typically has a uniform distribution. However, algorithms based on phase randomness are very sensitive to the SNR, which is a serious drawback. The proposed algorithm has a key rate comparable to that in [100], and a similar rate to that in [104] can be achieved if the probing signals are transmitted more frequently. In fact, the performance can be superior depending on the channel conditions. To summarize, the proposed algorithm employs a new approach to generate key bits using measurements in the frequency-domain. Considering randomness, key rate and noise, this is an efficient and robust technique to obtain cryptographic keys for secure wireless communications.

5.9 Security analysis

The uncorrelated effect of the channel on frequency components of the chaotic signal separated by the coherence bandwidth ensures the randomness of the key bits

Table 5.4: Key Generation Rates for Various Algorithms

Algorithm	Fading Channel Type	Measurement Method	Key Generation Rate
[88]	narrowband	deep fade	1.3 bps (25 dB SINR)
[95]	MIMO narrowband	RSSI	10 bps
[96]	ultra-wideband	RSSI	58 bps (35 dB SNR)
[94]	vehicular narrowband	RSSI	1 bps (every 25 measurements)
[87]	narrowband	channel impulse response	1 bps
[97]	ultra-wideband	channel impulse response	17 bits/measurement (30 dB SNR)
[90]	narrowband	channel impulse response	22 bps (DKR 2.2%)
[106]	MIMO narrowband	channel impulse response	30 bits/measurement (4×4 MIMO, 30 dB SNR)
[107]	relay narrowband	channel impulse response	1 bit/measurement (30 dB SNR)
[108]	OFDM narrowband	channel impulse response	95 bits/measurement (256 OFDM slots, 30 dB SNR)
[100]	narrowband	phase of the channel impulse response	104 bps (analytical) (20 dB SNR)
[104]	cooperative narrowband	phase of the channel impulse response	900 bps (25 dB SNR)
Proposed	wideband	channel frequency response	72 bits/measurement (32 dB SNR)

obtained, and also the security. Further, the non-periodicity of the chaotic signal results in uncorrelated key bits even when the channel is static (i.e., regardless of the coherence time of the channel). Masking of the positions of the candidate elements used in the reconciliation messages improves the system security. Transmitting the syndromes of these messages has two advantages. First, it makes it more difficult for Eve to determine the shared sub-blocks between the parties, and second it shortens the length of the reconciliation messages. Here, the reduction in message length is $(196 - 112)/196 = 42.8\%$. The key bits are obtained based on the signs of the values of the candidate elements, but the information reconciliation messages use the positions of these elements in the difference vectors. Thus, no information about the key bits themselves is leaked during this stage, and so privacy amplification is not necessary. The number of key bits agreed upon in each measurement (each execution of the algorithm) varies as shown in Figure 5.9, and will with high probability differ from the number of bits generated by an eavesdropper. This makes it very difficult

for an adversary to predict the key.

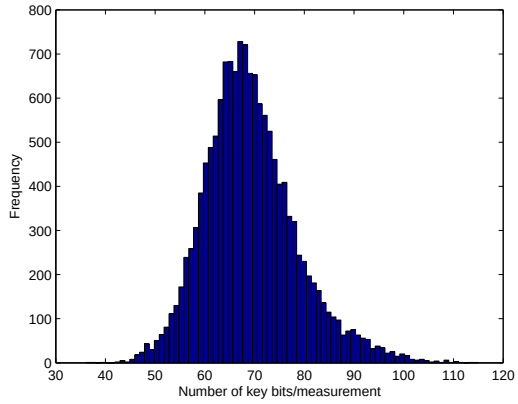


Figure 5.9: Histogram of the number of key bits generated in each measurement.

The security of the algorithm is now examined from the perspective of an eavesdropper (Eve), who receives the transmitted signals from Alice and Bob and the reconciliation messages sent between them, and who also knows the initial values of the chaotic generator (their generator is synchronized with those of Alice and Bob). Ten thousand key bits were generated between Alice and Bob using the static fading channel given in Section 5.5 with an average SNR of 24.5 dB. Eve receives the signals over different 6 taps channels having delays and amplitudes generated randomly from a Rayleigh distribution. Thus Eve receives the transmitted signals over two independent channels. The average SNR for both channels is 33.9 dB. While the correlation coefficient (5.10) between the signals received by Alice and Bob is 0.9997, the correlation coefficient between the signals received by Eve and Alice is only 0.1775. This confirms the small correlation expected from (5.13).

Alice and Bob typically generate the same masks using the positions of the two largest values in their difference vectors. However, these largest values are unlikely to be the same as those at Eve. Thus it will be difficult for Eve to determine the positions of the shared values between Alice and Bob using the received reconciliation messages. Table 5.5 shows how often the positions of the two and five largest values in the difference vectors at Eve match the positions of the two largest values used by Alice and Bob. These results were obtained using 10,000 measurements with the proposed algorithm. This shows that A and B at Alice and Bob rarely occur in the position of the five largest values at Eve, and the probability is almost zero for just the two largest values at Eve. Thus, it is virtually impossible for Eve to determine

the unmasked reconciliation messages.

Table 5.5: Frequency of the Two Mask Values at Alice and Bob Appearing in the Five Largest Values at Eve for 10,000 Measurements

The Largest Value (A)	The Second Largest Value (B)	(A, B) in the Five Largest Values at Eve	(A, B) in the Two Largest Values at Eve
4.74%	6.71%	0.0041%	0.0005%

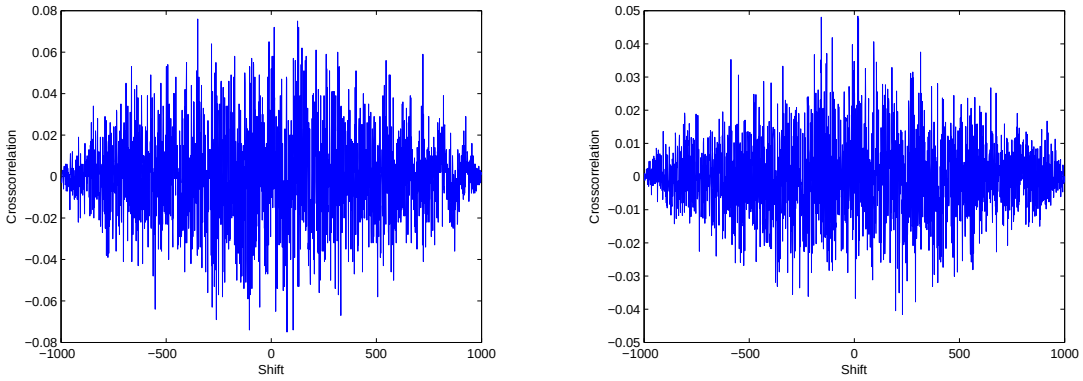
Suppose now that Eve has complete knowledge of the reconciliation messages. Since they do not reveal any information about the signs of the candidate elements, Eve cannot obtain the key using this information and her difference vectors. Table 5.6 gives the average correlation for 100 keys generated at Alice/Bob and Eve, and the average number of mismatched bits between these keys. The correlation was calculated for key lengths of 1000 and 10,000 bits with an average SNR of 24.5 dB. These results show that there is a minimal correlation between the key bits generated. Conversely, there is a perfect match between the keys generated by Alice and Bob. The correlation between the keys generated at Alice/Bob and Eve with different channels is given in Figure 5.10-a, and is similar to the correlation between two 1000 bit random sequences shown in Figure 5.10-b. This low cross-correlation occurs even though the chaotic signals transmitted by Alice, Bob and Eve are identical.

Table 5.6: Correlation of the Keys Generated at Alice/Bob and Eve, and the Percentage of Mismatched Bits

Key Length (bits) (bits)	Average Correlation Coefficient	Average Number of Mismatched Bits	Percentage of Mismatched Bits
1000	0.0284	486	48.6%
10,000	0.0026	5013	50.1%

5.10 Conclusion

In this chapter, a new algorithm to extract a shared cryptographic key was proposed based on the reciprocity of the channel between two parties. The key was obtained using the effect of the channel on the frequency spectrum of a broadband chaotic signal. The performance of the algorithm in the presence of noise and timing errors between the parties was examined in terms of their effect on the key generation rate



(a) Cross-correlation between keys

(b) Cross-correlation of two random sequences

Figure 5.10: (a) The cross-correlation between 1000 bit keys at Alice/Bob and Eve, and (b) the cross-correlation of two uncorrelated 1000 bit random sequences.

and key disagreement probability. An advantage of this algorithm is that it can be employed with static or time-varying fading channels regardless of the channel coherence time due to the use of a non-periodic chaotic probing signal which provides uncorrelated key bits every measurement.

The Lorenz chaotic attractor was used for illustration purposes, but other chaotic attractors which generate a broadband signal can be employed. An investigation of other attractors will be useful in designing chaotic cryptographic key generators for wireless applications. To improve the security, the initial values for the chaotic generator can be exchanged securely, and this is left for future work.

Chapter 6

Secure OFDM with PAPR Reduction using Chaotic Signals

6.1 Introduction

In this chapter, a new physical layer security technique is proposed for orthogonal frequency division multiplexing (OFDM) communication systems. The security is achieved by modifying the OFDM symbols using the frequency-domain phase of a chaotic sequence. In addition, this scheme reduces the peak to average power ratio (PAPR), which is one of the major drawbacks of OFDM. The selected mapping (SLM) technique for PAPR reduction is employed to exploit the random characteristics of chaotic sequences. The PAPR reduction with this algorithm is shown to be similar to that of other SLM schemes, but it has lower computational complexity and side-information does not have to be sent to the receiver. The security of this technique stems from the noise-like behavior of chaotic sequences and their dependence on the initial conditions of the chaotic generator (which are used as the key). Even a slight difference in the initial conditions will result in a different phase sequence, which prevents an eavesdropper from recovering the transmitted OFDM symbols.

6.2 Literature Review

Orthogonal frequency division multiplexing is widely used in wireless communications because of its many advantages. Analog OFDM for multi carrier transmission was proposed in 1966 [115]. The popularity of OFDM is due to the fact that it overcomes

the intersymbol interference (ISI) and intercarrier interference (ICI) problems common with wideband communication systems. This is achieved because the wideband frequency selective channel is divided into parallel frequency flat sub-channels which simplifies the receiver design, particularly in terms of channel equalization. Digital baseband OFDM was proposed in [116] using an inverse fast Fourier transform (IFFT) and a fast Fourier transform (FFT) for modulation and demodulation, respectively. This eliminates the sub-carrier oscillators and coherent demodulators used in analog OFDM, which significantly reduces the cost and computational complexity.

One of the challenging issues with OFDM modulation is the potential for a high peak to average power ratio. This has a negative effect on OFDM performance because the high power amplifier (HPA) is typically non-linear due to efficiency and cost considerations. There are two solutions to this problem. The first is to use an expensive power amplifier with a large linear region (high dynamic range), but this lowers the power efficiency. The other solution is to reduce the PAPR by modifying the signal constellation before performing the IFFT at the transmitter.

The PAPR of an OFDM signal is defined as the ratio of the maximum instantaneous power to the average power

$$PAPR = \frac{\max_{0 \leq n \leq NT}\{|s(n)|^2\}}{E\{|s(n)|^2\}}, \tag{6.1}$$

where $s(n)$ is the complex baseband signal in the time-domain, N is the number of OFDM sub-carriers, and T is the oversampling rate which is typically 4. The PAPR is also evaluated using the crest factor (CF) which is defined as

$$CF = \sqrt{PAPR}. \tag{6.2}$$

The PAPR distribution can be expressed in terms of the complementary cumulative distribution function (CCDF) [117]

$$CCDF = \text{Prob}\{PAPR > z\}. \tag{6.3}$$

With no PAPR reduction, the theoretical CCDF is given by

$$CCDF = 1 - (1 - e^{z^2})^N, \tag{6.4}$$

where $z^2 = |s(n)|^2/\sigma^2$ and σ^2 is the variance of the complex samples $s(n)$ which

represents the average power of the baseband OFDM signals. This shows that as the number of sub-carriers increases, the probability that the peak value exceeds a given value also increases, so the PAPR problem is exacerbated.

6.2.1 Selected Mapping (SLM) Technique

Most PAPR reduction techniques can be divided into two categories, distortion and distortionless. Distortion techniques such as clipping create in-band interference between sub-carriers which affects the orthogonality and creates ISI, and out-of-band interference which produces ICI. Thus, additional processing is required to reduce these undesirable effects. Distortionless techniques such as selected mapping, coding, tone reservation, tone injection, and partial transmit sequence do not introduce ISI or ICI, but complexity can be an issue and side-information typically has to be sent to the receiver, which reduces the overall data rate.

Selected mapping (SLM) is a well-known distortionless technique. With SLM, the input data block is multiplied with M different phase sequences resulting in M OFDM symbols, and the symbol with the smallest PAPR is selected for transmission [118]. Therefore, the performance of this technique depends on the number of phase sequences and their design. Side-information must be sent to inform the receiver which sequence was used, which requires $\lfloor \log_2 M \rfloor$ bits. An algorithm to reduce the side-information was proposed in [119], and an approach which does not require explicit side-information was presented in [120], but the labels employed reduce the data rate. In [121, 122] a comparison of phase sequence generation techniques was given. It was concluded that these sequences should be independent to maximize the PAPR reduction. In [123] a scheme based on exhaustive entropy and chaotic sequences was proposed. A chaotic generator is used to generate U binary phase sequences, and exhaustive entropy is employed to select the M sequences for SLM. However, these techniques are very complex. Several approaches to reduce the complexity of SLM have been developed. In [124], conversion vectors obtained using the IFFT of the phase sequences are employed, while in [125], the IFFT structure was exploited to reduce the number of computations required.

6.3 The Proposed Algorithm

6.3.1 Chaotic phase

Chaotic signals are random-like signals as their autocorrelation is similar to that of a random signal. These signals are unpredictable, and have a broad spectrum in the frequency-domain. Instead of the time-domain representation typically used in the literature, the proposed algorithm uses the phase of the frequency components of the chaotic signal. The objective is to generate phase sequences which can be used to achieve both security and PAPR reduction, and thus these sequences should have random-like characteristics.

The logistic map is a very simple non-linear dynamical equation which exhibits chaotic behavior. It is considered here to generate the chaotic phase sequences, but other generators can be employed. This map is a 1D discrete chaotic generator which has been used extensively in the literature because it can easily be implemented in hardware or software [6]. The dynamics of the logistic map are described by the difference equation

$$x_{n+1} = r x_n (1 - x_n), \tag{6.5}$$

where the state variable x_i is a number between zero and one. The parameter r plays a critical role in the chaotic behavior, and it has been shown that for $3.56995 \leq r < 4$ the dynamics of the map exhibit chaotic behavior. The randomness of the state variable x_i has been studied extensively in the literature [126]. Here, the initial value x_0 and the control parameter r are considered to be the secret key shared between legitimate users. Unlike other chaotic cryptosystems that have been developed for secure communications, the phase of the frequency spectrum of the chaotic sequence x is used in the proposed algorithm.

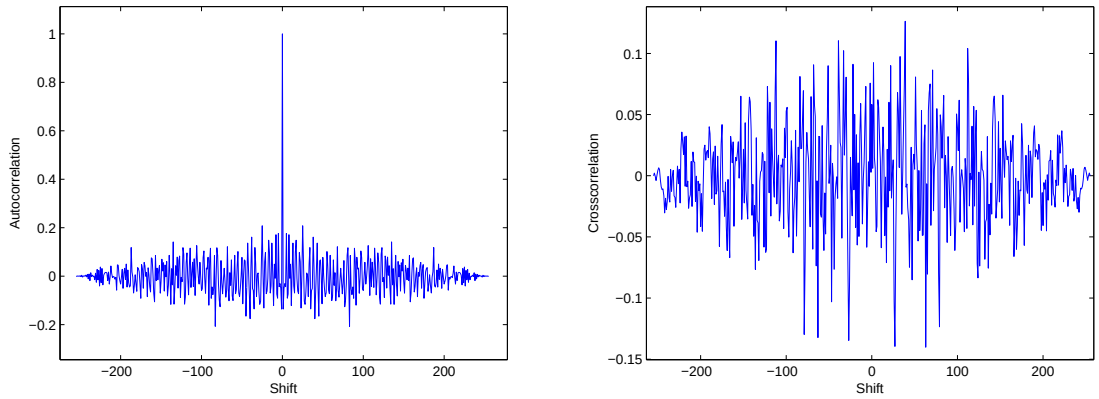
To test the randomness of the chaotic phase sequences obtained, the logistic map was used with $r = 3.9$ and $x_0 = 0.24$ to generate a time-domain sequence of length $N = 256$. A 256-point DFT was used to obtain the corresponding frequency-domain phase sequence. Figure 6.1-a shows the autocorrelation of this sequence, x, and Figure 6.1-b shows the cross-correlation of x and a second sequence y generated using $r = 3.9$ and $x_0 = 0.37$. These results indicate that the chaotic phase sequences have random characteristics. The correlation coefficient for sequences x and y is given by

$$r_{xy} = \frac{C_{xy}}{\sigma_x \sigma_y}, \tag{6.6}$$

where σ_x and σ_y are the corresponding variances and C_{xy} is the autocovariance given by

$$C_{xy} = E\left[\{X - \mu_x\}\{Y - \mu_y\}\right],$$

with means μ_x and μ_y. The correlation coefficient for the phase sequences x and y obtained using the parameters above is 0.0261, which shows that there is minimal correlation between them [127]. Thus, it can be concluded that the phase of a chaotic



(a) The autocorrelation of the logistic map phase (b) The cross-correlation of two different phases

Figure 6.1: (a) The autocorrelation of the logistic map phase sequence for $N = 256$, $r = 3.9$ and $x_0 = 0.24$, and (b) the cross-correlation of this sequence with the corresponding phase sequence for $r = 3.9$ and $x_0 = 0.37$.

signal is a good candidate for use in secure communications applications. As these sequences have random characteristics, it is expected that they can also be used to improve the PAPR.

6.3.2 Chaotic SLM

As stated in the previous section, with the SLM technique M different sequences are created from the data to be transmitted, and the one with the smallest PAPR is selected for transmission. The generation of these sequences increases the computational complexity at the transmitter. The proposed algorithm benefits from the random characteristics of chaotic phase sequences to produce the M sequences with low complexity. The chaotic generator is used to obtain a phase sequence of length N, and using cyclic shifts, M versions of this sequence are obtained. The length of

the cyclic shift L is given by

$$L = \lfloor N/M \rfloor, \tag{6.7}$$

where $\lfloor z \rfloor$ denotes the largest integer less than or equal to z. The M sequences are used to modify the phase of the OFDM data, and the sequence with the smallest PAPR is selected for transmission. Figure 6.2 shows the PAPR performance using 16-QAM modulation with $N = 128$ and $M = 4$, 6 and 8, along with the performance with no PAPR reduction. Figure 6.3 shows the PAPR performance for QPSK with $N = 64$ and $M = 4$, 6 and 8, along with the theoretical performance.

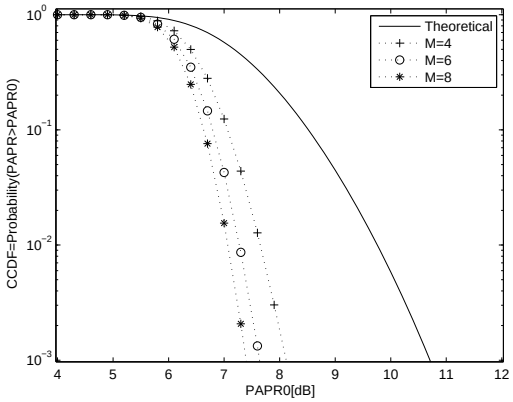


Figure 6.2: PAPR reduction for 16-QAM modulation with $N = 128$ and different numbers of chaotic SLM sequences.

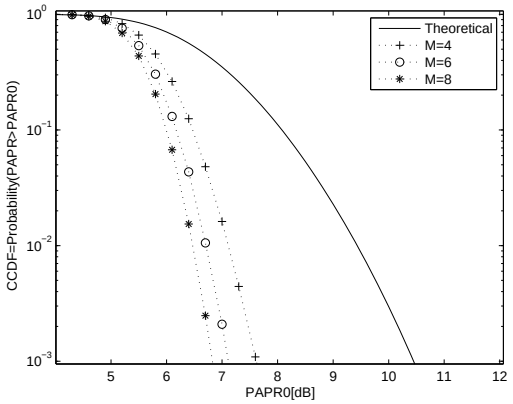


Figure 6.3: PAPR reduction for QPSK modulation with $N = 64$ and different numbers of chaotic SLM sequences.

6.3.3 Quantization of the chaotic phase sequences

Since the phase of the chaotic signal is continuous, the complexity of combining the corresponding phase sequences with the OFDM symbols is high. To reduce this complexity, the phase space is partitioned into a number of regions and the phase in each region is assigned a fixed value. Then a lookup table can be used instead of multiplications. For example, if the OFDM data constellation has S points and the number of phase regions is K, the look-up table will have size $S \times K$. Figure 6.4 shows the PAPR performance with QPSK modulation, $N = 64$ and $M = 8$ chaotic SLM sequences using $K = 8, 16$ and 32 phase quantization regions. Comparing this figure with the corresponding results in Figure 6.3 indicates that the quantization has a minimal effect on the performance.

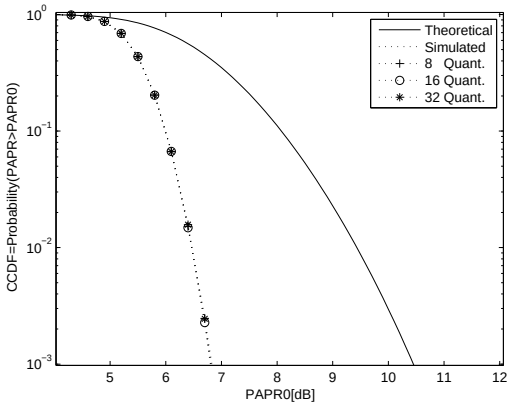


Figure 6.4: PAPR reduction using QPSK modulation with $N = 64$ and 8 chaotic SLM sequences with $K = 8, 16$ and 32 regions.

The number of regions K has an impact on the security of the algorithm, as a greater number of regions increases the complexity, making it harder for an attacker to determine the OFDM symbols. However, this also increases the size of the look-up table. As the PAPR performance is not greatly affected by the value of K, it can be chosen based on the available computational resources and required security.

The proposed system is now compared with several SLM techniques in the literature. Figure 6.5 presents the PAPR performance of 16-QAM with $N = 256$ and $M = 8, 10, 16$ and 32. For $M = 32$ the proposed algorithm is slightly better than the approach in [128], and for $M = 16$ is similar to the results in [129]. Compared with the technique in [130], with $M = 10$ the proposed solution is 0.7 dB better at CCDF $= 10^{-3}$. For $M = 8$, it is better than the results in [131], and is much better

than those in [132]. Note that the proposed algorithm is superior from the security perspective, and has low computational complexity.

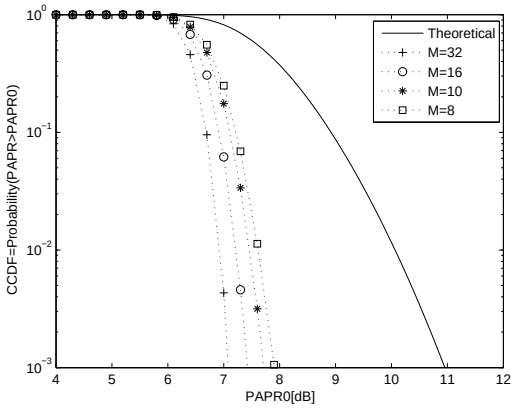


Figure 6.5: The proposed chaotic SLM compared with SLM techniques in the literature using 16-QAM with $N = 256$ and $M = 8$, 10, 16 and 32.

6.3.4 Data recovery without side-information

With most SLM PAPR reduction techniques, the original data are recovered at the receiver using side-information sent from the transmitter to identify the sequence that was used. However, the phase sequence employed can be predicted at the receiver using the proposed algorithm without side-information. This is based on the fact that the transmitter and receiver have the same phase sequences from the chaotic generator. The Euclidean distance between the recovered symbols q_i^j using the jth shifted phase sequence and the OFDM data constellation p_i is

$$d(j) = \sqrt{\sum_{i=1}^{N} \left(p_i - q_i^j\right)^2}, j = 1, \ldots, M. \tag{6.8}$$

These distances are calculated at the receiver to predict the chosen phase sequence, as described below.

At the receiver, the same chaotic phase sequence as at the transmitter is generated based on the shared key between them, so it has the same M shifted sequences. The received signal after the IFFT is combined with these sequences to remove the effect of the phase sequences. Then the Euclidean distance (6.8) is calculated for the resulting sequences, and the one corresponding to the lowest value of $d(j)$ is considered as

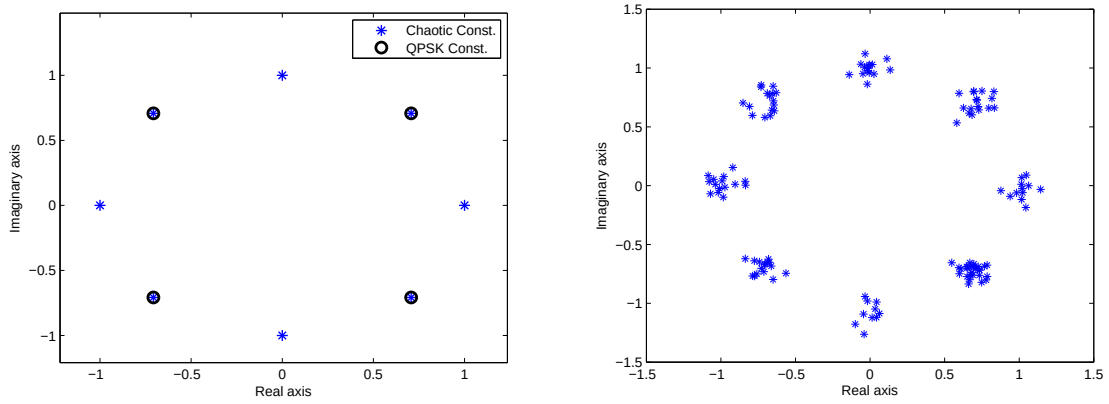
the original data. This approach requires that the chaotic phase constellation differ from the constellation of the data modulation, which is typically PSK or QAM. Thus, the transmitted OFDM symbol has a constellation which differs from the data constellation. At the receiver, if the received OFDM symbol is modified with the shifted phase sequence used by the transmitter, the resulting constellation will be the data constellation. However, if an incorrect phase sequence is used at the receiver, the resulting sequence will still be a combination of the phase sequence and data constellations. Thus, the correct phase sequence should result in the lowest Euclidean distance. Note that the chaotic phase constellation points have unit magnitude, so that although the transmitted symbols are from a new constellation, the magnitudes are unchanged, so the proposed algorithm does not increase the average transmit power as with other PAPR reduction approaches.

6.3.5 Performance Results

In this section, an OFDM system with $N = 128$ is considered with QPSK and 16-QAM modulation. The number of SLM sequences is $M = 8$, and these are obtained using the logistic map as the chaotic generator. The initial conditions and control parameter are identical at the transmitter and receiver (the legitimate users), and are $x_0 = 0.24$ and $r = 3.9$.

We first consider an AWGN channel with SNR = 20 dB and $K = 8$ phase sequence constellation points. Figure 6.6-a presents the quantized chaotic phase sequence and QPSK constellations, and Figure 6.6-b the constellation of the corresponding received OFDM symbol. The results of reconstructing the OFDM data using the 8 phase sequences are given in Figure 6.7. This shows that chaotic phase sequence 5 is the most likely one employed at the transmitter. The corresponding Euclidean distance calculations are given in the first row of Table 6.1. As expected, the Euclidean distance for sequence 5 is the lowest, so the receiver selects this sequence to recover the data (which is correct). Figure 6.8 presents the constellations for 16-QAM with $K = 8$, and Figure 6.9 shows the results of removing the effects of the phase sequences on the received OFDM symbol. In this case, sequence 2 appears to be correct (which is indeed the case), and this is confirmed by the Euclidean distances in the second row of Table 6.1.

Since the proposed algorithm makes a decision based on the Euclidean distance without using side-information, the symbol error rate (SER) performance will be



(a) Chaotic and QPSK constellation (b) Received OFDM symbol constellation

Figure 6.6: (a) The constellations of the quantized chaotic phase sequences and QPSK, and (b) the received OFDM symbol constellation.

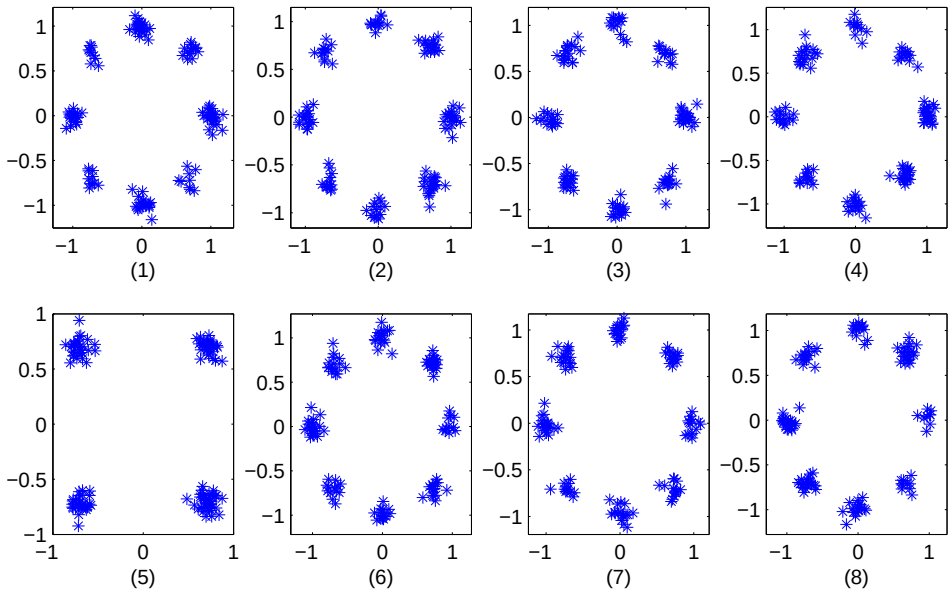
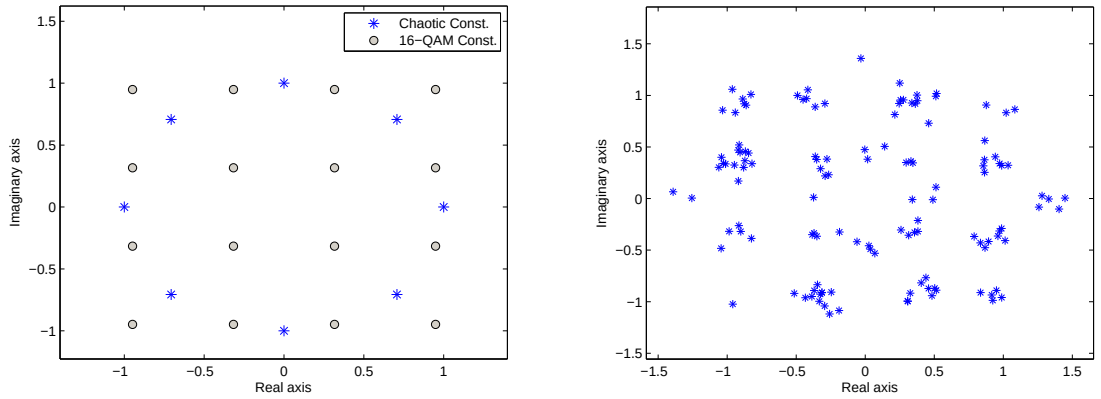


Figure 6.7: The constellations for the 8 recovered OFDM symbols with QPSK modulation.

affected. Thus, the SER was obtained using QPSK modulation with and without side-information (but without considering the loss in data rate in the former case). Figure 6.10 presents the results for $N = 32$ and 64 over an AWGN channel. This shows that the performance is identical for high SNRs. Further, increasing the length of the OFDM symbols improves the performance, as the difference is insignificant for an SNR

Table 6.1: The Euclidean Distances for the $M = 8$ SLM Chaotic Phase Sequences

Sequence no.	1	2	3	4	5	6	7	8
QPSK	6.66	5.67	6.02	5.87	1.10	5.77	5.90	5.66
16-QAM	2.68	1.13	2.43	2.72	2.57	2.92	2.54	2.70



(a) Chaotic and 16-QAM constellation

(b) Received OFDM symbol constellation

Figure 6.8: (a) The constellations of the quantized chaotic phase sequences and 16-QAM, and (b) the received OFDM symbol constellation.

of 6 dB with $N = 64$, but for $N = 32$ the SNR should be greater than 9 dB. Given that in practical OFDM systems N is typically 256 or larger, the loss in performance due to not using side-information is negligible. Further, this eliminates the loss in data rate due to transmitting this information. The corresponding performance over a Rayleigh fading channel is shown in Figure 6.11. This indicates that the fading channel performance is very similar to that in an AWGN channel in terms of the SNR values where the SER with side information is the same as the SER without side information. Thus fading does not affect the reliability of the proposed algorithm.

6.4 Security Analysis

In the literature, the outputs of chaotic generators in the time-domain are used to obtain random sequences of zeros and ones. These random sequences are used to encrypt the data either directly via modulo 2 addition, or indirectly as a key for a traditional cryptographic algorithm. The security depends on the control parameters and initial values of the chaotic generator, which constitute the shared key between legitimate

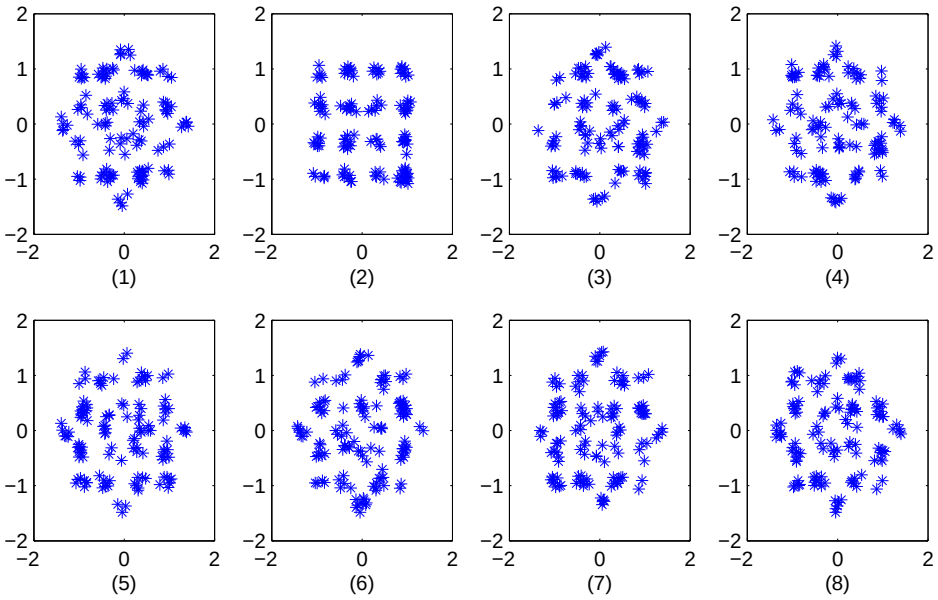
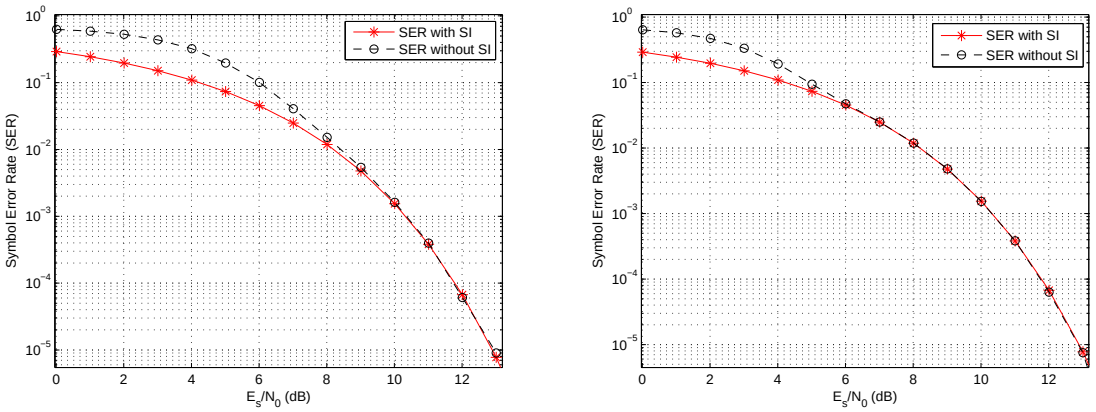


Figure 6.9: The constellations of the 8 recovered OFDM symbols with 16-QAM modulation.



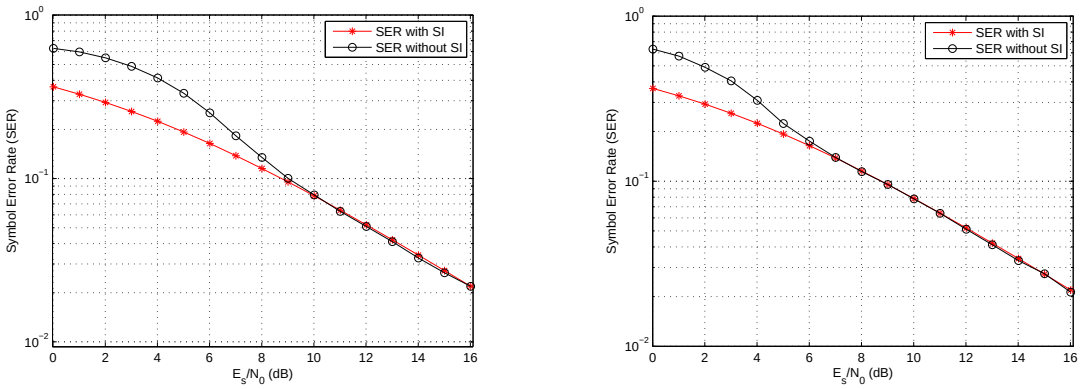
(a) SER of 32-OFDM

(b) SER of 64-OFDM

Figure 6.10: The symbol error rate (SER) with QPSK modulation using length (a) $N = 32$, and (b) $N = 64$ OFDM symbols over an AWGN channel.

users. The complexity of the chaotic system makes it hard for an eavesdropper to predict the key. However, some cryptanalysis methods exploit the synchronization of the chaotic generator with the transmitted signal, and some systems can be broken even when the control parameters and initial values are not known exactly.

Conversely, the proposed algorithm uses the phase of the frequency response of



(a) SER of 32-OFDM

(b) SER of 64-OFDM

Figure 6.11: The symbol error rate (SER) with QPSK modulation using length (a) $N = 32$, and (b) $N = 64$ OFDM symbols over a Rayleigh fading channel.

the chaotic signal. As shown in Section 6.3.1, the phase sequences of two different chaotic signals are uncorrelated, so the phase sequences for two parties having different keys will also be uncorrelated. Consider the transmitted 16-QAM OFDM symbol in Section 6.3.5 and an eavesdropper who has initial values $r = 3.9$ and $x_0 = 0.67$. Figure 6.12 shows the constellations of the recovered 16-QAM OFDM symbols with $K = 8$ at the eavesdropper. The similarity between these constellations makes it impossible to predict the transmitted OFDM symbol. This is confirmed by the corresponding Euclidean distances in Table 6.2, which are very close.

Note that quantizing the chaotic phase sequences prevents an eavesdropper from predicting the actual phases. Further, transmitting side-information as is common with PAPR reduction techniques in the literature can compromise the security of the system, but the proposed approach does not require that this information be sent. This prevents an eavesdropper from obtaining information which can assist in predicting the OFDM data. Finally, the non-periodicity of the chaotic signal guarantees that the phase sequences differ for each OFDM symbol.

The control parameters and the initial values of the chaotic generator form the key shared between legitimate users. The key length is a function of the dimensionality of the chaotic generator, the number of control parameters, and the precision of the hardware or software in use. Since the chaotic generator is described by difference equations and each equation has an initial value, higher dimensional generators will have a longer key length. Chaotic systems are sensitive to very small signal deviations, so that once the sensitivity threshold of the hardware or the software is reached, the

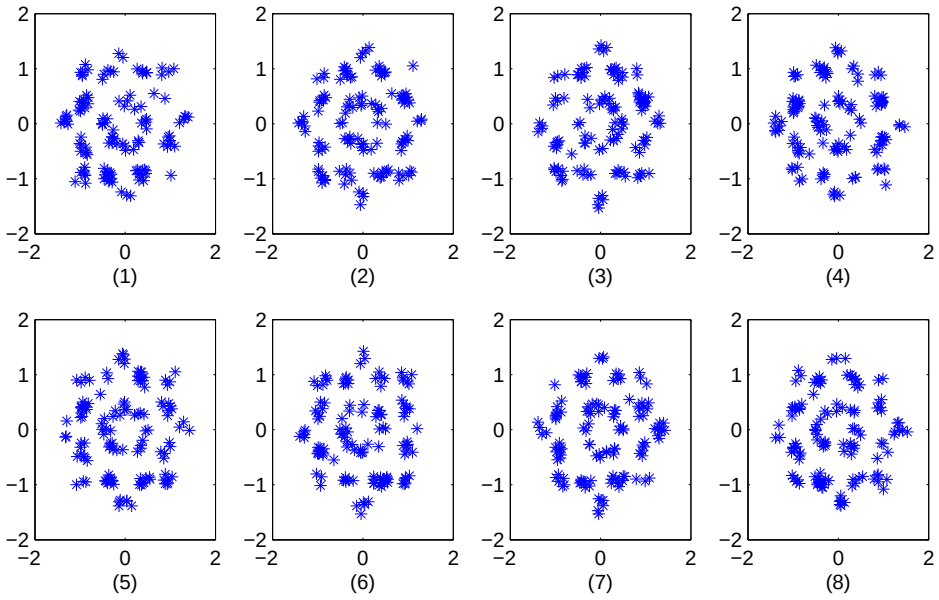


Figure 6.12: The 16-QAM constellations obtained by an eavesdropper with initial conditions different than at the transmitter.

orbit of the chaotic system will be affected. In [75], it was shown that a Lorenz generator implemented using double-precision floating-point has a key space of 10^{39}. This key length is greater than 2^{100} and so provides significant robustness against a brute-force attack [60]. On the other hand, using a high-dimensional chaotic generator means more computations per key bit. As a result, the chaotic generator should be chosen according to the required level of security and the available computational resources. In this chapter, the logistic map is used for illustration purposes.

Table 6.2: The Euclidean Distances for the $M = 8$ SLM Sequences at an Eavesdropper

Sequence no.	1	2	3	4	5	6	7	8
16-QAM	2.48	2.55	2.64	2.70	2.44	2.44	2.73	2.54

6.5 Conclusion

In this chapter, a secure OFDM communication system has been proposed. The security is based on the randomness of chaotic phase sequences which are combined with the OFDM symbols. Further, the selected mapping technique is employed with these

sequences to achieve a reduction in the peak-to-average power ratio. A single chaotic sequence is used to generate all of the SLM sequences for a given symbol using cyclic shifts so that the complexity is low compared to other techniques in the literature. It was shown that quantizing the chaotic phase sequence has a minimal effect on the performance, but significantly reduces the computational complexity. The non-periodicity of the chaotic signal guarantees a unique phase sequence for every OFDM symbol. The key length is determined by the chaotic generator and the hardware or software use. A higher dimensional generator will have a longer key length. In addition, a technique was presented to eliminate the need to send side-information to the receiver to recover the transmitted data, which improves the bandwidth efficiency and security compared to other approaches. Results were presented which show that the proposed technique has PAPR performance which is comparable to that with other solutions, but it also provides secure communications.

Chapter 7

Conclusions

The work presented in this dissertation concerns the topic of secure communications based on chaotic systems. Although, chaos is considered an ideal candidate for use in secure communications, its sensitivity to channel noise for physical layer transmission, or errors due to finite precision arithmetic of the numerical algorithms in higher layers such as presentation and data link layers limits the use of chaotic cryptosystems and restrict the practical implementations. Starting from the obstacles which face utilizing chaos in communications, this dissertation provides contributions to make chaotic cryptosystems in communications dependable and secure.

The dissertation presents a new class of 3D discrete chaotic systems developed from 3D continuous chaotic systems. It provides accurate chaotic values, which makes it suitable for use in secure digital communication systems. Based on this new class, new ciphers for symmetric key encryption were developed providing security in the higher layers. The developed ciphers provide good security level, besides simple hardware implementation and comparable speed.

With regard to wireless communications, this dissertation considered chaos systems from a different perspective. Instead of the time-domain representation of the signals, the characteristics in the frequency-domain (either the magnitude or phase) were used to provide secure communications in the physical layer. The first algorithm exploits the reciprocity of the fading channel to provide secure communication between two legitimate users. The frequency characteristics of a broadband chaotic signal were used to generate the key. It was shown that it is superior to other types of signal characteristics used in the literature in terms of accuracy and robustness to noise. The second algorithm provides security for OFDM symbols using the phase response of the chaotic signal. The randomness of the chaotic phase signals was ex-

ploited to reduce the peak to average power ration of the OFDM signals without affecting on the spectrum efficiency.

Several perspective investigations can be performed in the future.

1. Since this dissertation is the first to consider the frequency domain characteristics of the chaotic signal rather than the time domain characteristics, research can be done focusing on other representations of the chaotic signal such as time-frequency transformations (Wavelet transform, Short-time Fourier transform, Wigner distribution). This is promising field to further the use of chaos in communication systems.

2. As the world fully relies on wireless communications, the demand for high data rate transmission is increasing. The signals carrying information through wireless channels are strongly modified due to characteristics such as noise and multipath. This prevents information from being transmitted at high bit rates. New studies in chaos-based communications have shown that although the chaotic signal is strongly modified by the wireless channel, the dynamics of both the transmitted and received signals for some chaotic generators is identical. This needs more investigation as the dynamics may be vulnerable to an eavesdropper.

Bibliography

[1] E. Lorenz. Deterministic nonperiodic flow. *J. Atmos. Sci.*, 20(2):130–141, 1963.

[2] Y. Shimada. Detecting stretch-and-fold mechanism in chaotic dynamics. *Int. J. Bifurcation Chaos*, 22(11):1–13, 2012.

[3] J. Sprott. *Chaos and Time-series Analysis*. Oxford, UK: Oxford Univ. Press, 2003.

[4] L. Pivka, C. Wu, and A. Huang. Chua's oscillator: A compendium of chaotic phenomena. *J. Franklin Inst.*, 331(6):705–741, 1994.

[5] O. Rossler. An equation for continuous chaos. *Phys. Lett. A*, 57(5):397-398, 1976.

[6] R. May. Simple mathematical models with very complicated dynamics. *Nature*, 261(5560):459–467, 1976.

[7] J. Henon. A two-dimensional mapping with a strange attractor. *Commun. Math. Phys.*, 50(1):69–77, 1976.

[8] L. Pecora and T. Carroll. Driving systems with chaotic signals. *Phys. Rev. A*, 44(4):2374–2383, 1991.

[9] H. Nijmeijer and I. Mareels. An observer looks at synchronization. *IEEE Trans. Circuits Syst. I*, 44(10):882–890, 1997.

[10] M. Yassen. Controlling chaos and synchronization for new chaotic system using linear feedback control. *Chaos, Solitons Fractals*, 26(3):913-920, 2005.

[11] J. Effa, B. Essimbi, and J. Ngundam. Synchronization of improved chaotic Colpitts oscillators using nonlinear feedback control. *Nonlinear Dyn.*, 58(1):39–47, 2009.

[12] X. Zhou, Y. Wu, Y. Li, and H. Xue. Adaptive control and synchronization of a novel hyperchaotic system with uncertain parameters. *Appl. Math. Comput.*, 203(1):80-85, 2008.

[13] J. Laoye, U. Vincent, and S. Kareem. Chaos control of 4D chaotic systems using recursive backstepping nonlinear controller. *Chaos, Solitons Fractals*, 39(1):356-362, 2009.

[14] M. Naseh and M. Haeri. Robustness and robust stability of the active sliding mode synchronization. *Chaos, Solitons Fractals*, 39(1):196-203, 2009.

[15] R. Mainieri and J. Rehacek. Projective synchronization in three-dimensional chaotic systems. *Phys. Rev. Lett.*, 82(15):3042–3045, 1999.

[16] M. El-Dessoky. Synchronization and anti-synchronization of a hyperchaotic chen system. *Chaos, Solitons Fractals*, 39(4):1790–1797, 2009.

[17] Q. Wang and Y. Chen. Generalized Q-S (lag, anticipated and complete) synchronization in modified Chua's circuit and Hindmarsh-Rose systems. *Appl. Math. Comput.*, 181(1):48–56, 2006.

[18] G. Kolumban, M. Kennedy, and L. Chua. The role of synchronization in digital communications using chaos. I. Fundamentals of digital communications. *IEEE Trans. Circuits Syst. I*, 44(10):927–936, 1997.

[19] T. Yang and L. Chua. Chaotic digital code division multiple access (CDMA) communications. *Int. J. Bifurcation Chaos*, 7(12):2789–2805, 1997.

[20] H. Dedieu, M. Kennedy, and M. Hasler. Chaos shift keying: Modulation and demodulation of a chaotic carrier using self-synchronizing Chua's circuits. *IEEE Trans. Circuits Syst. II*, 40(10):634–642, 1993.

[21] I. Marino, L. Lopez, J. Miguez, and M. Sanjuan. A novel channel coding scheme based on continuous-time chaotic dynamics. *Proc. Int Conf. Digit Signal Process*, Santorini, Greece, 2002, 1321–1324.

[22] C.E. Shannon. Communication theory of secrecy systems. *Bell system technical journal*, 28(4):656–715, 1949.

[23] V. Anishchenko, T. Vadivasova, G. Okrokvertskhov, and G. Strelkova. Correlation analysis of dynamical chaos. *Physica A*, 325(1):199–212, 2003.

[24] G. lvarez and S. Li. Some basic cryptographic requirements for chaos-based cryptosystems. *Int. J. Bifurcation Chaos*, 16(8):2129–2151, 2006.

[25] L. Kocarev, K. Halle, K. Eckert, L. Chua, and U. Parlitz. Experimental demonstration of secure communications via chaotic synchronization. *Int. J. Bifurcation Chaos*, 2(3):709–713, 1992.

[26] G. lvarez, S. Li, F. Montoya, G. Pastor, and M. Romera. Breaking projective chaos synchronization secure communication using filtering and generalized synchronization. *Chaos, Solitons Fractals*, 24(3):775–783, 2005.

[27] T. Yang, L. Yang, and C. Yang. Breaking chaotic switching using generalized synchronization: examples. *IEEE Trans. Circuits Syst. I*, 45(10):1062–1067, 1998.

[28] T. Yang and L. Chua. Secure communication via chaotic parameter modulation. *IEEE Trans. Circuits Syst. I*, 43(9):817–819, 1996.

[29] M. Sobhy and A. Shehata. Secure computer communication using chaotic algorithms. *Int. J. Bifurcation Chaos*, 10(12):2831–2839, 2000.

[30] G. lvarez, F. Montoya, M. Romera, and G. Pastor. Breaking parameter modulated chaotic secure communication system. *Chaos, Solitons Fractals*, 21(4):783–787, 2004.

[31] T. Yang, C. Wu, and L. Chua. Cryptography based on chaotic systems. *IEEE Trans. Circuits Syst. I*, 44(5):469–472, 1997.

[32] X. Yong, W. Hua, L. Yongge, and P. Bin. Image encryption based on synchronization of fractional chaotic systems. *Commun. Nonlinear Sci. Numer. Simul.*, 19(10):3735–3744, 2014.

[33] J. Chen, Z. Zhu, and H. Yu. A fast chaos-based symmetric image cryptosystem with an improved diffusion scheme. *Optik - Int. J. Light Electron Opt.*, 125(11):2472–2478, 2014.

[34] D. Wheeler. Problems with chaotic cryptosystems. *Cryptologia*, 13(3):243–250, 1989.

[35] G. Vanwiggeren and R. Roy. Chaotic communication using time-delayed optical systems. *Int. J. Bifurcation Chaos*, 9(11):2129–2156, 1999.

[36] T. Carroll and L. Pecora. Cascading synchronized chaotic systems. *Physica D*, 67(1):126–140, 1993.

[37] A. Elwakil and M. Kennedy. Improved implementation of Chua's chaotic oscillator using current feedback op amp. *IEEE Trans. Circuits Syst.*, 47(1):76–79, 2000.

[38] J. Corron, R. Reed, N. Blakely, K. Myneni, and D. Pethel. Chaotic scrambling for wireless analog video. *Commun. Nonlinear Sci. Numer. Simul.*, 15(9):2504–2513, 2010.

[39] L. Kocarev and S. Lian. *Chaos-Based Cryptography Theory, Algorithms and Applications*. Heidelberg, Germany: Springer Berlin Heidelberg, 2011.

[40] M. Azzaz, C. Tanougast, S. Sadoudi, and A. Dandache. Robust chaotic key stream generator for real-time images encryption. *J. Real Time Image Process.*, 8(3):297–306, 2013.

[41] J. Cartwright and O. Piro. The dynamics of Runge-Kutta methods. *Int. J. Bifurcation Chaos*, 2(3):427–449, 1992.

[42] J. Butcher. A history of Runge-Kutta methods. *Appl. Numer. Math.*, 20(3):247-260, 1996.

[43] S. Sadoudi, C. Tanougast, A. Dandache, and M. Salah. Design and FPGA implementation of a wireless hyperchaotic communication system for secure real-time image transmission. *EURASIP J. Image Video Process.*, 2013(1), 2013.

[44] X. Zeng, R. Eykholt, and R. Pielke. Estimating the Lyapunov exponent spectrum from short time series of low precision. *Phys. Rev. Lett.*, 66(25):3229–3232, 1991.

[45] F. Christiansen and H. Rugh. Computing Lyapunov spectra with continuous Gram-Schmidt orthonormalization. *Nonlinearity*, 10:1063–1072, 1997.

[46] R. Brown and P. Bryant. Computing the Lyapunov spectrum of a dynamical system from an observed time series. *Phys. Rev. A*, 43(6):2787–2806, 1991.

[47] J. Roux, R. Simoyi, and H. Swinney. Observation of a strange attractor. *Physica D*, 8(1):257-266, 1983.

[48] A. Wolf, J. Swift, H. Swinney, and J. Vastano. Determining Lyapunov exponents from a time series. *Physica D*, 16(3):285-317, 1985.

[49] M. Rafikov and J. Balthazar. On control and synchronization in chaotic and hyperchaotic systems via linear feedback control. *Commun. Nonlinear Sci. Numer. Simul.*, 13(7):1246-1255, 2008.

[50] D. Wheeler. Supercomputer investigations of a chaotic encryption algorithm. *Cryptologia*, 15(2):140–152, 1991.

[51] F. Pichler and J.Scharinger. Finite dimensional generalized Baker dynamical systems for cryptographic applications. *Lecture Notes in Computer Science, Computer Aided Systems Theory EUROCAST '95*, 1030:465–476, 1996.

[52] M. Khan, T. Shah, H. Mahmood, and M. Gondal. An efficient method for the construction of block cipher with multi-chaotic systems. *Nonlinear Dyn.*, 71(3):489–492, 2013.

[53] A. Akhshani, A. Akhavan, A. Mobaraki, S. Lim, and Z. Hassan. Pseudo random number generator based on quantum chaotic map. *Commun. Nonlinear Sci. Numer. Simul.*, 19(1):101-111, 2014.

[54] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Trans. Inform. Theory*, 22(6):644–654, 1976.

[55] L. Kocarev and Z. Tasev. Public-key encryption based on Chebyshev maps. *Proc. IEEE Int. Symp. Circuits Syst.*, Bangkok, Thailand, 2003, 28–31.

[56] A. Akhavan, A. Samsudin, and A. Akhshani. A novel parallel hash function based on 3D chaotic map. *EURASIP J. on Adv. Signal Process.*, 2013(1), 2013.

[57] Y. Li, D. Xiao, H. Li, and S. Deng. Parallel chaotic hash function construction based on cellular neural network. *Neural Comput. Appl.*, 21(7):1563–1573, 2012.

[58] X. Wang and K. Guo. A new image alternate encryption algorithm based on chaotic map. *Nonlinear Dyn.*, 76(4):1943–1950, 2014.

[59] F. Eyebe and J. Armand. A fast chaotic block cipher for image encryption. *Commun. Nonlinear Sci. Numer. Simul.*, 19(3):578-588, 2014.

[60] B. Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. 2nd ed. New York, USA: Wiley, 1996.

[61] T. Yang, L. Yang, and C. Yang. Cryptanalyzing chaotic secure communications using return maps. *Phys. Lett. A*, 245(6):495-510, 1998.

[62] X. Wu, H. Hu, and B. Zhang. Analyzing and improving a chaotic encryption method. *Chaos, Solitons Fractals*, 22(2):367-373, 2004.

[63] S. Li, G. lvarez, and G. Chen. Breaking a chaos-based secure communication scheme designed by an improved modulation method. *Chaos, Solitons Fractals*, 25(1):109–120, 2005.

[64] A. Orue, G. lvarez, G. Pastor, M. Romera, F. Montoya, and S. Li. A new parameter determination method for some double-scroll chaotic systems and its applications to chaotic cryptanalysis. *Commun. Nonlinear Sci. Numer. Simul.*, 15(11):3471-3483, 2010.

[65] T. Stojanovski, L. Kocarev, and U. Parlitz. A simple method to reveal the parameters of the Lorenz system. *Int. J. Bifurcation Chaos*, 6(12):2645–2652, 1996.

[66] E. Solak. Partial identification of Lorenz system and its application to key space reduction of chaotic cryptosystems. *IEEE Trans. Circuits Syst. II*, 51(10):557–560, 2004.

[67] R. Rhouma, S. Meherzi, and S. Belghith. OCML-based colour image encryption. *Chaos, Solitons Fractals*, 40(1):309-318, 2009.

[68] H. Liu and X. Wang. Color image encryption based on one-time keys and robust chaotic maps. *Comput. Math. Appl.*, 59(10):3320-3327, 2010.

[69] X. Wang, J. Zhao, and H. Liu. A new image encryption algorithm based on chaos. *Opt. Commun.*, 285(5):562-566, 2012.

[70] A. Abd El-Latif and X. Niu. A hybrid chaotic system and cyclic elliptic curve for image encryption. *AEU Int. J. Electron. Commun.*, 67(2):136-143, 2013.

[71] V. Patidar, N. Pareek, G. Purohit, and K. Sud. A robust and secure chaotic standard map based pseudorandom permutation-substitution scheme for image encryption. *Opt. Commun.*, 284(19):4331-4339, 2011.

[72] S. Sayedzadeh and S. Mirzakuchaki. A fast color image encryption algorithm based on coupled two-dimensional piecewise chaotic map. *Signal Process.*, 92(5):1202-1215, 2012.

[73] G. Chen, Y. Mao, and C. Chui. A symmetric image encryption scheme based on 3D chaotic cat maps. *Chaos, Solitons Fractals*, 21(3):749-761, 2004.

[74] A. Kanso and M. Ghebleh. A novel image encryption algorithm based on a 3D chaotic map. *Commun. Nonlinear Sci. Numer. Simul.*, 17(7):2943-2959, 2012.

[75] M.F. Haroun and T.A. Gulliver. A new 3D chaotic cipher for encrypting two data streams simultaneously. *Nonlinear Dyn.*, 81(3):1053-1066, 2015.

[76] A. Orue, V. Fernandez, G. lvarez, G. Pastor, M. Romera, S. Li, and F. Montoya. Determination of the parameters for a Lorenz system and application to break the security of two-channel chaotic cryptosystems. *Phys. Lett. A*, 372(34):5588-5592, 2008.

[77] X. Zhang, Y. Mao, and Z. Zhao. An efficient chaotic image encryption based on alternate circular S-boxes. *Nonlinear Dyn.*, 78(1):359–369, 2014.

[78] Y. Zhou, L. Bao, and C. Chen. A new 1D chaotic system for image encryption. *Signal Process.*, 97:172–182, 2014.

[79] X. Huang and G. Ye. An efficient self-adaptive model for chaotic image encryption algorithm. *Commun. Nonlinear Sci. Numer. Simul.*, 19(12):4094–4104, 2014.

[80] M. Ghebleh, A. Kanso, and H. Noura. An image encryption scheme based on irregularly decimated chaotic maps. *Signal Process. Image Commun.*, 29(5):618–627, 2014.

[81] L. Zhang, X. Hu, Y. Liu, and K. Wong. A chaotic image encryption scheme owning temp-value feedback. *Comm. Nonlinear Sci. Numer. Simul.*, 19(10):3653–3659, 2014.

[82] B. Norouzi, S. Mirzakuchaki, S. Seyedzadeh, and M. Mosavi. A simple, sensitive and secure image encryption algorithm based on hyper-chaotic system with only one round diffusion process. *Multimedia Tools Appl.*, 71(3):1469–1497, 2014.

[83] T. Rappaport. *Wireless Communications: Principles and Practice*. 2nd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.

[84] R. Ahlswede and I. Csiszar. Common randomness in information theory and cryptography - Part I: Secret sharing. *IEEE Trans. Inf. Theory*, 39(4): 1121–1132, 1993.

[85] G. Durgin. *Space-time Wireless Channels*. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.

[86] U. Maurer. Secret key agreement by public discussion from common information. *IEEE Trans. Inf. Theory*, 39(3):733–742, 1993.

[87] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik. Radio-telepathy: Extracting a secret key from an unauthenticated wireless channel. *Proc. ACM Int. Conf. Mobile Comput. Netw.*, San Francisco, CA, USA, 2008, 128–139.

[88] B. Azimi-Sadjadi, A. Kiayias, A. Mercado, and B. Yener. Robust key generation from signal envelopes in wireless networks. *Proc. ACM Conf. Comput. Commun. Security*, Alexandria, VA, USA, 2007, pp. 401–410.

[89] S. Jana, S. Premnath, M. Clark, S. Kasera, N. Patwari, and S. Krishnamurthy. On the effectiveness of secret key extraction from wireless signal strength in real environments. *Proc. ACM Conf. Mobile Comput. Netw.*, New York, NY, USA, 2009, 321–332.

[90] N. Patwari, J. Croft, S. Jana, and S. Kasera. High-rate uncorrelated bit extraction for shared secret key generation from channel measurements. *IEEE Trans. Mobile Comput.*, 9(1):17–30, 2010.

[91] G. Brassard and L. Salvail. Secret key reconciliation by public discussion. in *Advances in Cryptology*, vol. 765, New York, NY, USA: Springer-Verlag, 1994, 410–423.

[92] C. Bennett, G. Brassard, and J. Robert. Privacy amplification by public discussion. *SIAM J. Comput.*, 17(2):210–229, 1988.

[93] W. Buttler, S. Lamoreaux, J. Torgerson, G. Nickel, C. Donahue, and C. Peterson. Fast, efficient error reconciliation for quantum cryptography. *Phys. Rev. A*, 67(5):052303.1–052303.8, 2003.

[94] B. Zan, M. Gruteser, and F. Hu. Key agreement algorithms for vehicular communication networks based on reciprocity and diversity theorems. *IEEE Trans. Veh. Technol.*, 62(8):4020–4027, 2013.

[95] K. Zeng, D. Wu, A. Chan, and P. Mohapatra. Exploiting multiple-antenna diversity for shared secret key generation in wireless networks. *Proc. IEEE INFOCOM*, San Diego, CA, USA, 2010, 1837–1845.

[96] R. Wilson, D. Tse, and R. Scholtz. Channel identification: Secret sharing using reciprocity in ultrawideband channels. *IEEE Trans. Inf. Forensics Secur.*, 2(3): 364–375, 2007.

[97] M. G. Madiseh, M. L. McGuire, S. W. Neville, and A. Shirazi. Secret key extraction in ultra wideband channels for unsynchronized radios. *Proc. Commun. Netw. Services Res. Conf.*, Halifax, NS, Canada, 2008, 88–95.

[98] M. G. Madiseh, S. He, M. L. McGuire, S. W. Neville, and X. Dong. Verification of secret key generation from UWB channel observations. *Proc. IEEE Int. Conf. Commun.*, Dresden, Germany, 2009, 1–5.

[99] A. Hassan, W. Stark, J. Hershey, and S. Chennakeshu. Cryptographic key agreement for mobile radio. *Digital Signal Process.*, 6(4):207–212, 1996.

[100] Q. Wang, H. Su, K. Ren, and K. Kim. Fast and scalable secret key generation exploiting channel phase randomness in wireless networks. *Proc. IEEE INFOCOM*, Shanghai, China, 2011, 1422–1430.

[101] H. Koorapaty, A. Hassan, and S. Chennakeshu. Secure information transmission for mobile radio. *IEEE Commun. Lett.*, 4(2):52–55, 2000.

[102] A. Goldsmith. *Wireless Communications*. New York, NY, USA: Cambridge Univ. Press, 2005.

[103] A. Sayeed and A. Perrig. Secure wireless communications: secret keys through multipath. *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Las Vegas, NV, USA, 2008, 3013–3016.

[104] Q. Wang, K. Xu, and K. Ren. Cooperative secret key generation from phase estimation in narrowband fading channels. *IEEE J. Sel. Areas Commun.*, 30(9):1666–1674, 2012.

[105] K. Ren, H. Su, and Q. Wang. Secret key generation exploiting channel characteristics in wireless communications. *IEEE Wireless Commun.*, 18(4):6–12, 2011.

[106] C. Chen, and M. Jensen. Secret key establishment using temporally and spatially correlated wireless channel coefficients. *IEEE Trans. Mob. Comput.*, 9(2):205–215, 2011.

[107] T. Shimizu, H. Iwai, and H. Sasaoka. Physical-layer secret key agreement in two-way wireless relaying systems. *IEEE Trans. Inf. Forensics Secur.*, 6(3):650–660, 2011.

[108] Y. Shehadeh, O. Alfandi, and D. Hogrefe. Towards robust key extraction from multipath wireless channels. *J. Commun. Networks*, 14(4):385–395, 2012.

[109] T. Aono, K. Higuchi, T. Ohira, B. Komiyama, and H. Sasaok. Wireless secret key generation exploiting reactance-domain scalar response of multipath fading channels. *IEEE Trans. Antennas Propag.*, 53(11):3776–3784, 2005.

[110] N. Wang, N. Zhang, and T. A. Gulliver. Cooperative key agreement for wireless networking: key rates and practical protocol design. *IEEE Trans. Inf. Forensics Secur.*, 9(2):272–284, 2014.

[111] P. Billingsley. *Ergodic Theory and Information*, Hoboken, NJ, USA: Wiley, 1965.

[112] J.P. Eckmann and D. Ruelle. Ergodic theory of chaos and strange attractors. *Rev. Mod. Phys.*, 57(3):617–656, 1985.

[113] F. Christiansen, G. Paladin, and H. H. Rugh. Determination of correlation spectra in chaotic systems. *Phys. Rev. Lett.*, 65(17):2087–2090, 1990.

[114] COST 207 Management Committee. COST 207 digital land mobile radio communications. Commission Eur. Communities, Luxembourg, Final Rep., 1989.

[115] R. Chang. Synthesis of band-limited orthogonal signals for multichannel data transmission. *Bell Syst. Technol. J.*, 45(10):1775–1796, 1966.

[116] S. Weinstein and P. Ebert. Data transmission by frequency division multiplexing using the discrete Fourier transform. *IEEE Trans. Commun. Technol.*, 19(5):628–634, 1971.

[117] H. Ochiai and H. Imai. On the distribution of the peak-to-average power ratio in OFDM signals. *IEEE Trans. Commun.*, 49(2):282–289, 2001.

[118] R. Bauml, R. Fisher, and J. Huber. Reducing the peak-to-average power ratio of multicarrier modulation by selected mapping. *Elect. Lett.*, 32(22):2056–2057, 1996.

[119] N. Carson and T. A. Gulliver. Peak-to-average power ratio reduction of OFDM using repeat-accumulate codes and selective mapping. *Proc. IEEE Int. Symp. Inf. Theory*, Lausanne, Switzerland, 2002, 244.

[120] H. Breiling, S. Muller-Weinfurtner, and J. Huber. SLM peak-power reduction without explicit side information. *IEEE Commun. Lett.*, 5(6):239–241, 2001.

[121] N. Ohkubo and T. Ohtsuki. Design criteria for phase sequences in selected mapping. *Proc. IEEE Veh. Technol. Conf.*, Jeju, South Korea, 2003, 373–377.

[122] G. Zhou and L. Peng. Optimality condition for selected mapping in OFDM. *IEEE Trans. Signal Process.*, 54(8):3159–3165, 2006.

[123] L. Ning, M. Yang, Z. Wang, and Q. Guo. A novel SLM method for PAPR reduction of OFDM system. *Proc. IEEE Veh. Technol. Conf.*, Yokohama, Japan, 2012, 1–5.

[124] C. Li, S. Wang, and C. Wang. Novel low-complexity SLM schemes for PAPR reduction in OFDM systems. *IEEE Trans. Signal Process.*, 58(5):2916–2921, 2010.

[125] A. Ghassemi and T. A. Gulliver. Low-complexity distortionless techniques for peak power reduction in OFDM communication systems. *J. Comput. Networks Commun.*, 2012:1–13, 2012.

[126] G. Wu and D. Baleanu. Discrete fractional logistic map and its chaos. *Nonlinear Dyn.*, 75(1):283–287, 2014.

[127] Q. Hu, J. Liu, and D. Yu. Stability analysis on rough set based feature evaluation. in *Rough Sets and Knowledge Technology*, vol. 5009, New York, USA: Springer, 2008, 88-96.

[128] S. Wang, C. Li, K. Lee, and H. Su. A novel low-complexity precoded OFDM system With reduced PAPR. *IEEE Trans. Signal Process.*, 63(6):1366–1376, 2015.

[129] S. A. Adegbite, S. McMeekin, and B. G. Stewart. Modified Shapiro-Rudin sequences for SLM-PAPR reduction in wireless OFDM systems. *Proc. CSNDSP*, Manchester, UK, 2014, 302–307.

[130] S. Le, S. Al-Samahi, K. Boon, C. Tsimenidis, and B. Sharif. Selected mapping without side information for PAPR reduction in OFDM. *IEEE Trans. Wireless Commun.*, 8(7):3320–3325, 2009.

[131] A. Goel, P. Gupta, and M. Agrawal. Generalized M-2M mapping scheme for SLM and PTS based OFDM systems without side-information. *Wireless Pers. Commun.*, 74(2):285–305, 2014.

[132] H. Liang. Integrating CE and modified SLM to reduce the PAPR of OFDM systems. *Wireless Pers. Commun.*, 80(2):709–722, 2015.