REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

A NEW IMAGE CIPHER USING COLOR SPACE TRANSFORM AND LORENZ MAP

Fatıma Yeşim İKİKAT

MASTER OF SCIENCE THESIS
Department of Computer Engineering
Program of Computer Engineering

Advisor
Assoc. Prof. Dr. Sırma YAVUZ

April, 2019

REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

A NEW IMAGE CIPHER USING COLOR SPACE TRANSFORM AND LORENZ MAP

A thesis submitted by Fatıma Yeşim İKİKAT in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE** is approved by the committee on 03.04.2019 in Department of Computer Engineering, Program of Computer Engineering .

Assoc. Prof. Dr. Sırma YAVUZ
Yildiz Technical University
Advisor

Approved By the Examining Committee

Assoc. Prof. Dr. Sırma YAVUZ, Advisor
Yildiz Technical University ______________________

Assoc. Prof. Dr. M. Fatih AMASYALI, Member
Yildiz Technical University ______________________

Assist. Prof. Dr. Fatih KELEŞ, Member
İstanbul University - Cerrahpaşa ______________________

I hereby declare that I have obtained the required legal permissions during data collection and exploitation procedures, that I have made the in-text citations and cited the references properly, that I haven’t falsified and/or fabricated research data and results of the study and that I have abided by the principles of the scientific research and ethics during my Thesis Study under the title of A New Image Cipher Using Color Space Transform and Lorenz Map supervised by my supervisor, Assoc. Prof. Dr. Sırma YAVUZ. In the case of a discovery of false statement, I am to acknowledge any legal consequence.

Fatıma Yeşim İKİKAT

Signature

To my mom...

ACKNOWLEDGEMENTS

It is an indisputable fact that chaos is one of the most interesting and fascinating phenomena that nature presents to humanity. And combining this entity, which is secretly present in every part of the universe, with cryptography was one of the most informative works I've done throughout my education life.

In this thesis, I aimed to transform my knowledge about chaos, chaotic cryptography and image encryption as a result of my researches into a secure cryptographic design. I used basic image processing techniques to develop a plain text attack-resistant method, which is the weakness of many chaotic image encryption algorithms. My goal was to achieve a simple, secure and easy to understand cipher design that would rivet my knowledge.

I would like to thank all my lecturers that I have benefited from during my education life, especially my mentor Assoc. Prof. Dr. Sırma YAVUZ, who guided me on this path. I wouldn't have written this thesis today, if it wasn't their sacrifices and efforts.

I also would like to thank my friends; Nihal ALTUNTAŞ for during the time I tried to complete my thesis, whenever I knocked on her door, which happened quite often, being there to help me; Zeynep Banu ÖZGER, for the great benefit of her suggestions during the process of preparation and submission of this thesis; Nurgül YÜZBAŞIOĞLU for giving me encouragement throughout this exhausting process; Hacer BAYIROĞLU, Behire GÜRHAN and Merve ARITÜRK for their support and wonderful friendship.

Of course, I would like to express my endless gratitude to the Fifth Floor Team, especially Dilek KUZALİÇ and Selin YALÇIN. During the last few months, they have to put up with me and supported me with superhuman effort. Whenever I needed a little push, they extended their hands; this push could be a pumpkin dessert, or just chat.

And finally, I want to thank my dearest friend Seyda BAYARSLAN. I know she has been waiting for this day for a long time, and gladly declare that finally we can continue where we left off.

But there was one that she waited for three years to finish this thesis. Mom, I dedicate this thesis to you. For your endless patience and unique partnership.

Fatıma Yeşim İKİKAT

TABLE OF CONTENTS

LIST OF SYMBOLS

A	Alice; sender
B	Bob; intended receiver
C	Ciphertext
D	Decryption algorithm
E	Eve; eavesdropper
E	Encryption algorithm
K	Secret key
K_p	Public key
K_s	Private key
P	Plaintext
β	Physical dimensions of the layer
ρ	Rayleigh number
σ	Prandtl number

LIST OF ABBREVIATIONS

AES	Advanced Encryption Standard
ARX	Bitwise XOR
BCE	Before the Common Era
CE	Common Era
CIA	Confidentiality, Integrity, Availability
DCT	Discrete Cosine Transform
DES	Data Encryption Standard
ECB	Electronic Codebook
ENIGMA	Engine for the Neutralizing of Information by the Generation of Miasmic Alphabets
GCHQ	The Government Communications Headquarters
HTTP	Hyper-Text Transfer Protocol
JPEG	Joint Photographic Experts Group
LEA	Link Encryption Algorithm
MLE	The Maximal Lyapunov Exponent
OFDM	Orthogonal Frequency Division Multiplexing
OKW/Chi	Oberkommando der Wehrmacht Chiffrierabteilung
Ph.D.	Doctor of Philosophy
PRNG	Pseudo-Random Number Generator
QC	Quantization Coefficient
RGB	Red – Green – Blue Color Space
RSA	Rivest – Shamir – Adleman Cryptosystem
SAC	Strict Avalanche Criterion
SIS	Signal Intelligence Service

S-Box	Substitution Box
XOR	Exclusive OR
YCbCr	Luminance – Chroma:Blue – Chroma:Red Color Space
YUV	Luminance – Bandwidth – Chrominance Color Space
1-D	One Dimensional
2-D	Two Dimensional
3-D	Three Dimensional

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

A New Image Cipher Using Color Space Transform and Lorenz Map

Fatıma Yeşim İKİKAT

Department of Computer Engineering
Master of Science Thesis

Advisor: Assoc. Prof. Dr. Sırma YAVUZ

Information security has been an important issue for humanity in every age. But nowadays, the diversification of communication facilities, especially the rapid development of the Internet, has made it possible for this need to reach the highest level.

Cryptography is one of the most important tools to meet the need for information privacy. Since it was invented, cryptography techniques are constantly changing in parallel with the change in the communication channel used and the type of information transmitted. Nowadays, a wide range of secure cryptographic methods are developed and used.

However, standard encryption algorithms show different efficiencies according to the type of encrypted data and the compatibility of these methods with different file types are discussed. Especially multimedia files have different features than text files and these features have several advantages and disadvantages in encryption. For this reason, cryptography of image files is examined as one of the special subdivisions of cryptology.

Image data can be encrypted with conventional block encryption or chain encryption algorithms. But this is not efficient enough due to the large size of the data, the strong correlations between the pixel values and the high redundancy. For this reason, different methods for image encryption have been proposed over the years. The chaotic methods have been tried and tested since it is more than adequate for image encryption.

Chaotic functions can be integrated into image encryption in a variety of ways and can be used for a variety of purposes. One of them is the mixing of the picture with a chaotic map. Researchers have observed that confusion and diffusion requirements in encryption can be met by using chaotic maps and have shown particular interest in this technique.

In this thesis, a new algorithm is proposed using Lorenz maps and YCbCr color space transformation for encrypt a colored image. Algorithm was constructed to cover encryption and decryption processes and reliability was examined with various tests. It is also proposed what can be done for efficiency and development. Thus, a unique method that can be used for image encryption has been try to developed.

Keywords: Chaotic cryptography, image encryption, lorenz map, color space transform

ÖZET

Renk Uzayı Dönüşümü ve Lorenz Haritaları ile Yeni Bir Görüntü Şifreleme Algoritması

Fatıma Yeşim İKİKAT

Bilgisayar Mühendisliği Anabilim Dalı
Yüksek Lisans Tezi

Danışman: Doç. Dr. Sırma YAVUZ

Bilgi güvenliği her çağda insanlık için önemli bir konu olmuştur. Ama günümüzde iletişim olanaklarının çeşitlenmesi, özellikle de internetin hızlı gelişimi bu ihtiyacın olabileceği en yüksek düzeye ulaşmasını sağlamıştır.

Kriptografi, bilgi mahremiyeti ihtiyacını karşılamada en önemli araçlardan biridir. İcat edildiği günden bu güne, kriptografik teknikler, kullanılan iletişim kanalları ve aktarılan bilgi türüne paralel olarak sürekli değişmektedir. Günümüzde geniş bir yelpazede güvenli kriptografik yöntemler geliştirilmekte ve kullanılmaktadır.

Ancak standart şifreleme algoritmaları, şifrelenen datanın türüne göre farklı verimlilikler göstermekte ve bu yöntemlerin farklı dosya tiplerine uygunluğu tartışılmaktadır. Özellikle multimedya dosyaları, metin dosyalarından farklı özelliklere sahiptir. Bu özellikler şifrelemede çeşitli avantajlar ve dezavantajlar getirmektedir. Bu sebeple görüntü şifreleme kriptolojinin özel altdallarından biri olarak incelenmektedir.

Görüntü verileri, klasik blok şifreleme veya zincir şifreleme algoritmaları ile şifrelenebilir. Ama bu, verinin büyük olması, piksel değerleri arasında güçlü korelasyonlar bulunması ve kritik olmayan verinin fazlalığı gibi sebepler dolayısıyla yeterince verimli değildir. Bu nedenle, yıllar içinde görüntü şifreleme için farklı metodlar önerilmiştir. Kaotik metodlar, görüntü şifrelemeye fazlasıyla uygun olduğu için denenmiştir ve denenmeye devam etmektedir.

Kaotik fonksiyonlar görüntü şifrelemeye çok çeşitli şekillerde entegre edilebilmekte ve çeşitli amaçlarla kullanılabilmektedir. Bunlardan biri de resmin bir kaotik yer değiştirme haritası ile karıştırılmasıdır. Araştırmacılar, kaotik yer değiştirme haritaları kullanarak şifrelemedeki karıştırma ve yayılma gerekliliklerinin karşılanabileceğini görmüş ve bu tekniğe özel bir ilgi göstermişlerdir.

Bu tezde renkli bir resmi şifrelemek için Lorenz haritaları ve YCbCr renk uzayı dönüşümünden faydalanılarak yeni bir algoritma önerilmiştir. Algoritma şifreleme ve şifre çözme süreçlerini kapsayacak şekilde oluşturulmuş ve çeşitli testlerle güvenilirliği incelenmiştir. Ayrıca verimliliği ve geliştirilmesi için yapılabilecekler de önerilmiştir. Böylece görüntü şifreleme için kullanılabilecek özgün bir yöntem geliştirilmesine çalışılmıştır.

Anahtar Kelimeler: Kaotik kriptografi, görüntü şifreleme, lorenz haritaları, renk uzayı dönüşümü

YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

1
Introduction

Communication is the ability to transfer knowledge, that is one of the most basic skills not just people, but all living things have. This knowledge can sometimes be the location of a food, sometimes an enemy warning. Communication is vitally important in all cases, and in most situations, it may be critical to ensure the continuity of information transfer.

The three main elements of communication are source, receiver and message. These three elements need to be reliable in order to information transfer to take place in a healthy manner. Broadly speaking, the source should be able to control the message is sending to which receiver or receivers, the receiver should be able to verify the source of the message, and both should be able to trust that the content of the message is correct. Long story short, wherever there is communication, privacy emerge as a major problem.

As the natures of the communication channel and transferred knowledge changes, the proposed solutions to this problem also change. In the centuries when people communicated with paper and pen, a simple encryption was enough to secure information. But nowadays, while digital communication networks enter every part of life, these simple encodings have been replaced by complex encryption algorithms.

1.1 Literature Review

Thanks to the improvements in cryptography, we can use a number of strong encryption algorithms such as DES, AES and RSA. But not all requirements in the encryption area have been met. At the beginning of these needs comes encryption of media files such as image, audio and video files.

Most of the information transferred on digital networks today is media data. According to HTTP Archive, as of November 2018, images make up on average 21% of a total webpage's weight [1]. This fact means that images can now be considered one of the most used forms of information. Therefore, it is important to ensure the privacy of the image files. Image encryption have applications in various fields, including wireless communications, multimedia systems, medical imaging, telemedicine, and military communications. For this reason, having a secure and reliable means for communicating with images becomes a necessity, and its related issues must be carefully considered [2].

Surely, images can be encrypted with traditional methods that used to encrypt text. However, these methods are not efficient enough due to they have different qualities than text data. Images has some specific features like; they have bigger amount of data, higher redundancy and stronger correlation between pixels. Sometimes, image ciphers have to meet some specific requirements, like real-time processing, fidelity reservation, image format consistency, data compression for transmission. Simultaneous fulfillment of these requirements along with high-security and high-quality demands has presented great challenges to real-time imaging practice [2]. Hence standard stream or block cipher algorithms may not be suitable to meet all those needs.

Today, researchers are turning to alternative ways of encrypting images. But before look at these alternative ways, it is necessary to analyze the differences between image data and text data encryption applications. Basically, these differences can be listed as:

- When a text is encrypted, it must be decrypted to the original plaintext in a full lossless manner. But for decrypted image, containing small corruption is acceptable because of human perception is limited.
- Text data are sequences of characters. They can be encrypted directly by using block or stream ciphers. However, digital images are usually represented as two-dimensional arrays. For enciphering 2-D arrays of data with text cipher algorithms, they must be converted to 1-D arrays.
- Because the storage space of an image file is very large, encrypt or decrypt images directly may not be an ideal solution. Besides, traditional cryptosystems take so much time to encrypt the image data. Therefore, it would be more advantageous to use a method specific to image encryption only.

These differences in image encryption also bring with unique challenges. Unlike weaknesses in text encryption, image encryption has its own difficulties like; the value that each pixel can take is limited to a small range, there is pixel correlation and the fact that the breaking of encrypted data at a certain rate is sufficient to extract meaningful information. Therefore, the techniques to be used in image encryption should be examined in a different way than text encryption. The methods that are predicted to be highly successful in theory can cause unnoticeable weaknesses at first sight.

Over the years, computer scientists have experimented with different methods to encrypt image files in a more optimal way such as homomorphic image processing, chaotic functions, ECB mode, Fourier transform, OFDM and so on. In some methods, the information stored by all pixels is encrypted; when the image file is requested to be transferred easily, or has size as small as possible, only the information used during image compression is encrypted.

Naturally, each method has its own advantages and handicaps. For example, chaotic ciphers, which are one of the most preferred methods in the field of image encryption in recent years, have a lot of disadvantages in practice, despite the fact that they provide theoretically security requirements. Cryptanalysis studies on image encryption methods using chaotic functions and maps reveal that many algorithms have failed in the face of basic attacks.

1.2 Objective of the Thesis

One of the most useful ways to eliminate an image encryption algorithm is chosen plaintext attack. And most of the proposed chaos-based algorithms, especially that using chaotic maps, are not as robust against this attack adequately. In his Ph.D. thesis [3] Özkaynak has managed to break the selected chaotic encryption algorithms with using chosen plaintext attack and some other algebraic techniques in just a few steps. Özkaynak's this study showed that, no matter how complex a chaotic system is, the important issue is how to implement the algorithm.

In this thesis, a new chaotic image encryption algorithm that can be capable of withstanding to chosen plaintext attack has been tried to be proposed.

1.3 Hypothesis

The most direct way to make an encryption algorithm resistant to plain text attack is to guarantee that most of the pixels in the cipher image will change even if a single pixel of the plain image changes. Although this may seem to be easily solved by adding noise to the image during encryption process, knowing the added noise makes it possible to remove the noise from the image. So, despite the addition of noise, the plain text attack can succeed.

The most important feature of the method proposed in this thesis is that the plain image is passed through certain processes and the result is used as noise. Thus, the noise added to the picture cannot to be eliminated and plain text attack cannot be used.

1.4 Organization of the Thesis

This thesis consists of five chapters. The first section is the introduction section you are currently reading.

The second chapter, called Cryptography and Chaos, contains adequate information about cryptography and chaos theory. In this chapter, the first subsection is devoted to the history of cryptography, the second subsection to the basic principles of modern cryptography, and the third to the chaos theory and the relationship between cryptography and chaos.

The third chapter, called Chaotic Image Encryption, there is a literature review about chaotic image encryption. In this chapter, various studies on chaotic image encryption, and techniques are discussed.

The fourth chapter is devoted to the proposed method, which is the reason why this thesis is written. In this section, basic information about the techniques included in the method is given; how these techniques are implemented, and process steps of the algorithm is explained in detail.

The fifth section contains the test results of the proposed algorithm and discussions on its success. In this section, the results of different tests applied to the outputs produced by the algorithm are examined.

In the sixth and last chapter, the results of the thesis, suggestions on the development of the algorithm and the contributions it can make are emphasized.

I hope this thesis will be useful for those who want to work in this field.

2
Cryptography and Chaos

The etymology of the word “cryptography” or “cryptology” based on the Greek words: “kryptós” (ϰρυπτς), “hidden, secret”; and “graphein” (γραφειν), “to write”, or “-logia” (-λγια), “study” [4].

Cryptography is the study and practice of mathematical techniques for secure communication in the presence of adversaries. The basic goals of cryptography are privacy and reliability. Thus, it is intended to ensure that the information transmitted between the parties is inaccessible to a third party and both parties can rely on the information coming from each other, not from another source.

The cryptographers propose and test different methods for the realization of these objectives, and in doing so benefit from the facilities of mathematics and computer science. The proposed methods may vary depending on the nature of the network they will use, the structure of the information they transfer, and the capacity of the platform on which they will run the encryption method. Therefore, according to needs, dozens of cryptography methods have been proposed and broken to date.

In 1976 Diffie and Hellman proclaimed: *“We stand today on the brink of a revolution in cryptography”* [5]. The revolution they mentioned today took place, but did not end. Today, studies on cryptography still continue to expand with the contribution of many sciences; especially mathematics, computer science, electrical engineering, communication science, and physics.

2.1 History of Cryptography

It is said that cryptography is a young science. Indeed, cryptography as a science has almost a century of history. But as an art, there are actually thousands of years. Because not only in our century, but in every age, information security is an important problem need to be solved; so that many times it has changed the history. The famous story of ENIGMA is evidence from recent history to this. For the same reason, if we want to talk about cryptography, it is necessary to look centuries ahead.

The known history of cryptography dates back to 2000 BCE, the Ancient Egypt. David Kahn, the writer of The Codebreakers that one of the most comprehensive studies on cryptography history, narrates this situation as follows: *"ON A DAY nearly 4,000 years ago, in a town called Menet Khufu bordering the thin ribbon of the Nile, a master scribe sketched out the hieroglyphs that told the story of his lord's life—and in so doing he opened the recorded history of cryptology"* [6].

Today, Menet Khufu was located in the Beni Hasan, an Ancient Egyptian cemetery site, near the city of El Minye in Egypt. And there are still some cryptic epitaphs in the tombs continue to appear. But the archaeological relics that found here and the rest of the Egypt do not refer to a complete cryptographic system used to hide information. According to researchers, these first codes that were created with writing some unusual hieroglyphic symbols here and there in place of the more ordinary ones, were probably effort to create mystery, intrigue, or even amusement for literate onlookers.

In its first 3000 years, cryptography did not grow stably. It appeared in many places and in most of them it died with the deaths of its civilizations. In some places it survived and embedded in literature. After all, more was lost than retained. Only toward the Western Renaissance, the accreting knowledge began to gain an acceleration. According to the Kahn, *"the story of cryptology during these years is, in other words, exactly the story of mankind"* [6].

One of the earliest examples of encryption in the sense that we know was found in Mesopotamia. It is clearly seen that some cryptic clay tablets belonging to the period are for purpose of hiding information. It has been found that one of these tablets, which date back to around 1500 BCE, is probably encoded in a handicraft recipe for commercially valuable pottery secrets.

The Holy Scriptures themselves have not escaped a touch of cryptography, or proto-cryptography, to be precise, for the element of secrecy is lacking. When we reach around 500-600 BCE, we see that Hebrew scholars used simple mono-alphabetical substitution ciphers such as Atbash cipher. Although Hebrew tradition offers at least two such conversions in the Old Testament; today, simple scribal manipulations like Atbash that can be found in different cultures are considered as ancient scribes for amusing themselves with word and alphabet games.

Different examples of mono-alphabetical substitution ciphers are also found in China, India, Egypt, Mesopotamia and Ancient Greek. Through researches we know that

Spartan military used the scytale transposition cipher, Herodotus tells us about the stenography by describing the secret messages physically concealed beneath wax on wooden tablets or as a tattoo on a slave's head concealed by regrown hair, another Greek, Polybius, developed the "Polybius square", also the Romans used the "Caesar cipher" and its variations.

It is clear from this historical process that when a culture reaches a certain level, especially in the literacy rate, cryptography is spontaneously emerging and starts to be used because of the different demands and needs of people. This situation also explains the development of cryptography in many different civilizations and places independently. But of course, there are long periods in which this development slows down or even stops.

With the collapse of Roman Empire, Medieval that today we call it the "Dark Ages" began in Europe. In this period, the decline in literacy rate and suppression of cultures brought along scientific and artistic deprivation. Naturally, cryptography also got its share. For about a thousand years, between CE 500-1400, cryptography not only stopped in the western civilization; but also started to be associated with occultism. Further, it was seen as black magic on its own. This view was based on the superficial similarity in the context of "seeing what is hidden" between cryptography and prophecy. These effects are so profound that, even today, cryptography gives a mysterious impression and it acquires an esoteric place in popular culture.

In the middle of the 8th century, the first examples of modern cryptography were given by Arabs [6], [7]. In the Muslim world that including North Africa, Mesopotamia and the Arabian peninsula, it is seen that there are two different branches which are called "muamma" that sets out the rules and procedures for resolving a ciphered text and "ta'miye" that means writing a text with a secret code.

As a result of the formation of various divans and offices of Islamic states, the necessity of hiding some of the information in official correspondence necessitated learning of the techniques of encryption and decryption methods. As a matter of fact, writers such as Al-Kindi, Ibn Tabatabai and Ibn 'Adlan stated that they wrote the books about this subject on the request of the administrators [7].

It is said that, the first work called "Kitabîl-Mu'ammâ" was written by the famous Arabian philologist Khalil b. Ahmed [7]. Al-Qalqashandi's encyclopedic work "ubu'l-aşâ fî ınâati'l-inşâ" which talks about secret correspondence techniques in the fifth chapter [8], and the books that he cites; "Miftâu'l-künûz fî îżâi'r-rumûz" that were writed by Ibn al-Durayhim, and "Îżâu'l-mübhem fî alli'l-mütercem" which has not survived but has summarized in Miftâu'l-künûz, are important examples for works about cryptography around in the 1300s [9].

The first cryptanalysis studies are also based on this period. Frequency analysis, which is the only known technique for breaking mono-alphabetical substitution ciphers until the World War II, was found by an Arab mathematician, Al-Kindi in sometime around 800. He also described the first cryptanalytic techniques that including some for polyalphabetic ciphers, cipher classification, Arabic phonetics and syntax, and collected these works in his book on cryptography; “Risâle fi’stirâci’l-muammâ” (Manuscript for the Deciphering Cryptographic Messages) [10].

As an example of a ciphertext method that is much closer to our geography, “siyākath” (siyâkat) can be mentioned [11]. Siyākath is an old type of writing which is used in archival documents and records, very difficult to read, intricate, and does not carry art. In the emergence of this type of cipher writing, the need to security of the state, fast writing of the records and to keep a shorter space have been effective. It is accepted that the Siyākath comes up in the time of the Abbasids, and it is used in the Seljuks and other Islamic states, through Iran through the Ottomans, especially through the Ilkhanates.

The official financial records of the Ottoman Empire, from the period of Sultan Mehmed the Conqueror to Sultan Abdülaziz, especially the destruction and accounting books, and the notes from them, were written with the siyākath by using Persian patterns. As a reflection of the Ilkhanate tradition, especially the books containing the tax records are written in the style of the siyākath which has become more and more stylized in time and even their numbers are indicated as the siyākath numbers.

The numbers used in the writings of siyākath are also written in a special system. Instead of the Indian numerals, the Arabic numerals were ciphered by abbreviating the letters of their names and called them divan numbers because of the use of the court in the financial and accounting records of the divan. It is known that the divan numbers, which were estimated to have emerged in the time of the Umayyads, were passed to the Seljuks and then to the Ilkhanate and Ottomans [11].

The rise of cryptography in Europe began with the Renaissance. During this age, which was directly linked to the emergence of modern bureaucracy, almost every state and institution used and developed secret correspondence methods. So that, fulltime cipher secretaries were employed for encryption, decryption, generation of new keys and solving intercepted dispatches. In some places, apart from these secretaries, there were also cryptanalysis specialists or were called upon in need.

In this period, the first examples of the homophonic substitution ciphers that a bit more resistant ciphers against frequency analysis were also used by the officers in Mantua [12]. Ciphers which make the frequency analysis useless, and that also will be the basis of enigma in the future, was polyalphabetic ciphers. Leon Battista Alberti explained the first polyalphabetic cipher in twenty-five-page manuscripts in Latin in 1466 or early 1467, constitutes the West's oldest extant text on cryptanalysis. Alberti's three remarkable firsts—the earliest Western exposition of cryptanalysis, the invention of polyalphabetic substitution, and the invention of enciphered code—make him the "Father of Western Cryptology" [6].

The most elaborate organization on this subject was probably belong to Venice's. Venice was owed this success to Giovanni Soro, who was appointed as secretary of cryptanalysis in the 1506, and the first major cryptanalyst in the West. In fact, even the Papacy was sending the ciphers that no one could solve in Rome.

Johannes Trithemius, a German lexicographer, chronicler, cryptographer and occultist, published the first printed book on cryptography; "Poligraphia" in 1518 and according to some researchers, its other part, "Steganographia" in 1606. He invented the "tabula recta", that is a critical component of the "Vigenère cipher" that would be developed later. But these two works were not exactly scientific, they were related to occultism and witchcraft.

In contrast to Germany, cryptic communication techniques were no longer seen as a black art, but rather found place itself in political intrigues in Britain and France. Thomas Phelippes, England's first great cryptanalyst, helped uncover the evidence for the execution of Mary, Queen of Scots, and thus protect the Elizabeth I of England by deciphering the letter including the assassination plan for the throne of Mary in 1586. Also, François Viète, known today as the "Father of Algebra", did cryptanalysis for King Henry IV of France in 1589. Although he failed to solve the cryptic letter of the King of Spain, who was planning to attack France until Henry was defeated with it, this delay did not affect his luck. Because the solution he developed would enable him to resolve all subsequent letters [6].

Although the "Vigenère cipher", which is one of the most important polyalphabetic encryption techniques of the period, has been incorrectly attributed to the French cryptographer Blaise de Vigenère, it was first introduced by an Italian cryptologist; Giovan Battista Bellaso in 1553. But in 1586, Vigenère developed this method into a more powerful and autokey cipher.

The cryptography, which attracted attention again and began to be used without fear, did not show much progress until the 19th century. However, with a clearer understanding of the relationship between mathematics and cryptography in the 19th century, specific developments are gradually replaced by systematic developments, so that cryptography begins to gain momentum as a science. Furthermore, the journey of communication facilities from paper to electrical signals was the reason for the fundamental change and acceleration in cryptography.

Among those who worked on cryptography during this period were Charles Babbage that known as “the Father of the Computer”, the Dutch linguist and cryptographer Auguste Kerckhoffs, the Prussian infantry officer, cryptographer and archeologist Friedrich Kasiski and even the famous gothic literature writer Edgar Allan Poe [6].

Kerckhoffs and Kasiski are particularly important from these people. Because, Friedrich Kasinski proposed a new technique of cryptanalysis and was the source of an important test in his book “Die Geheimschriften und die Dechiffrir-Kunst” (Secret writing and the Art of Deciphering) in 1863. This technique that known as “Kasiski examination” now, relied on the analysis of gaps between repeated fragments in the ciphertext and can give hints as to the length of the key used.

Auguste Kerckhoffs has defined six principles of practical cipher design in his two significant essays that published in 1883 in “Le Journal des Sciences Militaires” (Journal of Military Science) entitled “La Cryptographie Militaire” (Military Cryptography). The most famous of these rules is the second one, called “Kerckhoffs’s Principle”, and says: “The design of a system should not require secrecy, and compromise of the system should not inconvenience the correspondents”. This rule can be understood as the idea that the security of a cryptosystem must to be dependent on only to the secrecy of the key, not on the confidentiality of any part of the system. And it is the most important and unchangeable condition of creating a safe cryptographic algorithm today.

In the late 19th century, the importance and use of cryptography had increased day by day. The reasons such as the widespread of telegraph lines, the fact that political developments caused the need for intelligence, increase the importance of secret communication capability. Eventually in the First World War, this capability began to turn into an important weapon in the war of superiority between states and become effective in determining the outcome of battles.

Along with the World War I, decoded texts changed the fate of many battles. But of course, its most important contribution was the inclusion of U.S.A in the battle and its direct effect on the result. The decoding by British Naval intelligence of the Zimmermann Telegram, a cable from the German Foreign Office sent via Washington to its ambassador Heinrich von Eckardt in Mexico, played a major part in bringing the United States that remained neutral until then into the war, on the side of Allied

Powers.

Eventually, the importance of having secure secret communication systems was well understood. In the short term between the first and the second world wars, the states have accelerated their work on this issue. The establishment of the Signals Intelligence Service by U.S.A in 1927 and the support that Polish naval-officers gave to the Japanese military with code and cipher development in 1920s can be given as obvious examples [6].

From now on, encryption and decryption processes needed to be automatic, error-free and fast. In 1917, Gilbert Vernam's invention of a teleprinter cipher in which used a previously prepared key kept on paper tape and combined character by character with the plaintext message to produce the cyphertext, led to the development of electromechanical devices as cipher machines, and to the only unbreakable cipher, the one-time pad. In the period before the Second World War, the use of mathematical methods in cryptography was also increased. The American engineer and US Army cryptographer William F. Friedman's application of statistical techniques to cryptanalysis and cipher development was an important event.

The most important cryptographic development of the Second World War was the development and breakdown of encryption machines. In this respect, the Second World War can be called a cryptography war. Mechanical and electromechanical encryption machines were widely used during the Second World War. Therefore, the development and cryptanalysis of these machines were important works for states.

The most of the different electro-mechanical cipher machines used for military intelligence such as: the ENIGMA used by the German Army, the Type B Cipher Machines that called codenames PURPLE and RED by the United States, used by the Japanese Foreign Office, TypeX that British adaptation of the Enigma used by the British armed forces, SIGABA that used by United States Army and Navy were the products of this short period of six years [6].

In the same period, a lot of institutions were established to perform cryptography and cryptanalysis studies like SIS by U.S., The Government Communications Headquarters (GCHQ) by U.K., OKW/Chi (Cipher Department of the High Command of the Wehrmacht) by Germans. There were numerous cryptography and cryptanalysis projects conducted in these institutions. Lots of important brains of 20th century like chess masters, mathematicians, logicians, linguists, engineers were working in these projects as cryptographer or cryptanalyst. Gordon Welchman, Max Newman, Wolfgang Franz and Alan Turing who is the conceptual founder of modern computing were among these people [6].

At the end of the two world wars, knowing that knowledge was a great power was also telling us that the rise of cryptography would never stop. In fact, the invention and spread of the computer, the development of cryptography also took a dizzying speed.

And today, cryptography is an important and inseparable part of information security, computer science and mathematics.

2.2 Basics of Modern Cryptography

The main purpose of modern cryptography is to ensure that the parties communicate safely in an unsafe communication channel. In this context, one of the basic concepts of cryptography is the secure channel. The secure channel is defined as a communication channel where parties can transfer information safely without the possibility of overhearing and tampering.

The perfectly secure channel in real life is impossible. The sender and the receiver are generally obliged to transfer data via channels that are accessible by third parties such as the Internet. Therefore, cryptographic methods attempt to emulate a secure channel in an unsafe environment and provide the same facilities to the user.

R. L. Rivest listed the properties that sought in a reliable communication system in [13] as follows:

- **Privacy:** An adversary learns nothing useful about the message sent.
- **Authentication:** The recipient of a message can convince himself that the message as received originated with the alleged sender.
- **Signatures:** The recipient of a message can convince a third party that the message as received originated with the alleged signer.
- **Minimality:** Nothing is communicated to other parties except that which is specifically desired to be communicated.
- **Simultaneous exchange:** Something of value (e.g., a signature on a contract) is not released until something else of value (e.g., the other party's signature) is received.
- **Coordination:** In a multi-party communication, the parties are able to coordinate their activities toward a common goal even in the presence of adversaries.

- **Collaboration threshold:** In a multi-party situation, the desired properties hold as long as the number of adversaries does not exceed a given threshold.

Some aspects in information security such as data confidentiality, data integrity, authentication, and non-repudiation are central to modern cryptography and these are properties of the secure channel. In the information security jargon, these three basic components of information security are called “CIA Triad”.

Data confidentiality is the privacy of transmitted data. In order to provide this condition, the transmitted data should not be understood by anyone other than the authorized person. Data integrity is guaranteed to ensure that the transmitted data is not exchanged during the communication. Authentication is intended to make sure that incoming data is received from the trusted source, not from a third-party adversary. The purpose of non-repudiation is to be guaranteed that the sender cannot deny the sent message.

Cryptography provides several protocols for users to fulfill all these security requirements. These protocols are actually a collection of various algorithms and softwares. Some of these are executed by the sender and some of them by the receiver. As a result, it is intended to provide a reliable and confidential data stream.

The encryption models used today can be grouped under two main headings: symmetric key ciphers and asymmetric key ciphers. The cryptography literature often uses the name Alice (A) for the sender, Bob (B) for the intended receiver, and Eve (E; eavesdropper) for the adversary. While explaining these models we will also follow this usage in the literature for easier understanding.

2.2.1 Symmetric Key Ciphers

The phenomenon that gives its name to the symmetric key ciphers is that the sender and the receiver use the same key for encryption and decryption. When evaluating the security of the encryption method, it is assumed that the key value is not and cannot known by the attacker. Except this, the Kerckhoffs’s Principle applies; every step of the cipher algorithm is open to the attacker.

Let us assume that the two users, Alice and Bob, want to transfer data in a secure way. They agree on a common key value that Eve does not know, by using a secure channel before they start transmitting data. This key value must be a uniformly distributed random collection of characters. The structure of this encryption model is given in Figure 2.1. In the figure, P is plaintext, C is ciphertext, E is encryption algorithm, D

is decryption algorithm, and *K* is the secret key. AES algorithm is one of the most common symmetric key cipher.

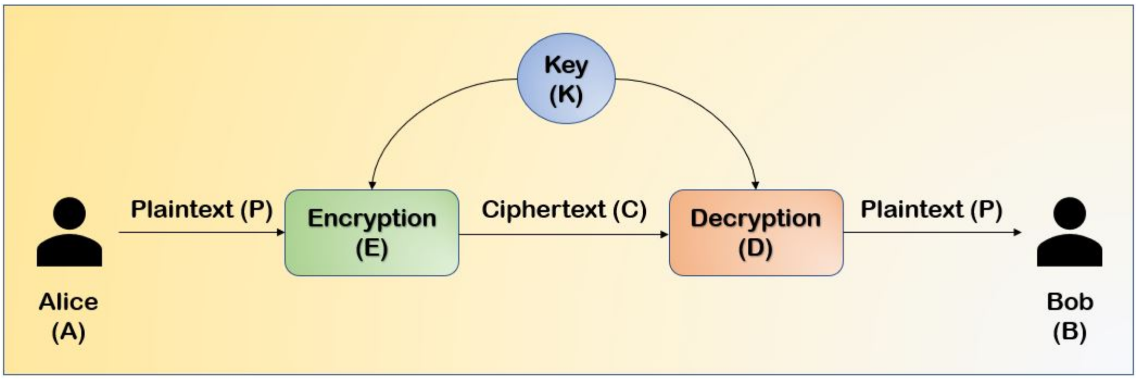

Figure 2.1 Diagram of secret-key or symmetric ciphers

A major problem in symmetric ciphers is the decision process of the secret key that be used by Alice and Bob. To agree on the key, Alice and Bob must begin by communicating with each other before using symmetric encryption. Therefore, this process cannot be protected by the symmetric cipher. This fundamental difficulty is known as the “key distribution problem” [14].

There are several ways to solve this key distribution problem like sending the key accompanied with armed guards before each message is transmitted, but of course this will not be a practical solution. A more practical approach is to evaluate alternatives to the symmetric key procedure and asymmetric key ciphers can be one of them.

2.2.2 Asymmetric Key Ciphers

The main question that asked to solve the key distribution problem is: can the receiver and sender parties encrypt and decrypt data using two independent keys, so that neither of them needs to know the other’s key?

Asymmetric encryption systems are intended to make this possible. Asymmetric encryption methods can be explained as follows: Bob has two keys, private key and public key. Public key can be known by everyone including the adversary, but private key is known only by Bob. When Alice wants to send a message to Bob, she encrypts it with Bob’s public key. And Bob decrypts the encrypted message with his private key. So, Eve can send a message to Bob, but she can’t decipher Alice’s message without she gets to Bob’s private key. This model also known as “public key cryptography”.

The structure of this encryption model is given in Figure 2.2. In the figure, *P* is plaintext, *C* is ciphertext, *E* is encryption algorithm, *D* is decryption algorithm, K_s is private key and K_p is public key. RSA algorithm is one of the most common asymmetric key cipher.

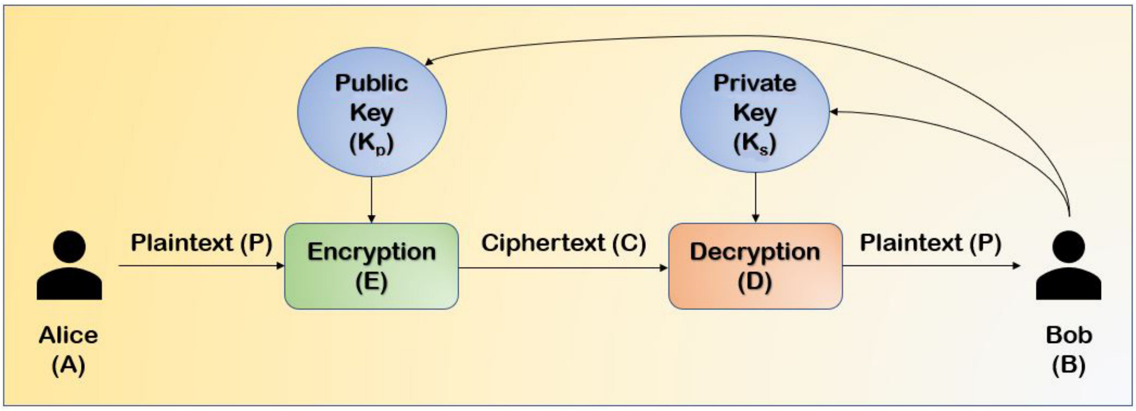

Figure 2.2 Diagram of public-key or asymmetric ciphers

Whether asymmetrical or asymmetric, there are some conditions that a safe cryptographic model must provide. The first and the most important one is the elimination of the attacker's effect only through the designed protocols. All other methods that can be used for this purpose are not interested with the cryptography science.

The second most important requirement is that there is no secret protocol in the proposed cipher. There are several different reasons for this. The first reason is, if the proposed protocol is intended to be used in many different platforms, to try to hide the protocol definition is not only unreasonable, but also very costly. Furthermore, hiding protocol design prevents the control of the proposed cryptographic system's reliability. It is assumed that there is nothing hidden other than key in a reliable and secure cryptosystem. The reason for this distinction is that the protocols are algorithms and the key is the data used in the algorithm.

2.2.3 Complexity Theory

Modern cryptography is heavily based on mathematical theory and computer science practices. Cryptographic algorithms are designed on the basis of calculation hardness assumptions; thus, it is hard to break in practice by any adversary. It is theoretically possible to break such a system, but it is infeasible to do so by any known practical means. These systems are called computationally secure.

If the maximum computational power available to the attacker is less than the minimum computational power required to break the cryptography system, then it would be pointless to perform an attack. In order to make this evaluation, a comparison of the source to be obtained by the attacker and the resource needed to break the system should also be done.

Cryptography must be interactive with the complexity theory to ensure this requirement. Complexity theory is an interdisciplinary theory that grew out of systems theory in the 1960s. Computational complexity theory evaluated under computer science focuses on classifying computational problems according to their inherent difficulty, and relating the resulting complexity classes to each other.

Complexity theory is rooted in chaos theory and uses complex systems. A complex system is the system in which the entire behavior of the system cannot be predicted even if its parts are perfectly understood because of the dependencies, relationships, or other interactions between their parts.

2.2.4 Principles of Shannon

In 1945, a Ph.D. mathematician who was not even 30 years old, published a paper called “A Mathematical Theory of Cryptography” [15]. The author of this article, which would later form the basis of cryptography was Claude Shannon.

Claude Elwood Shannon was an American mathematician, electrical engineer, and cryptographer known as “the father of information theory”. He is best known for his landmark paper that having founded information theory, “A Mathematical Theory of Communication” published in 1948. The most important contributions of him to the field of cryptography is the article in 1945 and the “Communication Theory of Secrecy Systems” published in 1949 [16]. In these two articles, Shannon has revealed two basic principles for a mathematical cryptographic method to be safe against statistical and other attacks: confusion and diffusion.

Confusion means that each bit of encrypted text is linked to more than one different part of the key. So even if a single bit of the key changes, the change in encrypted text will be in all parts of the text, not in a single region. The fulfillment of the confusion condition makes it hard to find the key even if there are too many plain text - encrypted text pairs to obtain with the same key. Because changing a bit of the key causes a homogeneous change in all of the encrypted text, and all resulting encrypted texts are statistically similar. Thus, it becomes difficult to capture a relationship between the key and the encrypted text.

Diffusion is that even the change in a single bit of plain text can be spread all over the encrypted text, or vice versa. More precisely, when a bit changes in the plain text, half of the bits in the encrypted text must be changed. The input bits and the output bits must be connected to each other in a very complex manner so that a cipher provides the diffusion property. In a well-diffused cipher, when a randomly selected input bit changes, the probability of change of any bit in the output is half a half. This condition is called the "Strict Avalanche Criterion" (SAC). It is the formalized version of "Avalanche Effect" and it builds in Webster and Tavares in 1985 [17]. "Avalanche Effect" term was first used by Horst Feistel who is a German-born cryptographer, in 1973 [18] and that means wherein if an input is changed slightly, the output changes significantly. It is the desirable property of cryptographic algorithms, typically block ciphers and cryptographic hash functions.

In the original definitions of Shannon in two articles, confusion is to make the relationship between key and cryptic text as complex as possible; diffusion is that changes in the statistics of plain text affect all of the statistics of the encrypted text. The simplest way to achieve these two conditions is a substitution-permutation system. In these algorithms, the plaintext and the key have a very similar role in producing the cipher text, therefore the same mechanism ensures both diffusion and confusion.

2.3 Fundamentals of Chaotic Cryptography

We have already mentioned that many cryptography methods have been proposed and broken from the invention of the writing up to this day. The development of the computer and the invention of the Internet also increased the need for secure information transfer. Powerful, practical and secure encryption protocols are proposed and used at the present time, but the needs are not fully met. Computer scientists continue to work in many areas to fill these gaps. And chaotic cryptography is one of these areas.

2.3.1 Chaos Theory

The origin of the term chaos is based on the ancient Greek word "khaos" ($\chi\alpha o\varsigma$). In the Greek creation myths, "khaos" is the name of a "gap" that refers to the void state preceding the creation of the cosmos or to the initial emptiness created by the separation of heaven and earth. So, in a way, "khaos" is the origin of the universe which carries all the potential that can compose it. Use of chaos in the meaning of "disorder" first appears in Elizabethan Early Modern English. "Chaos" in the sense of confuse system is in turn derived from this usage.

For both meanings, it is possible to say that a very suitable name is given to the theory of chaos. Because the behavior modeled by chaos theory is the most fundamental phenomenon that can be observed all over the known cosmos. In this respect, it is possible to say that chaos is the root of the universe. On the other hand, chaotic behavior is apparently complex. It does not follow a specific order or does not repeat itself. Hence, maybe it was given the most appropriate name to its nature.

Chaos theory is a branch of mathematics that focuses on the behavior of dynamic systems that are highly sensitive to initial conditions. In these systems, the very small differences in initial conditions can have very different results. The behavior of chaotic systems is strictly dependent on their initial conditions and there are no random items; hence it is deterministic. But although they are deterministic, their behaviors are not predictable. The theory was summarized by Edward Lorenz as [19]:

"Chaos: When the present determines the future, but the approximate present does not approximately determine the future."

Chaos theory began in the field of ergodic theory. The first proponent of it was French mathematician, theoretical physicist, engineer, and philosopher of science, Henri Poincaré. In the 1880s, while studying the three-body problem, he found that there can be orbits that are non-periodic, and yet not forever increasing nor approaching a fixed point. In 1898, Jacques Hadamard published a study of a free particle's motion that move on a frictionless surface of constant negative curvature. Hadamard showed that all trajectories are unstable and diverge exponentially from each other, with a positive Lyapunov exponent. Later this system was called "Hadamard's billiards" and it was the first dynamical system to be proven chaotic.

Subsequent works were on the topic of nonlinear differential equations. These studies were all directly inspired by physics like the three-body problem, turbulence, astronomical phenomenon's and radio engineering. Despite these studies, chaos theory became formalized as such only after middle of the twentieth century. The main thing that the development of chaos theory owed was the electronic computers. Electronic computers made repeated calculations practical, while figures and images made it possible to visualize these systems.

The American mathematician and meteorologist Edward Norton Lorenz was the pioneer of Chaos theory in 1963 with his paper on the feasibility of very-long-range weather predictions [20]. Lorenz is also the introducer of the notion of strange attractors and the creator of the term butterfly effect.

Edward Lorenz's interest on chaos began with a small mistake he made in his studies on weather forecasting in 1961. He was using a simple digital computer to run his weather simulation. He wanted to see a sequence of data again but also wanted to save the time. In order to achieve this, he entered the data that corresponds to the results in the middle of the simulation to the system as initial conditions. The difference was tiny, and it should have no practical effect. However, to his surprise, the weather predictions was completely different from the previous calculation. Thus, Lorenz discovered that small changes in initial conditions produced large changes in long-term outcome. Lorenz's discovery that called "Lorenz attractors", showed that even detailed atmospheric modelling cannot make precise long-term weather predictions.

The studies on chaos gained momentum after this date. The Polish-born, French and American mathematician and polymath Benoît Mandelbrot started to work in the field of chaos with the article he published on the change in cotton prices in 1963 [21] and introduced the two important term "Noah effect" and "Joseph effect" to the scientific world. His article named "How Long Is the Coast of Britain?" [22] that published in 1967 revealed the important phenomenon in chaos; self-similarity. In this article he showed that a coastline's length varies with the scale of the measuring instrument, resembles itself at all scales, and is infinite in length for an infinitesimally small measuring device. In 1975, he would call this self-similarity as "fractal". In 1982, Mandelbrot published "The Fractal Geometry of Nature" [23], which became a classic of chaos theory, and he proved that many biological systems fit a fractal model.

In 1977, the New York Academy of Sciences organized the first symposium on chaos, attended by David Ruelle, Robert May, Robert Shaw, Edward Lorenz and James A. Yorke who is coiner of the term "chaos" as used in mathematics. The following year, independently Pierre Coullet-Charles Tresser, and Mitchell Feigenbaum described logistic maps. They notably discovered the universality in chaos, permitting the application of chaos theory to many different phenomena.

Chaotic behavior can be observed in many natural systems from macro levels to micro levels. In fact, many systems with artificial components occur spontaneously. These behaviors can be predicted for a while, then "randomness" occurs. This predictable duration is called the "Lyapunov time" that named after the Russian mathematician Aleksandr Lyapunov. Its length depends on three things: how much uncertainty can be tolerated in the forecast, how accurately the current state can be measured, and the time scale depending on the dynamics of the system. Lyapunov times of some systems are as follows: the solar system, 5 million years; weather systems, a few days; chemical chaotic oscillations, 5.4 minutes; 1 cubic cm of argon at room temperature, 3.7×10-11 seconds.

In chaotic systems, the uncertainty in a forecast increases exponentially depending on time. Therefore, mathematically, doubling the forecast time increases more than squares the uncertainty in the forecast. In practice, this means that, over an interval of more than two or three times the Lyapunov time, a meaningful prediction cannot be made. And when no meaningful prediction can be made, the system appears random.

The word chaos is often used in the sense of “confusion and disorder”, as in the Elizabethan Early Modern English. Although there is no universal definition of chaos in mathematics, to call that “chaotic” to a system as mathematically, some conditions must be proved. According to Carmen and Ricardo [24] these are:

- **Dynamic instability:** It describes how a small change in initial conditions of a chaotic system can result in large differences in a later state.
- **Topologically mixing:** It means that the system evolves over time. So that any given region of its phase space eventually overlaps with any other given region.
- **Aperiodicity:** The system evolves in an orbit that never repeats on itself.
- **Dense periodic orbits:** To have dense periodic orbits means that system follows a dynamic that can arbitrarily closely approach every possible asymptotic state.
- **Ergodicity:** In a chaotic system, the dynamics shows similar statistics when measured over time or space.
- **Self-similarity:** The evolution of the system shows the same appearance at different scales of observation in time or space.

Sensitivity to initial conditions is popularly known as the “butterfly effect”, because of the title of a paper given by Edward Lorenz in 1972; “Predictability: Does the Flap of a Butterfly’s Wings in Brazil set off a Tornado in Texas?” [25]. The butterfly flapping its wings represents a small change in the initial condition of the system, which causes a chain of events that be caused to explicit difference in a large-scale phenomenon.

Poincaré briefly explains this situation as follows: *“A very small cause, which escapes us, determines a considerable effect which we cannot ignore, and we then say that this effect is due to chance”* [26].

The consequence of sensitivity to initial conditions, when we start with a limited data about the system, then after a certain time the system is no longer predictable. In more mathematical terms, the Lyapunov exponent measures the sensitivity to initial conditions. The Maximal Lyapunov Exponent (MLE) determines the overall predictability of the system. A positive MLE is taken as an indication that the system is chaotic.

However, just sensitiveness on initial conditions does not give chaos. For example, let’s think a simple dynamical system that repeatedly doubling a starting value. It is sensitive to initial conditions, because any pair of nearby points in it eventually becomes widely separated. But its behavior is not chaotic; all points except zero tend to positive or negative infinity.

To be able to call a system chaotic, it must be topological mixing. If the system is a topological mixing or topological transitivity, it is only a matter of time the system conflicts with any phase region for any initial condition. The mathematical concept of “mixing” appears in ergodic theory. According to probability theory, an ergodic dynamical system has the same statistical measurements with averaged of all the system’s states in its phase space.

Finite-dimensional linear systems are never chaotic; for a dynamical system to display chaotic behavior, it must be either nonlinear or infinite-dimensional. One could define the term chaotic system as a nonlinear dynamical system that have at least a chaotic strange attractor [24].

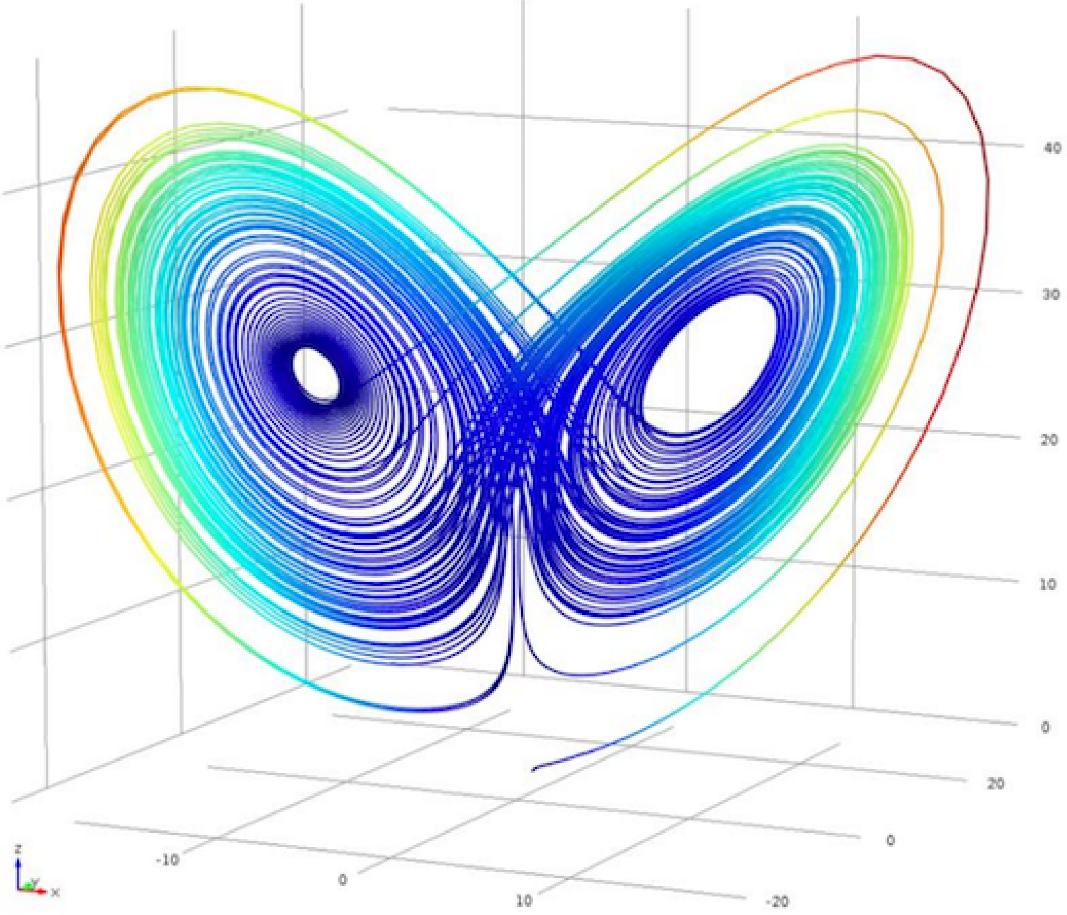

Figure 2.3 Lorenz attractor

Some dynamical systems like the logistic map are chaotic everywhere, but in many cases chaotic behavior is found only in a subset of phase space. When the chaotic behavior takes place on an attractor, a large set of initial conditions leads to orbits that converge to this chaotic region. The Lorenz attractor that can be seen in Figure 2.3, is perhaps one of the best-known strange attractor diagrams; because it was not only one of the first and most complex, but also has a very interesting pattern that looks like the wings of a butterfly.

Strange attractors occur in both continuous dynamical systems such as the Lorenz system and in some discrete systems like the Hénon map. Other discrete dynamical systems have a repelling structure called a Julia set. Both strange attractors and Julia sets typically have a fractal structure. Apart from those mentioned below, Arnold's cat map, Horseshoe map, Rössler attractor, Duffing equation and Standard map can be given as examples to chaotic systems.

Chaos is an interdisciplinary theory and has applications in various disciplines such as meteorology, anthropology, sociology, physics, environmental science, computer science, engineering, economics, biology, ecology and philosophy. The availability of cheaper, more powerful computers is broadening the applicability of it and currently, chaos theory is an active area of research, involving many different areas.

2.3.2 Relationship between Chaos and Cryptography

The fact that the internet is accessible from many different devices, anywhere and at any time, significantly increases the need for information security. The need to transfer more information faster and more secure forces cryptography to evolve. Therefore, researchers are constantly trying to propose, test and break new cryptographic methods. One of the branches of these works is chaotic cryptography.

The relationship between chaos and cryptography has been studied since the 1990s. But in fact, it is possible to find traces of this relationship even in Shannon's the cult article published in 1949 [16]:

"Good mixing transformations are often formed by repeated products of two simple non-commuting operations. Hopf has shown, for example, that pastry dough can be mixed by such a sequence of operations. The dough is first rolled out into a thin slab, then folded over, then rolled, and then folded again, etc."

The mixing process referred to by Shannon here is in fact to achieve chaos through stretching and folding that well known in today's chaos theory. After all, the dough example he given afterwards also is a chaotic system.

In recognition of this relationship, chaotic cryptography was also born. Since then, chaotic systems have been used in the design of hundreds of cryptographic protocols for many different aims like image encryption algorithms, hash functions, secure pseudo-random number generators, stream ciphers, watermarking and steganography.

The basic characteristics of the chaos and cryptography and relationships between them can be seen in Table 2.1 [27].

Table 2.1 Comparison between chaos and cryptography properties

Chaotic property	**Cryptographic property**	**Description**
Ergodicity	Confusion	The output has the same distribution for any input
Sensitivity to initial conditions / control parameter	Diffusion with a small change in the plaintext / secret key	A small deviation in the input can cause a large change at the output
Mixing property	Diffusion with a small change in one plain-block of the whole ciphertext	A small deviation in the local area can cause a large change in the whole
Deterministic dynamics	Deterministic pseudo-randomness	A deterministic process can cause a random-like (pseudo-random) behavior
Structure complexity	Algorithm (attack) complexity	A simple process has a very high complexity

Additionally, there are many advantages that chaotic cryptosystems may provide to cryptography. First of all, chaotic systems occur spontaneously in nature and can be directly applied to security processes, as it is the case of physical devices used in communications. Those show naturally non-linear and chaotic behaviors that can be straightforwardly used to secure communications. Also, chaotic non-linear dynamical systems have the advantage of being implemented with simple computable deterministic algorithms. Additionally, these algorithms may refer to N-dimensional equations or to several systems combined at a time. They could be subject of implantation with parallel computing and become faster algorithms [24].

There exist two main approaches of designing chaos-based cryptosystems; analog and digital.

Most analog chaos-based cryptosystems are secure communication schemes designed for noisy channels, based on the technique of chaos synchronization. In chaos-synchronization-based cryptosystems, the information can be transmitted by one or more chaotic signals in number of ways like chaotic masking, chaotic switching or chaos shift keying, chaotic modulation, chaos control methods, and inverse system approach [27].

In these systems which are mostly used to create a secure communication channel for sender and receiver, the plain text is masked with a chaotic signal at physical. The natural non-linearity of electric and optical communication devices is controlled to produce a chaotic waveform that modulates the message. At the receiver, the signal is demodulated with using chaotic synchronization techniques and produce the plain text. To provide a synchronous performance of sender and receiver, a chaotic control signal is used.

Because of chaotic signals are deterministic and intrinsically correlated although they appear to be random, analog systems are not completely secure. Therefore, patterns that may allow to decipher the message can be found in the communication signals. But their technical complexity and high transmission rates makes them useful for communications that require security for a limited time [24].

On the other hand, digital chaos-based cryptosystems or digital chaotic ciphers are the protocols that one or more chaotic maps are implemented in finite computing precision to encrypt the plain-message in a number of ways such as; stream ciphers that uses chaos-based pseudo-random number generators, chaotic stream ciphers via inverse system approach, block ciphers based on chaotic round function or S-Boxes, block ciphers based on forward/backward chaotic iterations, and chaotic ciphers based on searching plain-bits in a chaotic pseudo-random sequence [27].

Basically, digital chaotic ciphers are algorithms implemented in digital circuits. In these algorithms, after the production of chaotic series that are based in iterative computations of chaotic functions, basic cryptographic operations such as substitution and mixing are used to encrypt the plain text. Digital chaotic cryptosystems involve one or more chaotic systems and use the secret key as their initial conditions and/or control parameters.

One of the weaknesses of digital chaotic cryptosystems is that: chaos implemented on digital system might diverge from real-precision chaos because of computers has finite precision. This is the reason it is called “pseudo chaos”. It is possible to minimize their dynamical degradation of pseudo chaotic systems and using high dimensional chaotic systems might also be helpful. Still, only a detailed study of the dynamical system can guarantee their performance [24].

Another method of design a digital chaos-based cryptosystem is to use chaotic function as a pseudo-random number generator. Since 1990s, the proposals of chaos based PRNG have demonstrated a significant increase.

To build a chaotic PRNG, a deterministic discrete-time dynamical system and construct an algorithm that transforms the states of the system into binary numbers are used. The binary sequence generated by chaotic PRNG is used as a keystream and to encrypt the plain text, the keystream is added to it through a binary XOR operation. The secret key is used as initial conditions and the vector of parameters [24].

The details of these methods can be examined from Table 2.2 [24].

Table 2.2 Different kinds of chaos-based cryptosystems proposed in literature

Category	**Method**	**Description**	
Analog cryptosystems	Additive chaos masking	A chaotic signal is added to the message	
	Chaotic shift keying	A digital message signal switches among different chaotic systems to be added to the message	
	Chaotic modulation	A message signal is used to change the parameters or the phase space of the chaotic transmitter	
	Chaotic Control	A message signal is ciphered in a classical way and used to perturbate the chaotic system	
Digital cryptosystems	Stream ciphers	Chaotic PRNG	A chaotic signal generates a pseudorandom sequence (keystream) to XORed the message
		Chaotic Inverse System approach	A message signal is added to the output of the chaotic signal, which has been feeded by the ciphered message signal in previous instants
	Block ciphers	Backwards iterative	A block of a clear message is ciphered using of inverse chaotic systems
		Forwards iterative	A block of ciphered message is obtained by pseudoramdom permutations obtained from a chaotic system
		S-Boxes	An S-Box is created from the chaotic system. There can be dynamic or static S-Boxes
	Miscellaneous	Searching based chaotic ciphers	A table of characters is generated from a chaotic system. The table is used to cipher the characters of the message text
		Cellular Automata	The chaotic system is a Cellular Automata

2.3.3 Implementation of Chaos-Based Cryptosystems

In many papers on chaos-based cryptosystems, basic concepts are described but detailed practical issues are ignored. However, the encryption speed and the application cost depend on implementation details which are very important to appraise the security of a cryptosystem. Therefore, the lack of implementation details makes it difficult to evaluation of the reliability and importance of the proposed cryptosystem.

As mentioned before there are two basic approaches to the design of chaos-based cryptosystems: analog and digital. For an analog implementation, the detailed information about the circuitry responsible for chaos generation, at least the explicit form of the differential equation system should be given. For a digital implementation, the following details should be provided: the finite computing precision, the adopted digital arithmetic (fixed-point or floating-point), the hardware/software configuration, etc. [27]

Álvarez et al. have proposed the following three rules [27] for the digital implementation of chaotic systems:

- **Suggested Rule 1:** A thorough description of the implementation of the chaotic systems involved should be provided.
- **Suggested Rule 2:** For chaotic systems implemented in digital form, the negative effects of dynamical degradation should be taken into consideration with careful evaluation.
- **Suggested Rule 3:** Without loss of security, the cryptosystem should be easy to implement with acceptable cost and speed.

When chaotic systems are completely or partially implemented in digital form, dynamical degradation will occur, and the dynamical properties of digital chaotic systems may become non-ideal. The most well-known problem is the existence of many short-length chaotic orbits, which may weaken the desired statistical properties and lower the security of the ciphers. To overcome the problems like these, some methods should be used to improve the dynamical degradation of digital chaotic systems such as to timely perturb the underlying chaotic system with a small pseudo-random signal [27].

There is a well-known saying between cryptographs: “there is nothing easier than design a secure but very slow cipher or a secure but very large cipher”. Even if a digital chaotic cipher is extremely safe, it will be impossible to use it in practice if it lacks operating efficiency. Therefore, level of security, performance, and ease of implementation are the three main criteria to evaluate new cryptosystems.

In his Ph.D. thesis, S. Li make some basic suggestions that given for the design of fast and low-cost digital chaotic ciphers [28]:

- The simpler the employed chaotic system is, the simpler the realization will be and the smaller the cost will be.
- The fixed-point arithmetic is better than the floating-point one since the latter needs much more cost and computation complexity.
- For hardware implementations supporting parallel computation, (coupled or independent) multiple chaotic systems will be useful to promote the encryption speed dramatically and add complexity of possible attacks.
- Another desired requirement is the extensible security and accessional functions with considerably extra cost and complexity.

There may many other suggestions to design a powerful and efficient digital chaotic cipher. Each user group, each type of data to be encrypted and each platform to which the data will be transferred has its own characteristics. Therefore, the designer should be decided which solutions will be used according to the features and needs of the system.

3
Chaotic Image Encryption

Image encryption is one of the most appealing areas of chaotic cryptography. There are several reasons for this. Firstly, cryptographic image encryption algorithms have high randomness, unpredictability, sensitivity and topological transitivity. The second, chaos-based image encryption algorithms have shown some remarkably good properties in many concerned aspects like security, complexity, performance, speed. The third, methods that can be applied to use a chaotic system for image encryption are very diverse and they are easy to implement. A chaotic system in the image encryption algorithm can be used as a PRNG (pseudo-random number generator) or displacement map, or in generating a mask sequence or S-Box.

For all these reasons, researchers have proposed different methods to encrypt images with chaotic cryptography for years. As mentioned before, there are many ways to use chaotic systems in image encryption. Before proceeding to the method used in the proposed algorithm, we will give a few examples of various techniques.

One of the popular methods of design a chaotic image cipher is designing S-Box with using chaotic systems like Farwa et al. did [29]. They proposed an image cipher that utilizes a composition of chaotic substitution based on tent map with the scrambling effect of the Arnold transform. The proposed algorithm uses an S-box that is based on 1-D chaotic tent map. They partially encrypt the image using this S-box and then apply certain number of iterations of the Arnold transform to attain the fully encrypted image.

Choi et al. follows a different method to take advantage of chaotic systems [30]. Their proposed algorithm is also based on the lightweight and fast block cipher published by Hong et al., called LEA [31]. It uses simply three operations: modular addition, bitwise rotation, and bitwise XOR (ARX). Also, in the proposed scheme, the encryption process comprises three phases: XOR phase, round phase, and rotation phase. The XOR and rotation phases are used to satisfy the confusion property, while the round phase is used to satisfy the diffusion property. In the confusion process, two key sequences that used in rotation and XOR are calculated via two logistic maps using two key pairs.

Li et al. used a combination of Tent and Lorenz chaotic systems for image encryption [32]. First, they produce and normalize three Lorenz sequences. After this process, average of these sequences was used for generating the Tent sequence. And as the last step, an initial key was produced with using Tent sequence and Lorenz sequences and pixels enciphered with this key.

Another popular method of chaotic image encryption is to use fractals. For example, Rozouvan uses the fractals as key in the encryption and decryption process [33]. In this method, a unique image is generated on a fractal set like Mandelbrot or Julia with some parameters such as coordinate, iterations, zoom etc. obtained from the encryption key. Then the image to be encrypted with the parts taken from this unique image is subjected to modulo operation. Advantage of this method is the fractal key needs very small memory space because of a few numbers can represent a unique fractal image.

Sun et al. are following a similar method in their article [34]. They generate a Julia set and scramble it with the Hilbert curve in bit-level, and calculate modulo of this Julia set and plain image. Secret key used as parameters of Julia set, forward step and backward step along the Hilbert curve and diffusion keys.

Naturally, there are many more researchers who try different methods for designing a chaotic image cipher. But we do not need to examine all methods. Since an algorithm that uses chaotic displacement map and color space transformation is proposed in this thesis, similar encryption methods are examined in more detail in the following two headings.

3.1 Chaotic Maps in Image Encryption

One of the earliest examples of the use of chaotic maps for image encryption was given by Pichler and Scharinger [35]. Aim of their study was introduce a new product cipher which encrypts large blocks of plain-text such as images by repeated intertwined application of substitution and permutation operations. Their approach involves parameterizable permutations on large data-blocks induced by specific chaotic system that called Kolmogoroff-flow.

Jiri Fridrich took this approach a step further and did important works by many researchers working on image encryption by using chaotic maps [36], [37]. Therefore, we will explain how chaotic maps are used for image encryption by using Fridrich's article [36]. In his paper, Fridrich summarized the process of developing a chaos-based cipher can be as follows:

"First, a chaotic map is generalized by introducing parameters into the map. Geometrical arguments are often used at this stage. Then, the map is modified so that its domain and range are both the same square lattices of points (pixels, or some other general data items). The map is extended to three dimensions so that the values of the pixels (the gray levels) can be changed. A diffusion step is introduced by composing the generalized discretized map with a simple diffusion mechanism."

Although the methods used today do not follow the road map drawn by Fridrich, the steps he mentioned give important information about how chaotic maps can be used. But let's talk a little bit about chaotic maps before giving more details.

Chaotic maps are displacement maps that can be one or more dimensions. In these maps, the input and output of the chaotic system is considered to be a coordinate value. If the chaotic system equation or equations used have a single variable, then the chaotic map produced using these equations is one dimensional, and if it has two variables, it is two dimensional etc. For example, Logistic map and Tent map are 1-D, Arnold's cat map and Baker's map are 2-D, Lorenz and Lotka–Volterra are 3-D maps.

Chaotic maps can be used to mix the pixels of an image or to change their value. Fridrich worked on gray scale images and used Baker map in his study. But he generalized and discretized Baker map, and after that, it was extended to three dimensions with using an invertible function in order to mix the gray levels. So, with the Baker map, he changed not only the pixel positions, but also the values. This procedure used can be applied to any 2-D chaotic map. The resulting substitution cipher can create a random looking image with uniform histogram in only a few iterations [36], [37].

Fridrich's articles paved the way for more work using chaotic maps to encrypt images. The researchers tried to integrate chaotic maps into their encryption systems in different ways and eliminate their weaknesses.

Chen et al. tried to implement the idea of obtaining a three-dimensional map from two-dimensional map that set by Fridrich on the Arnold's cat map [38]. For the purpose of diffusion, XOR and modulo operation applied in the scheme, to each pixel in between every two adjacent rounds of the 3-D cat map and this method render the discretized chaotic map non-invertible.

Like a single chaotic map can be used for image encryption, more than one chaotic map can also be used in a sequential or alternate format. For example, Taneja et al. used Hénon and Arnold's Cat map in their study [39]. The suggested cryptosystem is based on permutation-substitution architecture, where the permutation operation is performed using Arnold cat map, while substitution is performed using Hénon map. The Arnold iterations and the initial condition of Hénon map generated form the security keys.

And some researchers preferred to use 3-D maps instead of 3-D maps generated from 2-D maps. Steffi and Sharma suggested an image cipher based on Baker map and Lorenz system [40]. At the confusion stage, the position of the pixels is scrambled over the entire image without disturbing the value of the pixels and the image becomes unrecognizable. And in the diffusion stage, the pixel values are modified sequentially by the sequence generated from one of the two chaotic systems selected by external key. The whole confusion-diffusion round repeats for several times. In both stages, one of the chaotic maps is selected and used according to the value in the secret key. Also, all of the initial conditions and control parameters generated from it.

As seen, it is not enough only confusing the pixels to design a secure image encryption system. The common feature of encryption systems that use a chaotic map for image encryption is to change the pixel values in addition to being replaced with these maps. For reach to this goal, different methods can be used. Generally, chaotic image encryption algorithms are hybrid systems where variety of techniques and multiple procedures are performed.

3.2 Image Ciphers Based on JPEG Algorithm

Although image compression techniques are not directly related to chaotic image encryption, since they can be used to strengthen the chaotic ciphers, they deserve the attention of researchers that study in this field. However, within the scope of this thesis, we will only examine the encryption studies based on the JPEG compression algorithm.

JPEG is a digital image encoding format standardized by the Joint Photographic Experts Group and the name of the algorithm is an acronym of it [41]. It is a complex compression method that can be applied to color or grayscale images lossy or lossless. It does not work very well in black and white images but works best on continuous-tone images, where adjacent pixels have similar colors. An important feature of JPEG is its use of many parameters, allowing the user to adjust the amount of the data lost and thus also the compression ratio over a very wide range. Often, the eye cannot see any image degradation even at high compression factors. There are two operating modes, lossy and lossless. Most implementations support just the lossy mode [42].

The JPEG compression algorithm includes various steps such as RGB into luminance/chrominance (YCbCr) color space conversion, downsampling process, Discrete Cosine Transform (DCT), quantization and entropy coding. According to Salomon et al. [42], the compression steps of the jpeg algorithm can be listed as follows:

- **Step 1:** Color images are transformed from RGB into a luminance/chrominance (YCbCr) color space. This step is optional but important. Without transforming the color space, none of the three color components will tolerate much loss, leading to worse compression.
- **Step 2:** Color images are downsampled by creating low-resolution pixels from the original ones. The downsampling is not done for the luminance component. This procedure will be explained in detail in section 4.2.
- **Step 3:** The pixels of each color component are organized in groups of 8×8 pixels called data units, and each data unit is compressed separately. If the number of image rows or columns is not a multiple of 8, the bottom row and the rightmost column are duplicated as many times as necessary. The fact that each data unit is compressed separately is one of the downsides of JPEG. If the user asks for maximum compression, the decompressed image may exhibit blocking artifacts due to differences between blocks.
- **Step 4:** The discrete cosine transform (DCT) is then applied to each data unit to create an 8×8 map of frequency components. They represent the average pixel value and successive higher-frequency changes within the group. Since DCT involves the transcendental function cosine, it must involve some loss of information due to the limited precision of computer arithmetic, but it is normally small.
- **Step 5:** Each of the 64 frequency components in a data unit is divided by a separate number called its quantization coefficient (QC), and then rounded to an integer. This is where information is irretrievably lost. In practice, most JPEG implementations use the QC tables recommended by the JPEG standard for the luminance and chrominance image components.
- **Step 6:** The 64 quantized frequency coefficients (which are now integers) of each data unit are encoded using a combination of RLE and Huffman coding.

An arithmetic coding variant known as the QM coder can optionally be used instead of Huffman coding.

- **Step 7:** The last step adds headers and all the required JPEG parameters to the output, and gives the result.

First and second steps are always skipped for grayscale images.

JPEG is currently one of the most common image compression algorithms in the world. Therefore, image encryption methods based on the JPEG algorithm are tried to be designed.

One of them is the algorithm proposed by Lian et al [43]. According to the method they suggest, the DCT blocks in luminance and chrominance plane are confused by pseudo-random SFCs (Space Filling Curves). In each DCT block, DCT coefficients are confused according to different frequency bands and their signs are encrypted by a chaotic stream cipher. They claim that, the algorithm is of high security and low cost and it supports direct bit-rate control or recompression, which means that the encrypted image can still be decrypted correctly even if its compression ratio has been changed.

In another proposed algorithm, the objective of Niu et al. is to keep the file size for JPEG image after encryption and do not affect the signal processing of JPEG [44]. DCT coefficients are composed of DC and AC. DC coefficient is the coefficient with zero frequency in both dimensions and represents the average color of the 8×8 region. AC coefficients are remaining 63 coefficients with non-zero frequencies and represent color change across the block. In the algorithm, the DC differential residues are encrypted through XOR with the key that is the same length with the data stream. After that, DCT blocks are scrambled using a key-controlled chaotic map. The information of pre-steps is encrypted by cipher and embedded in the second category of AC coefficients.

In their paper, Zhang et al. proposed two different ciphers that based on JPEG algorithm and compare them [45]. In the first algorithm, two one-dimensional chaotic sequences generated by arbitrarily selecting two sets of initialization are used to scramble row and column positions of all the data of a two-dimensional data matrix. The two one-dimensional chaotic sequences combine together, and scramble the row and column positions of all the 8×8 data blocks of three two-dimensional data matrices of the color image. Because every two-dimensional data matrix of three two-dimensional data matrices is scrambled by two chaotic sequences in the same way, the color of every pixel of the encrypted image is the same as the one of the every corresponding pixel of the original image.

In the second algorithm they suggested, arbitrarily select six sets of initialization and respectively generate six one-dimensional chaotic sequences, every two sequences combine, in this way, three groups of assembled sequences respectively scramble the row and column positions of all the 8×8 data blocks of three two-dimensional data matrices of the original image. The row and column positions of all the 8×8 data blocks of very two-dimensional data matrix corresponding to every component of the color image is scrambled by a group of assembled sequences respectively, therefore, the color of every pixel of the encrypted image is reset, the encrypted image looks disorganized, the encrypting effect is better. As a result of the analysis, Zhang et al. observed that the second algorithm was more successful.

Although the use of encryption methods based on JPEG algorithm seems to be narrower than their equivalent, some of the suggested ideas can be used to develop image encryption algorithms.

While there are several advantages and disadvantages to all the chaotic image ciphers, very few have been fully capable of security. Efforts to find a secure chaos-based encryption method for image encryption is still ongoing.

4
Proposed Algorithm

The proposed algorithm in this thesis is a chaotic map-based image cipher with using color space transform. The purpose of this study is to develop a method that can withstand plain text attack that is the weakness of a large proportion of chaotic image encryption algorithms. But before going into the details of it, let us give some information about the methods used in the algorithm.

4.1 Lorenz System and Lorenz Map

As stated in Section 2.3.1, Edward Lorenz was the person who is a pioneer of chaos theory and brings the “strange attractor” notion and term “butterfly effect” to the world of mathematics. At the origin of all these studies, there was a simplified mathematical model of atmospheric convection that developed by him in 1963. This model was the first chaotic system; it has chaotic solutions for certain parameter values and initial conditions. In particular, the “Lorenz Attractor” is a set of chaotic solutions of it, and when plotted it resembles a butterfly or figure eight (Figure 2.3).

The mathematical model of Edward Lorenz is a system of three ordinary differential equations now known as the “Lorenz Equations” that given below:

$$x' = \sigma(y - x) \tag{4.1}$$

$$y' = \rho x - y - xz \tag{4.2}$$

$$z' = xy - \beta z \tag{4.3}$$

In fact, these simple-looking equations (1-3) relate the properties of a two-dimensional fluid layer uniformly warmed from below and cooled from above. The three variables x, y, and z are respectively proportional to the rate of convection, the horizontal temperature variation, and the vertical temperature variation. The constants σ, ρ, and β are system parameters proportional to the Prandtl number, Rayleigh number, and certain physical dimensions of the layer itself. The Lorenz model arise not only in atmospheric convection, but also in simplified models for lasers, dynamos, thermosyphons, brushless DC motors, electric circuits, chemical reactions and forward osmosis.

From a technical standpoint, the Lorenz system is nonlinear, non-periodic, three-dimensional and deterministic. But most importantly, when $\sigma = 10$, $\rho = 28$ and $\beta = 8/3$, it exhibits chaotic behavior for these (and nearby) values. The famous Lorenz attractor are thus also revealed.

As it seen, the Lorenz system takes three inputs and produces three outputs with the solution of the equations (1-3). This data corresponds to the coordinates of a three-dimensional coordinate system. When constants are in their chaotic values, these coordinate relationships would create a 3-D chaotic map at the end of the process.

4.2 YCbCr Color Space and Downsampling

The YCbCr color system is a color space especially used in video encoding and is a scaled and offset version of the YUV color space.

YCbCr color space is defined by a mathematical coordinate transformation from an associated RGB color space. In this color system, Y is the luma component, Cb and Cr are the blue-difference and red-difference chroma components. Y in the YCbCr color system is different from Y value in the YUV color system which is luminance, meaning that light intensity is nonlinearly encoded based on gamma corrected RGB primaries. There are several YCbCr sampling formats, such as 4:4:4 (default), 4:2:2, 4:1:1 and 4:2:0. These are downsampled versions of 4:4:4 format [46].

The transformation equations between RGB and YCbCr color spaces are as follows:

$$Y = 0.299R + 0.587G + 0.114B \tag{4.4}$$

$$Cb = -0.172R - 0.339G + 0.511B + 128 \tag{4.5}$$

$$Cr = 0.511R - 0.428G - 0.083B + 128 \tag{4.6}$$

The RGB to YCbCr color conversion is also used in JPEG compression as a preparation to compress the data.

The downsampling operation is a compression process with assuming that the Cb and Cr values of the pixel blocks are the same. The reason of choosing Cb and Cr values for compression is the fact that these values contain less critical information for the pixel color. The human eye is sensitive to small changes in luminance but not in chrominance, so the chrominance part can highly compressed, without visually impairing the overall image quality much and since the luminance component is not touched, there is no noticeable loss of image quality [42]. Figure 4.1 shows a diagram about downsampling.

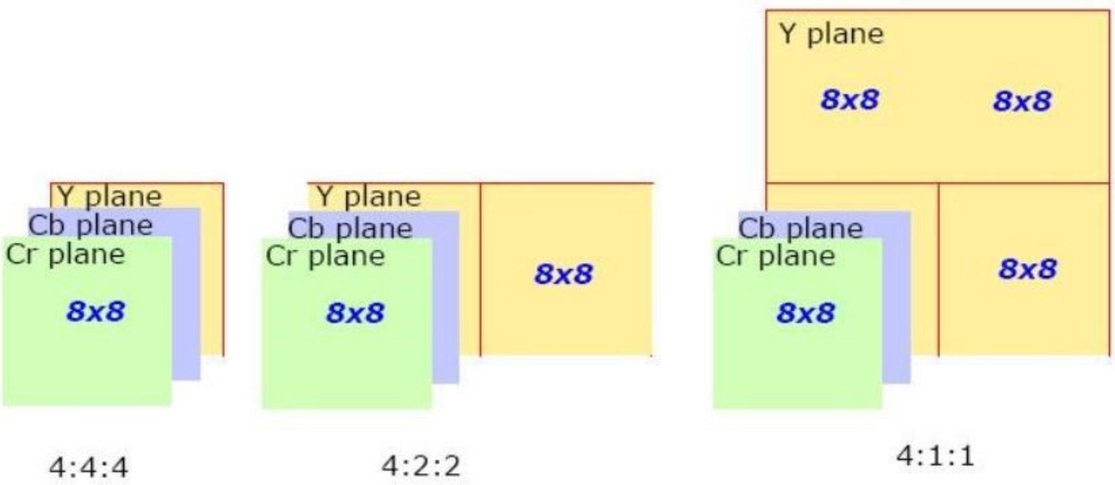

Figure 4.1 Diagram of different downsampling types

In downsampling process, the Cb and Cr values of each pixel at the same place in the two adjoining pixel blocks are averaged. In this way, the size of the Cb and Cr values is halved. Then this average value is written instead of both values in both pixels. Although there is some loss of data, it is possible to get a result that is very close to the original picture after the inverse conversion to YCbCr to RGB.

4.3 How the Algorithm Works

The proposed algorithm in this thesis is based on the Lorenz equations and the color space transformation. The chaotic displacement map obtained from the Lorenz equations was used in mixing operations, and the color space transformation was used to change the pixel values.

The idea of changing the color space was inspired by the JPEG compression algorithm. The algorithm proposed in this thesis uses the steps of RGB into YCbCr color space transform and downsampling process.

The algorithm consists of two parts; encryption and decryption. The decryption process is essentially the opposite of the encryption process. Therefore, the encryption process will be explained in detail and the decryption process will be explained with reference to these explanations.

4.3.1 Using the Key

The algorithm uses a 128-bit key. From this key three double values for constants of Lorenz equations and one boolean value for downsampling direction are generated. For this purpose, the key is divided into three 42-bits and two 1-bit parts. Figure 4.2 shows how the key is fragmented.

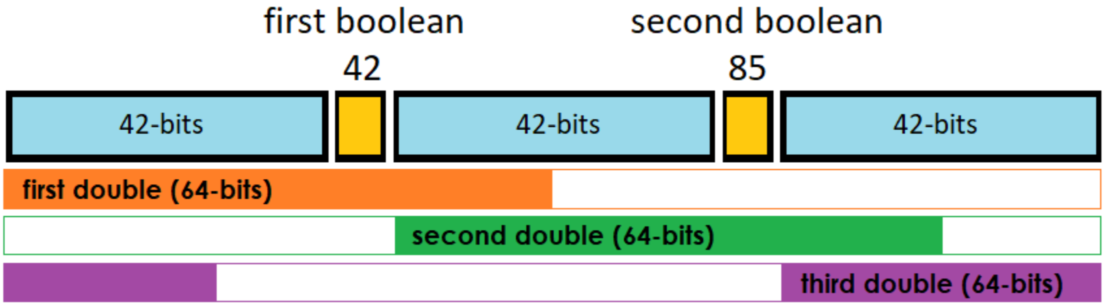

Figure 4.2 Use of secret key bits

First double value started from first bit and ended with 63rd bit, second double started from 43rd bit and ended with 106th bit, third double started from 86th bit and ended with 21st bit. The two single-bit pieces that are 42nd and 85th bites also seen in the figure are XORed and stored as the downsampling direction value.

Each of the three double numbers obtained from these 64-bit bit-array pieces are added to chaotic sigma, beta and rho values that 10, 8/3 and 28 respectively, and saved for use in the Lorenz equations as constants. For this aim, "0," is added to the beginning of the three double numbers obtained with 64 bits, for making them smaller than 1. Thus, sigma, beta and rho numbers to be used in encryption are close to 10, 8/3 and 28 values. Thus, the Lorenz map to be obtained is guaranteed to be chaotic.

The following pseudocode can be used to examine how to calculation of sigma, beta and rho values is done:

```
for (int i = 0; i < 64; i++){
sigmaBits.Set(i, keyBits.Get(i));
betaBits.Set(i, keyBits.Get(i + 43));
rhoBits.Set(i, keyBits.Get((i + 86) % 128));
}
direction = keyBits.Get(42) ∧ keyBits.Get(85);
rhoBits.Set(63, direction);
//tempInt: a positive integer that extracted from the encryption key
//reduction number
cryptexData.ReductionIteration = 16 + (4 * (tempInt % 5));
```

```
//lorenz iteration number
cryptexData.LorenzIteration = 4 + (tempInt % 5);
//suffle number
cryptexData.SuffleIteration =
cryptexData.ReductionIteration / 8;
```

As can be seen, the ReductionIteration and LorenzIteration numbers can be at least 16 and 4 and the maximum is 32 and 8. These numbers are the optimal iteration values that chosen according to the results obtained from the tests. The processes that these variables are used is discussed in the following sections.

The use of the key is the same for encryption and decryption operations. In the experiments, the AES algorithm key generating methods in the standard C# library were used.

4.3.2 Encryption Process

The encryption section is the most processing part of the algorithm and has a lot of different layer. In this section, all process steps will be explained in detail with separate subheadings. Additionally, the flow-chart diagram of the encryption part can be examined in Figure 4.3.

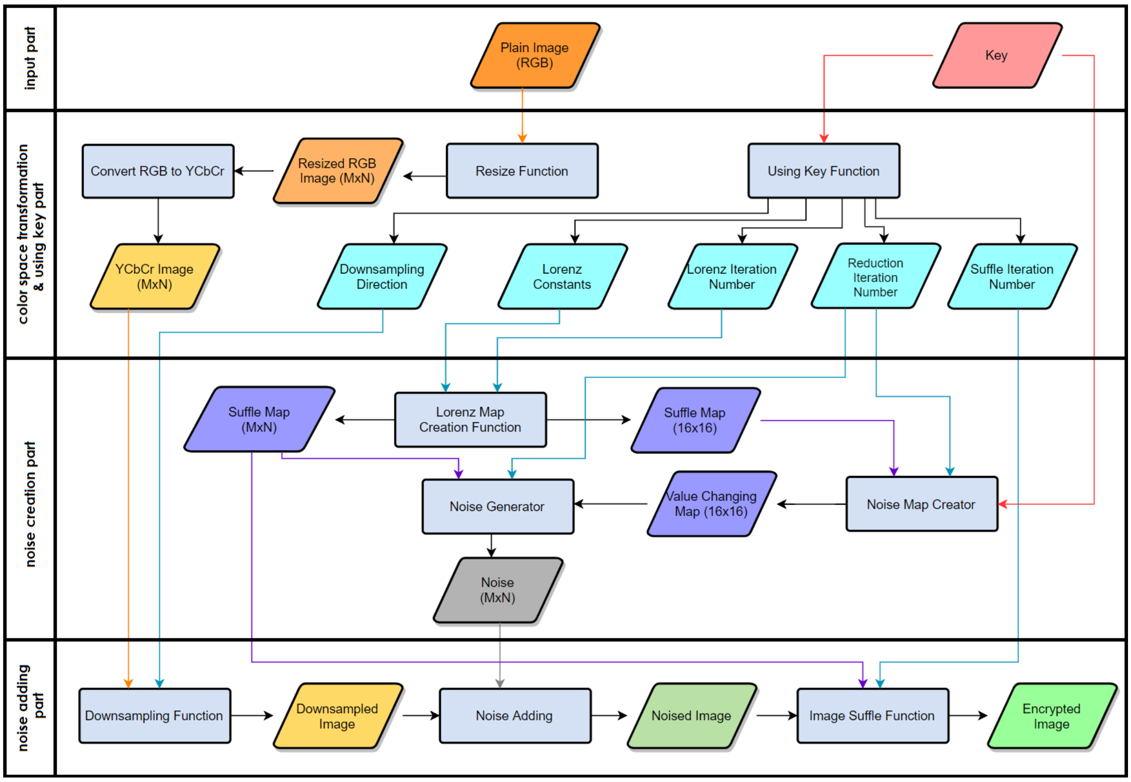

Figure 4.3 The flow-chart diagram of the encryption part

As mentioned earlier, the proposed algorithm uses a 3-D displacement map derived which is obtained by solving the Lorenz equations for pixel mixing. But just changing of pixels location is not enough to satisfy the security requirements. To make the algorithm safe enough, it is necessary to also change the pixel values.

Even if these two conditions are met, an image encryption algorithm may not resist some attacks. Plaintext attack is one of them and many image encryption methods can be broken by it. In this thesis, it was aimed to design an algorithm resistant to plaintext attack.

4.3.2.1 Creating the Lorenz Map

Because of producing Lorenz map step forms the chaotic part of the algorithm, this process is particularly important. But as in all chaotic cryptography algorithms, the pseudo chaos caused by the conversion of irregular numbers into integers is an important problem in this algorithm. Furthermore, in order to obtain a Lorenz map, the results obtained from the Lorenz equations (4.1-4.3) must be in a specific range and correspond to a pixel position.

The biggest problem in obtaining a Lorenz map is that the multiple solutions of Lorenz equations show the same pixel. Therefore, repetitive values must be rearranged to refer to different pixel positions. While designing the algorithm, different ways to solve this problem have been tried, and it is decided in the following way in order not to decrease performance.

While the Lorenz equations are solved once for all pixel values, the repetitive values are stored in memory and assigned to the remaining pixels after the solution is completed. To reduce the effect of this placement, the map creation process was repeated more than once using the previous Lorenz map. The `LorenzIteration` variable is the value that indicates how many times this procedure is repeated.

4.3.2.2 Color Space Conversion

The first step of the proposed algorithm is to convert the plain image from RGB color space to YCbCr color space. The equations in (4.4-4.6) are used for this conversion.

RGB to YCbCr color space transformation is used to prepare the ground for compression in the downsampling process. In this respect, the proposed algorithm has a common direction with the JPEG algorithm. The advantage of this translation is that Cb and Cr values contain less important information than the Y value. Thus, downsampling can be performed on Cb and Cr values, but data loss can be minimized when converted to RGB space again.

4.3.2.3 Downsampling Process

The downsampling process is also inspired by the JPEG algorithm but there are some differences. Although conventional downsampling is applied to pixel blocks, the proposed algorithm is applied vertically or horizontally on a pixel basis to minimize data loss. After this process, the Cb and Cr values of the pixels next to each other are the same and the Y values are preserved.

Because of the downsampling operation is applied to image on pixel based, the resulting data loss is less than the eye can perceive. The direction is selected according to the key. In the Section 4.3.1, we mentioned that a boolean direction value obtained by XORing the two bits of the key is stored for downsampling. If this boolean value is true then downsampling is done in the vertical direction, if its false then it's done in the horizontal.

Additionally, since the downsampling process can cause problems with the images that have width and height values as odd numbers; before the start of this process, if necessary, the image is cut off at its edges, so its sizes become even numbers.

The following pseudocode can be used to examine how vertical downsampling is done:

```
for (int i = 0; i < Width; i++){
    for (int j = 0; j < Height; j += 2){
        Set pixel1 to inputImage(i, j);
        Set pixel2 to inputImage(i, j + 1);
        cb = (pixel1.G + pixel2.G) / 2;
        cr = (pixel1.B + pixel2.B) / 2;
        Set pixel3 with color values pixel1.R, cb, cr;
        Set pixel4 with color values pixel1.R, 0, 0;
        Set newImage(i, j) to pixel3;
        Set newImage(i, j + 1) to pixel4;
    }
}
```

As it seen, in the downsampling process, the `cb` and `cr` values of some pixels are left as 0, which is shown as `pixel4` in pseudocode. These zero values will be filled using the noise value in next step.

4.3.2.4 Noise Adding

After these two steps, some noise is added to the Cb and Cr values of the one of the downsampled pixels. The added noise can be any picture or value. But the noise used in the proposed algorithm is produced from plain image.

To produce this noise, each color value that the original image can retrieve is reduced using a map. The map used is made of key. Each byte value in the key is converted to an unsigned integer value between 0-256 and is written to a 16×16 matrix in a loop format. This matrix is then mixed using a 16×16 Lorenz map and then transformed into vector. The resulting vector is the reduction map to be used. The number of times this reduction map is applied to the image is stored in the `ReductionIteration` variable.

After the map is generated, the R, G and B values of each pixel in the original image are replaced with their counterparts in this map. In this way, noise is also produced. The values that left in 0 in the downsampling process are replaced by the values of noise pixels in same location. Thus, noise adding step is completed and the pixel values are changed.

Noise insertion is the basic procedure that will make the proposed algorithm strong against plain text attack. Even if the attacker chooses a completely 0 plaintext, the resulting image will not give Lorenz map or noise, so an encrypted picture cannot be broken by removing the chaotic map.

Because there are repetitive values in map, it is also difficult to determine which map value corresponds to which color value in the image. Moreover, while generating noise, map and mixing operations are repeated more than once. It means that, even the plaintext is completely black or white, it can be transformed into a complex color vortex. The number of times the process is repeated also depends on the key.

In the Figure 4.4, these three steps of algorithm summarized.

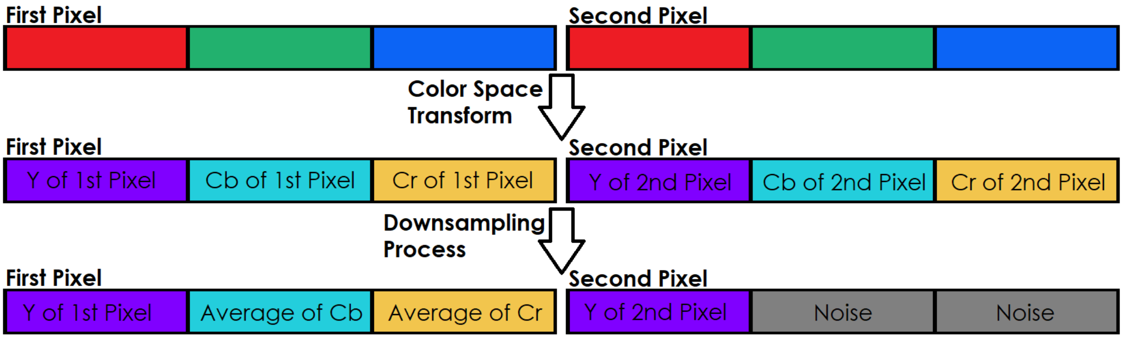

Figure 4.4 First three steps of algorithm in pixel level

4.3.2.5 Pixel Mixing

The noise added picture is mixed several times using a Lorenz map. The number of times to be mixed depends on the key and this number is kept in the `SuffleIteration` variable.

In the proposed algorithm, the mixing process is done on bit basis, not on pixel basis. Because of size of each pixel value is 3 bytes, the size of the chaotic map to be used for an image of M×N dimensions is M×N×24. This is the step of the algorithm that needs the most time and memory. Repeating calculation of the Lorenz equations repeatedly increases processing time. As a result, the processing time is proportional to the size of the plain image.

4.3.3 Decryption Process

The same steps are followed in the decryption process except one part. The noise is completely ignored in the decryption, and the color space conversion is performed after noise values on the image has been deleted. This image is the same image obtained after the downsampling process.

Unfortunately, the downsampling process is not a reversible process. What is done at this stage is to accept that the Cb and Cr values of the pixels side by side are close. Since these values are averaged and stored in the downsampling process, the empty parts are filled with the same average value after the noise is removed. Naturally, there is some loss of data, but it is so low that the human eye cannot see it.

Now, the image is ready for inverse color space transformation. Finally, the image in the YCbCr color space is converted back to the RGB color space and the decryption process is completed. The flow-chart diagram of the decryption part of algorithm can be examined in Figure 4.5

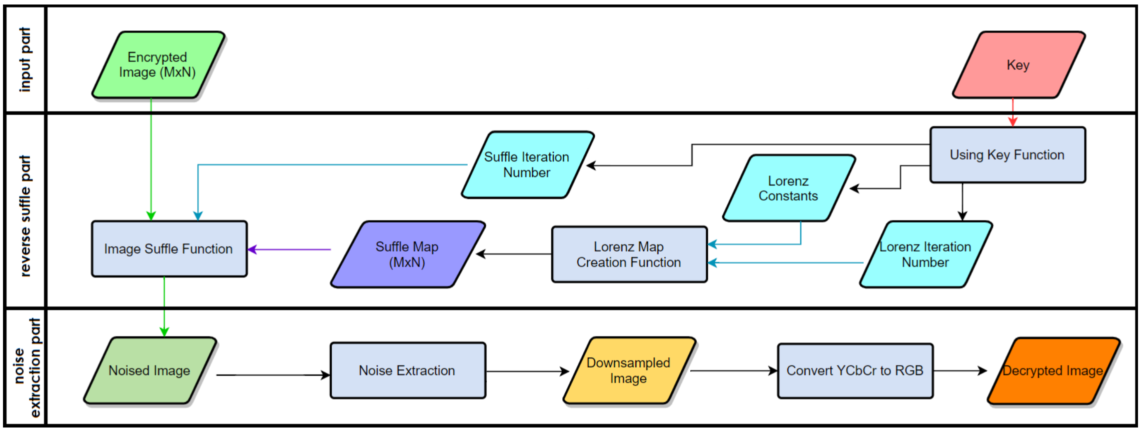

Figure 4.5 The flow-chart diagram of the decryption part

5
Results And Discussion

To prove that an image encryption algorithm is safe, it must be proved that it is successful according to some evaluation metrics. Some of the tests performed in order to make a good image encryption system successful are explained and evaluations of the proposed system are given in the subheadings of this section. But before that, to observe whether the algorithm causes a significant loss of data, we can examine an example of the encrypted image and then decrypted.

Plain Image Cipher Image Deciphered Image

Figure 5.1 Plain, cipher and deciphered versions of a sample image

As can be seen in Figure 5.1, the proposed algorithm does not cause a change in the image that can be noticed by the human eye. The memory sizes of plain, cipher and deciphered image are 64.6 KB, 763 KB and 80.0 KB respectively. Although the size of the cipher image is larger than the plain image, the size of deciphered image is fairly close to the plain image.

5.1 Histogram Analysis

The histogram analysis is to examine how much the histogram of the encoded image differs from the original picture. For image encryption algorithms, the histogram of the encrypted image must be totally different from the histogram of the original image and have a uniform distribution, it means that the probability of occurrence of any grayscale value is the same.

Whether the system meets these conditions, the histogram graph can be seen by examining it with the naked eye. For a more measurable analysis, the histograms of plain and cipher images are calculated first. Then the absolute difference between these two histograms is taken and the area under its curve is divided by the total area of the image [2]. In Figure 5.2, histogram analysis of a sample color image and its cipher image can be observed as; upper left –plain image; upper right – cipher image; lower left – histogram of plain image; lower right – histogram of cipher image.

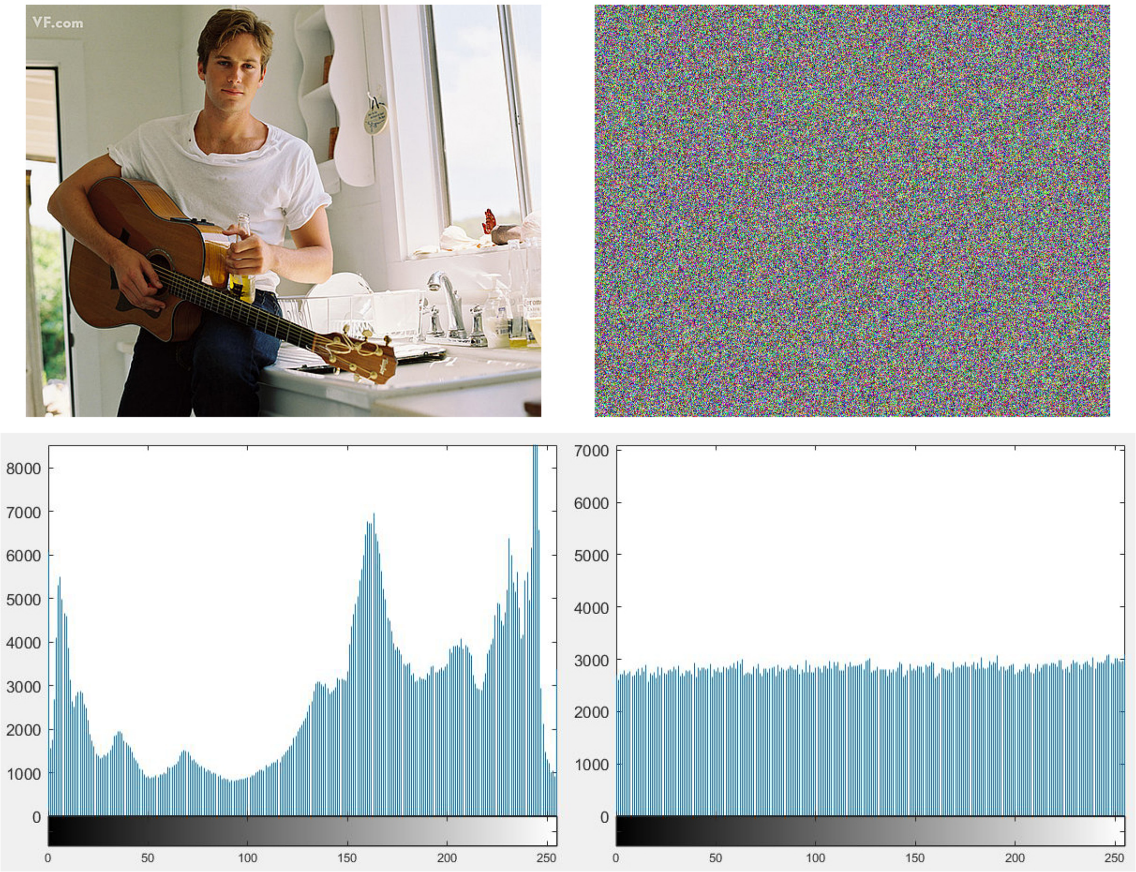

Figure 5.2 Histogram analysis of the sample image and its cipher image

5.2 Correlation Analysis

For correlation analysis, correlation coefficient between plain and cipher images is calculated. The correlation coefficient is a numerical measure of a statistical relationship between two variables. Several types of correlation coefficient exist, but all of them assume values in the range from -1 to +1, where +1 indicates the strongest agreement and -1 the strongest disagreement. A result of zero indicates no relationship at all.

It is also a useful metric to assess the encryption quality of any image cryptosystem. The correlation coefficient is calculated with using numerical difference between pixels at the same indices in the plain image and the cipher image [2]. What is expected of a good cryptography algorithm is that the correlation between plain and encrypted image is 0.

In addition, the correlation analysis can be used to measure that the pixels of the generated cipher image are completely random. For this purpose, similarities of the adjacent pixels of the picture are examined horizontally, vertically and diagonally. The correlation coefficient calculated on adjacent pixels should be high for plain image and low for cipher image. Also, these results can be visualized with a graph for better understanding.

Table 5.1 Correlation analysis results of a sample image

	Iteration Numbers	**Horizontal**	**Vertical**	**Diagonal**
Plain image	Lorenz: 4	0.96333	0.96411	0.94566
Cipher image	Suffle: 16	-0.0019281	-0.016545	0.012841
Correlation between Plain and Cipher Image			0,025027	
Plain image	Lorenz: 8	0.96406	0.96748	0.94363
Cipher image	Suffle: 32	-0.011906	-0.0078374	-0.014297
Correlation between Plain and Cipher Image			-0.046985	

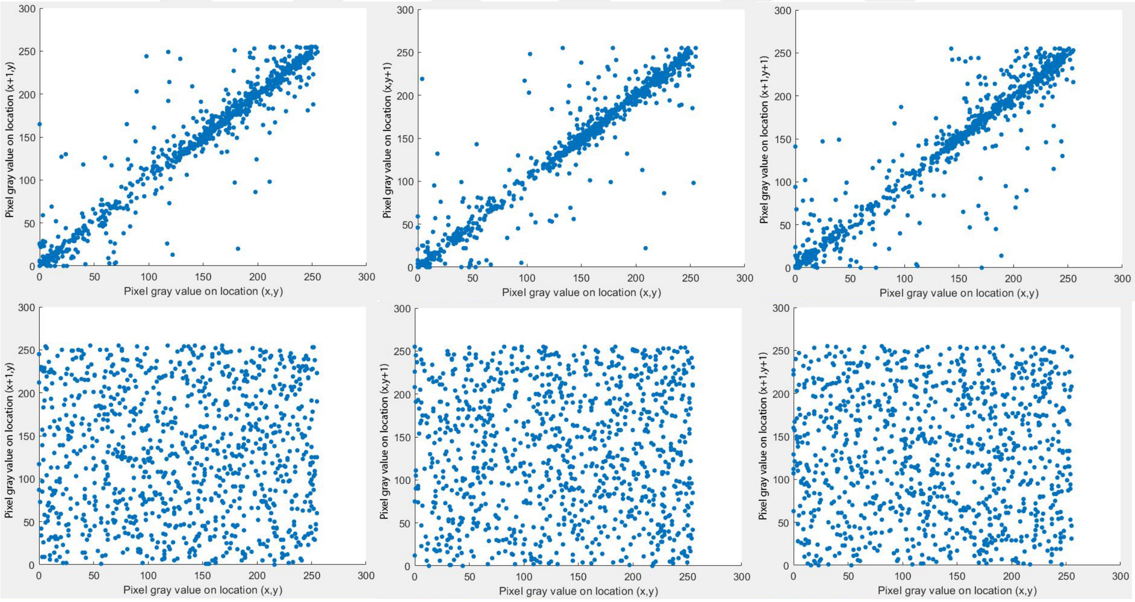

Figure 5.3 Correlation graphs of the sample image

Horizontal, vertical, diagonal correlation and correlation between the sample plain and cipher image graphs can be examined in Table 5.1 and Figure 5.3 Upper graphs are about correlation between adjacent pixels of plain image, lower graphs about cipher image; graphs in different columns show respectively horizontal, vertical and diagonal in both row.

5.3 Avalanche Effect

The Avalanche effect that mentioned in Section 2.2.4 can be used as an effective metric to test the efficiency of the diffusion mechanism. The Avalanche effect metric is the percentage of different bits between the two cipher images. If they are different from each other in half of their bits, it can said that the encryption algorithm possesses good diffusion characteristics.

To measure the influence of this effect, a single pixel of a plain image is modified, and both images are encrypted with the examined encryption algorithm. The same test should be done on the keys as well. The plain image is encrypted with two separate keys with a single bit different and the resulting cipher images are compared with each other. It is important that the test is performed in both cases. Because when the key is changed, the measured difference also helps to evaluate the success of the confusion mechanism.

In Figure 5.4, there are two cipher images and the difference that occurs when a pixel is changed on the top line; and two cipher images and the difference occurs when a bit of the key changes on the bottom line.

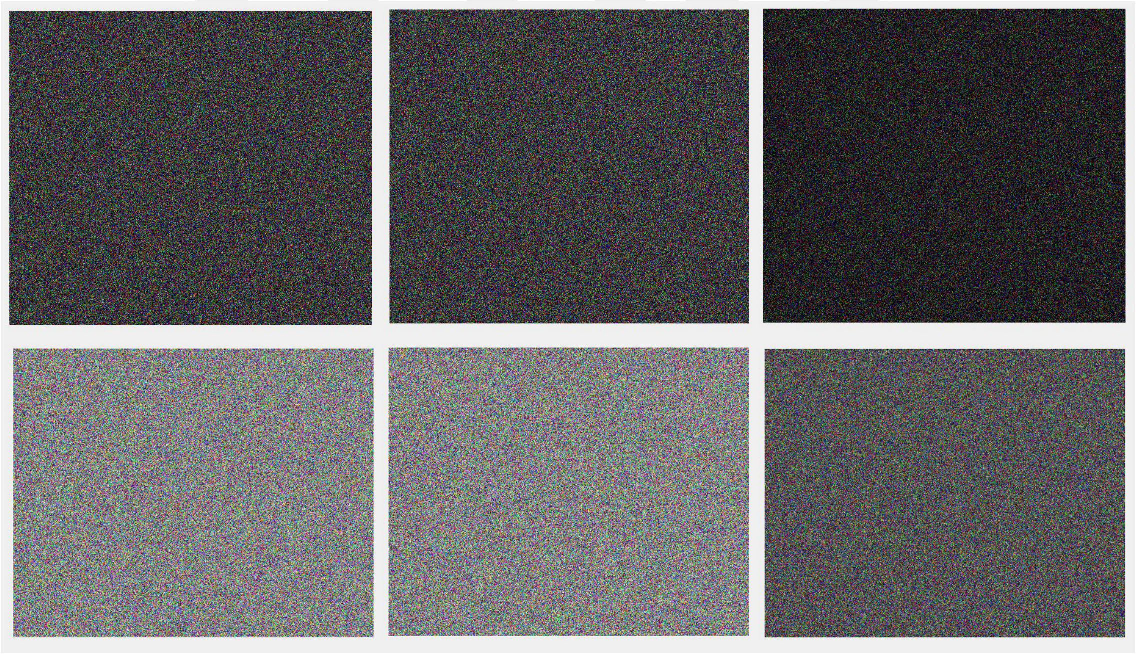

Figure 5.4 Avalanche effect test results

5.4 Performance and Optimization

In order to implement the tests described in the first three sections of the chapter; three different data sets have been formed.

The first data set contains 10 images: an RGB and a grayscale human picture; completely black, white, pure blue, red and green single-color images; two simple pictures with a black circle and square frame on a white background; and a picture with a yellow text on a black background. Also, only one pixel in different versions of all of these images are also available in the dataset. The total number of images is 20.

The second data set contains first data set and 90 more .bmp, .jpg and .png images of different sizes and specifications: 5 of them are the pictures are grayscale, 15 of them are synthetic images and 80 of them are color photographs that have different color tones and densities. Images vary in size from 363×420 to 1544×1368.

And lastly, the third data set contains square-shaped versions with different edge lengths of a sample RGB image with complex colors, such as; 16, 32, 64, 128, 256, 512 and 1024. This data set was used for obtaining the changing of process time according to image size.

Table 5.2 Process time changing according to iteration numbers

Iteration Number	Time (for Lorenz)	Time (for Reduction)
1	4.752	4.752
2	5.932	5.483
4	9.523	7.392
8	18.935	11.435
16	48.444	19.167
32	134.126	34.217
64	453.203	64.694

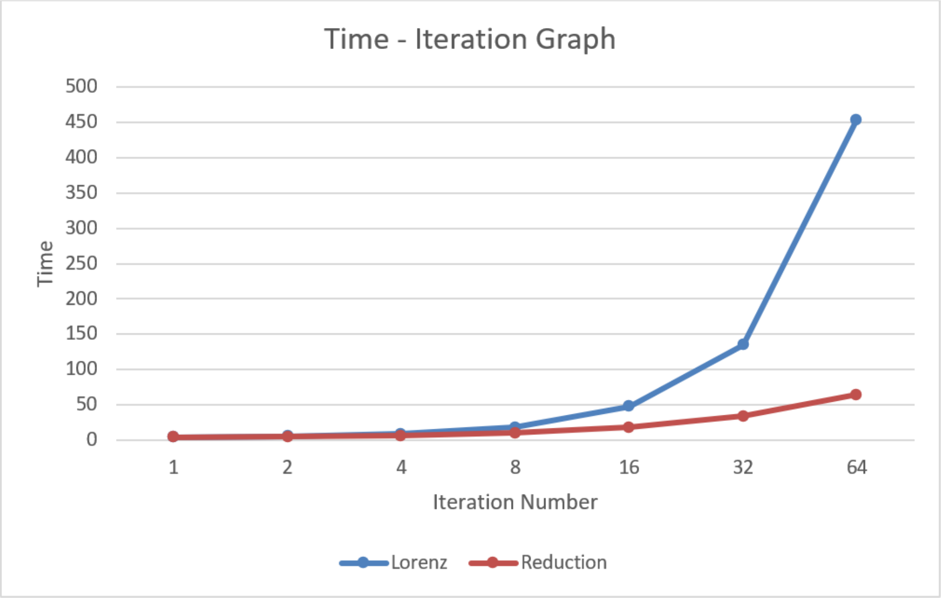

Figure 5.5 Time - Iteration Graph according to Table 5.2

First, the algorithm has been tested for process speed and time. In the Table 5.2, it can be observed how the number of iterations affects the process time. The chart is shown in Figure 5.5.

All calculations were made on a single example RGB image with using the same key. The first column shows how the process time changes when the number of Lorenz iterations increases while the reduction iteration number is one. In the second column, Lorenz iteration number is one.

As can be easily seen in the graph, in both cases the increase in process time is exponential but the Lorenz line increases faster. This indicates that the increase in the number of Lorenz iterations is more effective than the increase in reduction iterations.

Table 5.3 The relationship between process time and image size for different iteration values

File Name	Image Size	Cipher time (seconds)		
		R16 L4	32 L8	Substitution
16.png	16×16	0.041	0.078	0.037
32.png	32×32	0.143	0.205	0.062
64.png	64×64	0.384	0.715	0.331
128.png	128×128	1.418	2.991	1.573
256.png	256×256	5.979	12.74	6.761
512.png	512×512	25.279	54.019	28.74
1024.png	1024×1024	107.748	214.958	107.21

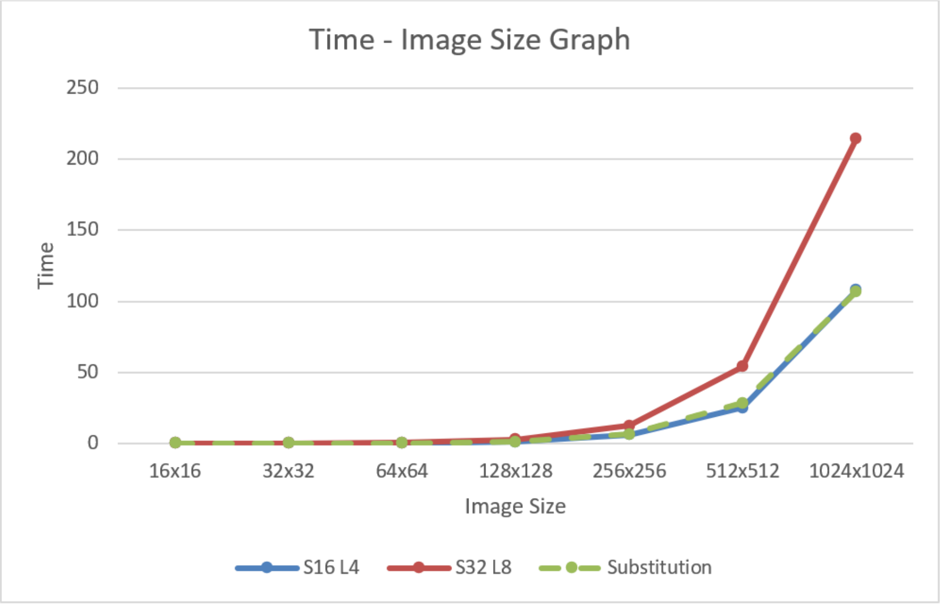

Figure 5.6 Time - Size Graph according to Table 5.3

The relationship between process speed and image size can also be observed in Table 5.3. This test was performed using the data set three and the determined optimal range of iteration numbers.

As the graph shows that the increase is exponential; when the edges sizes of image is doubled, the process time is also almost doubled. In this case, it can be said that the size of the picture directly affects the process time.

After the time tests, the relationship between the number of iterations and correlation was investigated. Tables in Appendix-C show the results of these tests. In the below, Table 5.4, 5.5 and 5.6 summarize the results of these tests for some selected images. In Figures 5.7, 5.8 and 5.9, these results are given graphically.

The first three rows of Table 5.4; horizontal, vertical and diagonal correlation test results of the ciphered RGB sample image. The values in the fourth row are correlation between plain and cipher versions of same image. The fifth row is about the Avalanche test results of same image; the correlation between cipher image of original sample image and cipher image of one pixel changed version of sample image. The last three rows are also about Avalanche effect tests, in order; for a gray-scale image, for a completely black or single colored image and for an image with yellow characters on black background. Since the data in the table are selected data from the tables that in Appendix-C, more data can be examined in there.

As seen in the graphs at Figure 5.7, there is no difference in the correlation analysis between determining the number of Lorenz iterations as one or more while the reduction iteration number is one. Applying Lorenz iteration number as one or more for sample RGB image downs the correlation between plain image and cipher image to around 0. The correlation between cipher image of original sample image and cipher image of one pixel changed version of sample image is 0.5 while the Lorenz iteration number is one or more. As a result, the correlation tests performed on a color image showed that the correlation was independent of the number of Lorenz iterations.

Tests to measure the effect of avalanche were performed for a completely black image, a grayscale image and an image that contains single color characters on a black background, and similar results were obtained. The number of Lorenz iterations for the grayscale image and the completely black image did not cause any change in the correlation, but for the image that contains single color characters on a black background, it showed a changing effect in the 0.5 - 0.8 band. Although, the fact that the number of Lorenz iterations increased but the correlation did not change is same.

Table 5.4 The relationship between correlation and Lorenz iteration number

	Plain	**L1**	**L2**	**L4**	**L8**	**L16**
horizontal correlation	0.98321	-0.022833	-0.012399	0.024133	-0.065212	0.062569
vertical correlation	0.95632	-0.009507	-0.041898	0.0027197	-0.021608	-0.018595
diagonal correlation	0.96768	-0.009507	-0.06934	-0.013684	0.040258	0.04926
btw plain - cipher	0.96768	-0.012936	0.032288	0.0045708	0.043955	0.044601
btw original - pointed	0.96768	0.5371	0.58924	0.5551	0.61989	0.56032
original - pointed gray	0.98728	0.98447	0.99448	0.99415	0.98714	0.99237
original - pointed char	0.94586	0.7026	0.69694	0.65588	0.7361	0.68979
original - pointed black	1	1	1	1	1	1

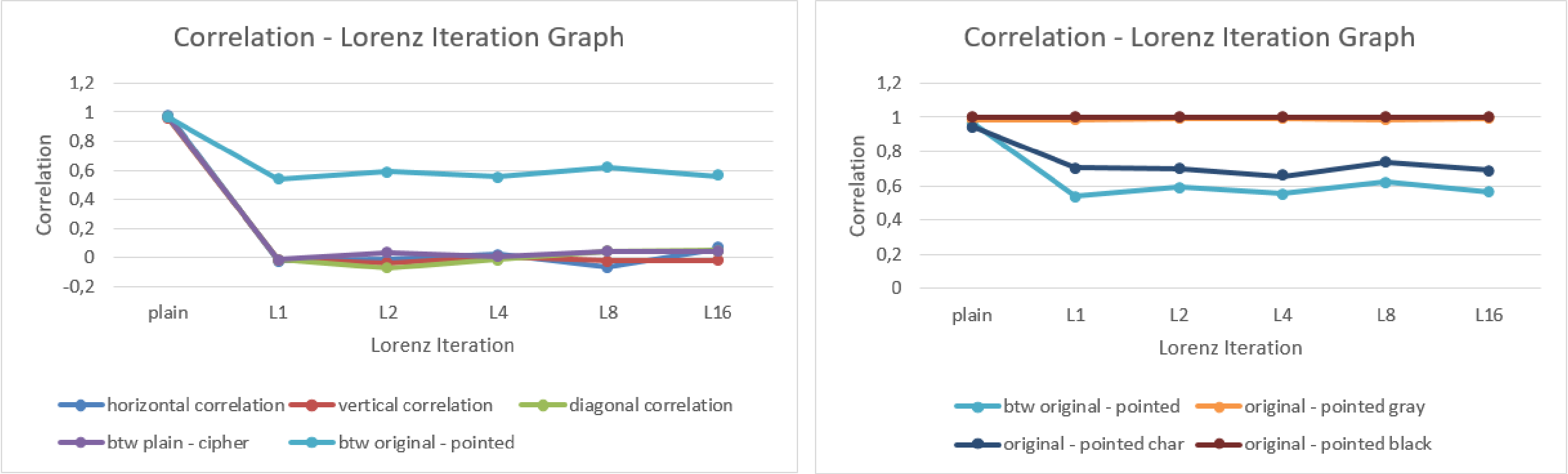

Figure 5.7 Correlation - Lorenz Iteration Graphs according to Table 5.4

Table 5.5 The relationship between correlation and Reduction iteration number

	Plain	**R1**	**R2**	**R4**	**R8**	**R16**	**R32**	**R64**
horizontal correlation	0.98321	-0.022833	0.039176	0.038197	-0.003112	0.066627	0.027361	0.014246
vertical correlation	0.95632	-0.009507	-0.02977	-0.048733	0.078402	-0.027458	-0.0026868	0.0024516
diagonal correlation	0.96768	-0.009507	0.013864	-0.024197	0.027009	0.038471	-0.053878	-0.015953
btw plain - cipher	0.96768	-0.012936	0.033095	0.058011	0.037177	-0.010488	0.049132	-0.082721
btw original - pointed	0.96768	0.5371	0.50402	0.47145	0.47257	0.49285	0.49292	0.011931
original - pointed gray	0.98728	0.98447	0.98608	0.89866	0.71488	0.64524	0.6672	0.031319
original - pointed char	0.94586	0.7026	0.55824	0.51458	0.50683	0.48465	0.49019	0.513368
original - pointed black	1	1	0.99947	1	0.97526	0.49153	0.50783	0.484136

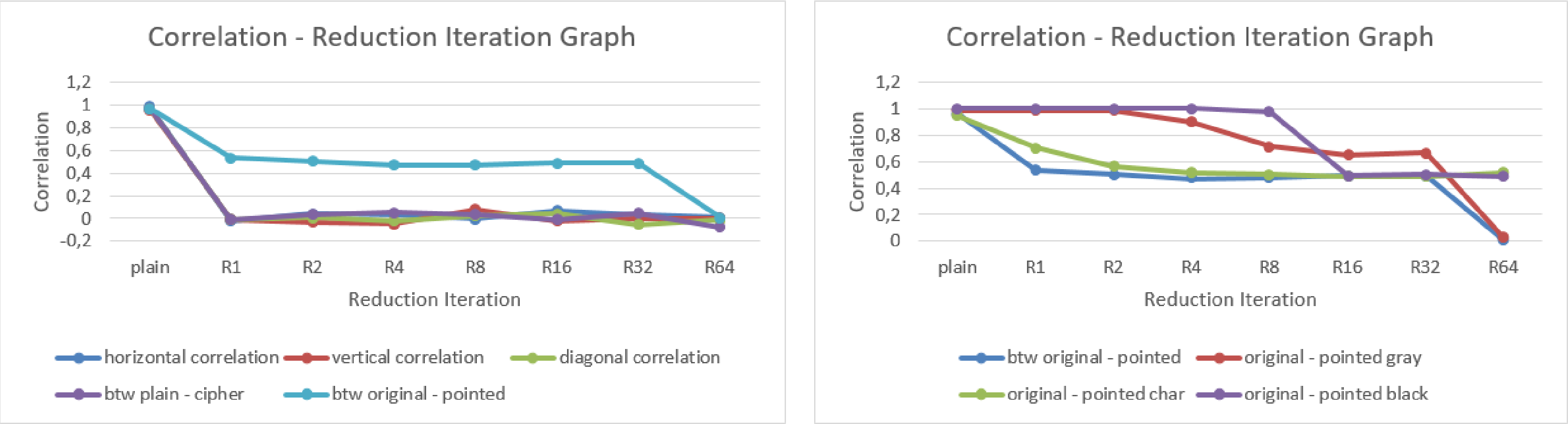

Figure 5.8 Correlation - Reduction Iteration Graphs according to Table 5.5

Table 5.6 The relationship between correlation and iteration numbers

	Plain	R1 L1	R16 L4	R32 L4	R16 L8	R32 L8	R64 L4
horizontal correlation	0.98321	-0.022833	-0.034536	0.048638	0.0069215	-0.013944	-0.035252
vertical correlation	0.95632	-0.009507	0.0010577	0.036172	0.0067443	0.0055503	0.046554
diagonal correlation	0.96768	-0.009507	0.023892	0.033337	-0.037288	0.016667	0.013884
btw plain - cipher	0.96768	-0.012936	0.020305	0.024883	-0.041751	0.0060137	-0.043453
btw original - pointed	0.96768	0.5371	0.015267	0.028768	0.045825	-0.030162	-0.0011326
original - pointed gray	0.98728	0.98447	-0.042448	0.029643	0.020411	-0.03698	0.012211
original - pointed char	0.94586	0.7026	0.0090348	-0.033806	-0.013197	0.0071253	-0.013363
original - pointed blue	0.99548	1	0.029258	-0.0082635	-0.060055	0.042641	0.031836

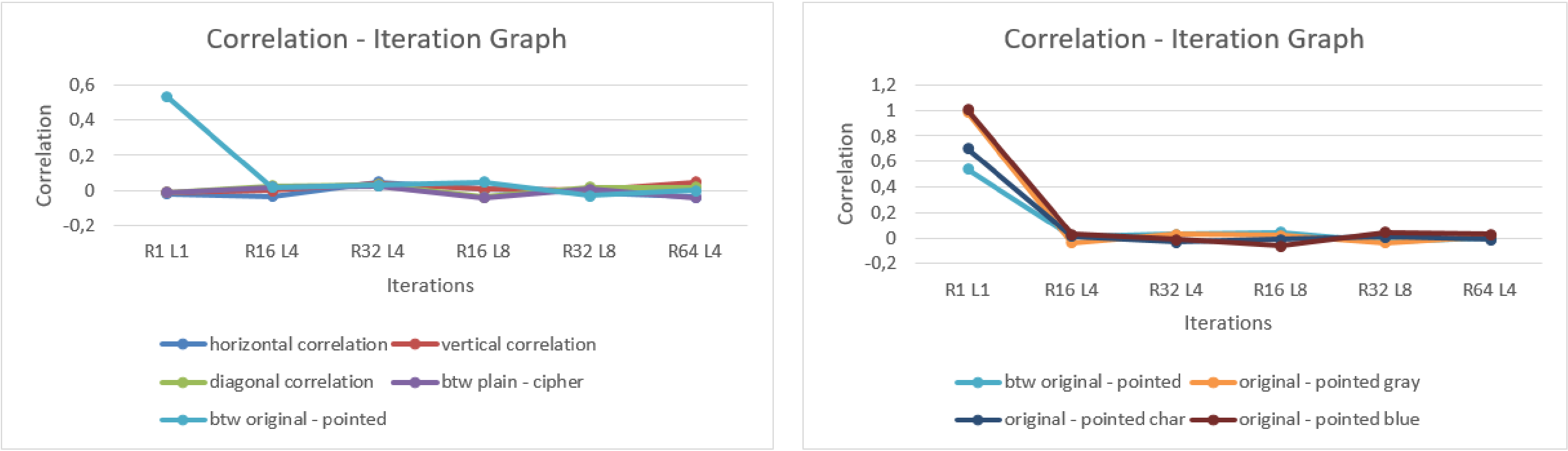

Figure 5.9 Correlation - Iteration Graphs according to Table 5.6

Table 5.5 and Figure 5.8 shows the results when the same tests are performed for the number of reduction iterations. The information in the rows is the same with the previous table. As seen in the graphs, performing one or more reduction operations to a RGB image does not cause much change in correlation; in each case the correlation is close to 0. On the other hand, when the tests to measure the Avalanche effect are examined, it is seen that while the number of iterations is between 1-32, the correlation changes by about 0.5, and when the iteration is 64, correlation decreases to 0.

When the Avalanche test is applied on the completely black image, the grayscale image and the image that containing the characters; it is seen that the correlation result of black image has only reached 0.5 in 16 iterations and it has not changed for 64 iterations. The correlation of grayscale image decreased from 2 iterations to 32 iterations and reached to 0 in 64 iterations. Correlation of the image with the characters was exponentially reduced until 16 iterations and then remained constant at around 0.5.

As can be seen from all these data, although the Lorenz process has no effect on the correlation results and Avalanche tests, the reduction process has a strong effect. The high effect of avalanche makes it impossible to apply plain text attack to the algorithm.

In order to observe the effects of these two processes, the correlation results and Avalanche effect for different Lorenz and reduction iteration values were examined and the results are summarized in Table 5.6 and the graphs of these results can be seen in Figure 5.9.

As shown in the graphs, when reduction iteration is 16 and Lorenz iteration is 4, the correlation is very close to 0 for each case and the iterations are 0 even though the iteration numbers change. Therefore, in order not to extend the processing time too much, the optimum values as the range of 4-8 for Lorenz and as the range of 16-32 for reduction were chosen. The values to be taken in this range was determined by the encryption key.

Assume that the pixel number of the plain image is n, Lorenz iteration number is l, reduction iteration number is r and suffle iteration number is r/8; the complexity of the proposed encryption algorithm is;

$$= l \times n \times 24 + l \times 256 + r \times n + r/8 \times n \tag{5.1}$$

According to the values that l can vary from 4 to 8 and r can vary from 16 to 32;

$$O = (8 \times n \times 24 + 8 \times 256 + 32 \times n + 32/8 \times n) = (n) \tag{5.2}$$

$$\Omega = (4 \times n \times 24 + 4 \times 256 + 16 \times n + 16/8 \times n) = (n) \tag{5.3}$$

The analysis shows that the complexity of the algorithm is (n).

5.5 Discussion

The tests described above were performed on 100 plain image – cipher image pairs that grayscale, black–white and colored with different keys. The results were similar, it was also seen that even in a completely black image, the algorithm made a large amount of change. In Figure 5.4, upper left image is cipher image of a completely black plain image and middle image is cipher image of a completely black except one white pixel plain image.

Some periodic repetition is observed in the histogram tests. The reason of this is reduction map that using for changing the color values of the pixels in the algorithm. The same map is a reason why even a black picture can be encrypted successfully.

As seen, there is no difference between the deciphered image and the original image that can be noticed by the human eye. This is because downsampling process is applied to pixel pairs, not to the 8×8 pixel blocks, instead of the JPEG algorithm. That is, almost no changes have been made in the image.

Apart from these, other important tests for image ciphers are performance and key space tests. The key space test was not performed, because the AES key generator was used when the algorithm was being tested. The length of the key used in the algorithm is 128 bits. Although it is possible to use a 256-bit or 512-bit key, it is doubtful that this will make a significant contribution to security.

One of the major problems faced by the proposed encryption algorithm was the production of the Lorenz map. The both Lorenz equations solution time, and the reduction of the results make the Lorenz map creation process critical. Because the failure to properly implement these processes will also damage the chaotic structure of the algorithm. Therefore, the creation of the Lorenz map is one of the most important parts of the proposed algorithm and should be carefully analyzed.

The proposed algorithm has an advantage; it can be used in any type of image file. But when the performance of the encryption system is evaluated, it is recognized that the working time of the algorithm is long, also it increases exponentially as the picture grows.

The main reason of this is that, because of displacement maps are the same size as the image, for each image and each key, Lorenz equations are solved from the beginning to reveal a new map. On the other hand, the displacement and pixel value changing can be performed more than once and not less than a certain threshold, according to the key value. All these are factors that increase the processing time.

In addition, it should also be tested with different attacks to ensure the security. On the other hand, the correlation tests show that, the desire to be plain text attack resistant was successfully accomplished.

In light of all these results, it can be said that the proposed algorithm is promising, open to development and needs to be evaluated. In future, it is planned to propose different ways to improve its performance and to be subjected to stronger tests in terms of security.

A
Time and Correlation Test Results of First Dataset

This appendix contains the results of some tests applied to the first dataset. These are process time changing according to iteration numbers test, the relationships tests between; correlation and Lorenz iteration number, correlation and reduction iteration number, correlation of plain and cipher images and Lorenz iteration, correlation of cipher original images and cipher one pixel changed images and Lorenz iteration, correlation of plain and cipher images and reduction iteration, correlation of cipher original images and cipher one pixel changed images and reduction iteration.

The data in Tables 5.2, 5.4, 5.5, 5.6 are taken from these test results. The values below are the values of the Lorenz parameters obtained from the switch and the key used in the test.

Key: D6-50-38-57-8B-EA-09-AC-29-5E-B4-BC-80-D6-D1-DF
Lorenz Sigma: 9.394995326
Lorenz Beta: 2.494060414
Lorenz Rho: 27.7785174

Table A.1 Encryption time for different iteration values and image sizes

File name	Cipher time (seconds)					
	Image Size	**L1**	**L2**	**L4**	**L8**	**L16**
armiehammer.jpg	550×440	4.752	5.932	9.523	18.935	48.444
armiehammerpoint.jpg	550×440	4.735	5.979	9.542	19.502	49.014
black.png	594×514	5.792	7.353	11.854	25.247	59.271
blackpoint.png	594×514	5.775	7.402	11.987	28.442	58.87
blue.png	594×514	5.845	7.445	12.107	24.704	60.15
bluepoint.png	594×514	5.91	7.68	52.513	25.426	59.213
circle.png	594×514	5.914	7.522	11.765	25.935	58.851
circlepoint.png	594×514	5.735	7.48	11.945	24.558	58.382
clarkgable.jpg	600×400	4.595	7.508	9.619	20.316	53.846
clarkgablepoint.jpg	600×400	4.585	6.087	9.584	19.952	52.351
green.png	594×514	5.853	7.887	11.838	24.783	57.956
greenpoint.png	594×514	7.808	7.73	12.137	24.888	58.547
red.png	594×514	5.798	7.781	12.064	25.474	58.928
redpoint.png	594×514	5.899	7.555	12.031	25.168	58.57
square.png	594×514	5.974	7.808	11.897	24.111	58.279
squarepoint.png	594×514	5.857	7.545	12.081	24.248	57.582
starwars.jpg	540×352	3.474	4.567	7.536	14.917	35.009
starwarspoint.jpg	540×352	3.506	4.901	7.447	14.642	34.437
white.png	594×514	5.763	7.469	12.012	24.525	58.061
whitepoint.png	594×514	5.699	7.777	11.952	24.353	57.151

Table A.2 Horizontal correlation results for different Lorenz iteration values

File name	Horizontal correlation					
	Plain	L1	L2	L4	L8	L16
armiehammer.jpg	0.98321	-0.022833	-0.012399	0.024133	-0.065212	0.062569
armiehammerpoint.jpg	0.97048	-0.045262	0.017537	0.051891	0.00053952	0.075821
black.png	1	-0.043936	0.056853	0.0041359	-0.0060274	-0.015271
blackpoint.png	1	0.018913	0.043106	0.0074641	-0.0056522	-0.0021137
blue.png	1	0.015982	-0.0050117	0.0047388	0.0284	-0.024549
bluepoint.png	1	-0.0091526	-0.012462	0.031041	-0.013953	0.011136
circle.png	0.93911	0.042148	0.022206	-0.042289	-0.056425	0.053598
circlepoint.png	0.91315	-0.044096	-0.030931	0.016496	0.0019811	-0.047426
clarkgable.jpg	0.99186	-0.022829	0.0056608	-0.039026	0.059074	0.0018227
clarkgablepoint.jpg	0.98802	0.012882	0.017718	-0.045951	-0.0080103	-0.062589
green.png	1	-0.023181	-0.0080953	-0.020676	0.0077525	0.00031626
greenpoint.png	1	-0.0069084	-0.037386	0.00053291	-0.0047521	0.0014193
red.png	1	-0.068136	0.024713	0.034033	-0.04928	0.011891
redpoint.png	1	-0.05848	0.0031344	-0.016758	0.021632	-0.016782
square.png	0.99082	0.14241	0.047222	0.0067795	0.017157	0.023319
squarepoint.png	1	0.12217	0.11005	-0.0075865	0.06203	-0.017872
starwars.jpg	0.96297	0.067832	-0.0030479	-0.025665	0.031616	0.0093543
starwarspoint.jpg	0.93729	0.042715	-0.0030446	0.0035232	-0.013054	0.0018151
white.png	1	-0.036349	-0.0064069	-0.00025977	0.024962	-0.014568
whitepoint.png	1	-0.023665	0.018169	0.0027224	-0.0093152	0.061914

Table A.3 Vertical correlation results for different Lorenz iteration values

File name	Vertical correlation					
	Plain	**L1**	**L2**	**L4**	**L8**	**L16**
armiehammer.jpg	0.95632	-0.009507	-0.041898	0.0027197	-0.021608	-0.018595
armiehammerpoint.jpg	0.94457	-0.029672	-0.024496	0.067915	0.00086945	-0.024719
black.png	1	-0.039084	-0.0041812	-0.0049078	-0.012759	-0.047666
blackpoint.png	1	-0.090028	-0.073071	-0.075406	-0.010663	-0.055837
blue.png	0.9976	0.044619	-0.017359	0.0093629	-0.040703	-0.049193
bluepoint.png	1	0.013964	-0.0073742	0.012439	-0.034292	0.023087
circle.png	0.93134	-0.06003	-0.021781	-0.021226	-0.062789	-0.0041974
circlepoint.png	0.77614	0.0053372	-0.015142	-0.049749	0.0030992	0.059185
clarkgable.jpg	0.97928	-0.017521	-0.044321	0.029174	0.048773	-0.020083
clarkgablepoint.jpg	0.98209	0.032581	0.017928	0.0068553	-0.0023812	-0.018852
green.png	0.99551	-0.033692	0.03034	-0.04997	-0.049342	0.031687
greenpoint.png	1	0.012334	-0.010225	0.028357	0.0031991	0.025849
red.png	1	0.053575	-0.063717	-0.038079	-0.032125	0.0081589
redpoint.png	1	0.003056	-0.0010858	-0.028885	-0.069002	-0.045794
square.png	0.99318	0.058219	0.13524	0.068328	0.09354	0.12276
squarepoint.png	0.99322	-0.012269	0.084003	0.10657	0.077745	0.090218
starwars.jpg	0.89457	-0.065962	0.01266	0.043792	0.011462	0.040126
starwarspoint.jpg	0.89205	0.016477	0.03398	0.029117	0.077764	0.014408
white.png	1	0.012401	-0.0136	-0.014131	-0.04328	-0.025283
whitepoint.png	1	-0.027617	-0.02041	-0.039957	0.0022537	0.0033125

Table A.4 Diagonal correlation results for different Lorenz iteration values

File name	**Diagonal correlation**					
	Plain	**L1**	**L2**	**L4**	**L8**	**L16**
armiehammer.jpg	0.96768	-0.009507	-0.06934	-0.013684	0.040258	0.04926
armiehammerpoint.jpg	0.9648	-0.029672	-0.020775	-0.044162	-0.0011602	-0.0029922
black.png	1	-0.039084	0.020275	0.0013537	-0.018833	0.030253
blackpoint.png	1	-0.090028	-0.034069	-0.051118	0.017165	0.051903
blue.png	0.99548	0.044619	0.0081596	0.060537	-0.0062841	0.047645
bluepoint.png	1	0.013964	0.017591	-0.011791	-0.042571	-0.026417
circle.png	0.97418	-0.06003	-0.02536	0.09628	0.010889	-0.017603
circlepoint.png	0.90105	0.0053372	0.1044	0.0088144	0.032523	0.0052521
clarkgable.jpg	0.98728	-0.017521	-0.017911	0.030068	0.0032848	-0.033353
clarkgablepoint.jpg	0.989	0.032581	-0.040174	0.042025	0.075604	-0.028038
green.png	1	-0.033692	-0.019052	0.0012912	0.039073	-0.012399
greenpoint.png	0.99772	0.012334	0.060169	0.029209	0.019466	-0.055096
red.png	1	0.053575	-0.013964	0.017454	-0.0092574	-0.032517
redpoint.png	0.99767	0.003056	-0.01819	-0.01565	-0.015233	-0.012359
square.png	0.99536	0.058219	0.072108	0.021031	0.059805	0.063421
squarepoint.png	0.99102	-0.012269	0.015393	0.042816	0.0055457	0.028211
starwars.jpg	0.94586	-0.065962	0.029025	-0.014731	0.011752	0.037456
starwarspoint.jpg	0.94777	0.016477	0.0022301	0.019925	0.0078421	0.047202
white.png	1	0.012401	0.0074026	0.035013	-0.012096	-0.0057429
whitepoint.png	1	-0.027617	0.015996	0.0063806	-0.014624	0.056634

Table A.5 Correlation between plain and cipher image results for different Lorenz iteration values

File name	Correlation between plain – cipher images					
	Plain	**L1**	**L2**	**L4**	**L8**	**L16**
armiehammer.jpg	0.96768	-0.012936	0.032288	0.0045708	0.043955	0.044601
armiehammerpoint.jpg	0.9648	0.016761	0.02299	-0.020574	0.023518	0.0027295
black.png	1	NaN	NaN	NaN	NaN	NaN
blackpoint.png	1	NaN	NaN	NaN	NaN	NaN
blue.png	0.99548	-0.010791	-0.0017823	0.011896	-0.070152	-0.062279
bluepoint.png	1	0.025307	-0.0088331	-0.05488	0.020411	0.0026482
circle.png	0.97418	0.03498	0.014683	0.061105	0.046613	-0.017393
circlepoint.png	0.90105	-0.03202	0.027688	-0.043683	0.013301	-0.0068105
clarkgable.jpg	0.98728	-0.0091124	0.020136	-0.0051012	-0.041295	-0.011512
clarkgablepoint.jpg	0.989	0.028396	0.011294	-0.057931	0.0020861	-0.021758
green.png	1	0.016774	0.017342	0.0092299	0.03761	-0.030973
greenpoint.png	0.99772	-0.010957	0.019635	-0.0012234	-0.0030371	-0.046945
red.png	1	0.02434	-0.0098895	-0.0020695	-0.0072381	-0.025401
redpoint.png	0.99767	-0.00057383	-0.0045497	-0.01546	-0.010025	0.020611
square.png	0.99536	0.069721	0.080981	0.10128	0.0042404	0.05253
squarepoint.png	0.99102	0.045297	0.042167	0.062838	0.1143	0.027748
starwars.jpg	0.94586	-0.021218	0.020042	0.014198	-0.0202	0.011825
starwarspoint.jpg	0.94777	-0.010651	-0.022522	-0.02363	-0.033813	-0.0026012
white.png	1	NaN	NaN	NaN	NaN	NaN
whitepoint.png	1	NaN	NaN	NaN	NaN	NaN

Table A.6 Correlation between cipher original image and cipher one pixel changed image results for different Lorenz iteration values

File name	Correlation between original – pointed images					
	Plain	**L1**	**L2**	**L4**	**L8**	**L16**
armiehammer.jpg	0.96768	0.5371	0.58924	0.5551	0.61989	0.56032
armiehammerpoint.jpg	0.9648	0.5371	0.58924	0.5551	0.61989	0.56032
black.png	1	1	1	1	1	1
blackpoint.png	1	1	1	1	1	1
blue.png	0.99548	1	1	1	1	1
bluepoint.png	1	1	1	1	1	1
circle.png	0.97418	1	1	1	1	1
circlepoint.png	0.90105	1	1	1	1	1
clarkgable.jpg	0.98728	0.98447	0.99448	0.99415	0.98714	0.99237
clarkgablepoint.jpg	0.989	0.98447	0.99448	0.99415	0.98714	0.99237
green.png	1	1	1	1	1	1
greenpoint.png	0.99772	1	1	1	1	1
red.png	1	1	1	1	1	1
redpoint.png	0.99767	1	1	1	1	1
square.png	0.99536	1	1	1	1	1
squarepoint.png	0.99102	1	1	1	1	1
starwars.jpg	0.94586	0.7026	0.69694	0.65588	0.7361	0.68979
starwarspoint.jpg	0.94777	0.7026	0.69694	0.65588	0.7361	0.68979
white.png	1	1	1	1	1	1
whitepoint.png	1	1	1	1	1	1

Table A.7 Encryption time for different reduction iteration values

File name	Cipher time (seconds)						
	Image Size	**R1**	**R2**	**R4**	**R8**	**R16**	**R32**
armiehammer.jpg	550×440	4.752	5.483	7.392	11.435	19.167	34.217
armiehammerpoint.jpg	550×440	4.735	5.457	7.363	11.428	18.968	36.363
black.png	594×514	5.792	6.858	9.576	13.965	25.024	44.754
blackpoint.png	594×514	5.775	8.05	9.104	14.188	23.785	44.689
blue.png	594×514	5.845	7.539	8.994	15.494	24.178	43.809
bluepoint.png	594×514	5.91	7.565	9.25	14.421	23.46	44.17
circle.png	594×514	5.914	7.029	9.005	14.265	23.335	43.826
circlepoint.png	594×514	5.735	7.136	10.384	14.055	24.011	43.828
clarkgable.jpg	600×400	4.595	5.78	7.306	11.412	18.926	34.833
clarkgablepoint.jpg	600×400	4.585	7.355	7.304	11.301	18.615	34.127
green.png	594×514	5.853	7.299	9.263	14.156	24.194	44.235
greenpoint.png	594×514	7.808	6.795	9.054	14.503	23.258	43.601
red.png	594×514	5.798	6.804	9.172	14.245	23.865	43.467
redpoint.png	594×514	5.899	7.261	9.056	14.346	24.673	43.578
square.png	594×514	5.974	6.806	9.243	14.34	24.434	41.853
squarepoint.png	594×514	5.857	6.96	9.204	14.169	24.661	42.636
starwars.jpg	540×352	3.474	4.419	5.684	9.057	15.148	26.125
starwarspoint.jpg	540×352	3.506	4.398	5.87	8.818	14.755	26.16
white.png	594×514	5.763	7.196	9.47	14.359	23.836	42.529
whitepoint.png	594×514	5.699	7.011	8.901	14.082	23.948	42.753

Table A.8 Horizontal correlation results for different reduction iteration values

File name	Horizontal correlation						
	Plain	R1	R2	R4	R8	R16	R32
armiehammer.jpg	0.98321	-0.022833	0.039176	0.038197	-0.003112	0.066627	0.027361
armiehammerpoint.jpg	0.97048	-0.045262	-0.049041	-0.029849	0.0094775	0.02554	-0.0077215
black.png	1	-0.043936	-0.019948	0.052425	-0.018014	-0.0083516	-0.02804
blackpoint.png	1	0.018913	-0.015288	-0.035943	0.011788	-0.019606	-0.0044289
blue.png	1	0.015982	-0.01172	-0.029004	0.055341	0.030846	-0.079936
bluepoint.png	1	-0.0091526	0.028998	-0.024801	0.065118	0.012871	-0.016771
circle.png	0.93911	0.042148	0.014405	0.0064408	0.012569	0.012426	-0.024122
circlepoint.png	0.91315	-0.044096	-0.078851	-0.062971	0.035525	0.037772	-0.0035578
clarkgable.jpg	0.99186	-0.022829	0.057618	-0.028586	-0.0085902	0.040309	0.043716
clarkgablepoint.jpg	0.98802	0.012882	-0.012412	0.01636	0.016951	0.025029	0.042726
green.png	1	-0.023181	-0.015261	-0.017971	-0.049006	0.040263	0.058851
greenpoint.png	1	-0.0069084	-0.04136	0.033627	0.00316	0.03199	-0.024191
red.png	1	-0.068136	0.022793	0.031063	-0.0056595	0.027295	-0.072419
redpoint.png	1	-0.05848	-0.016382	0.017942	0.066082	0.015069	-0.000952
square.png	0.99082	0.14241	-0.028757	0.00089911	0.083413	0.051077	0.036311
squarepoint.png	1	0.12217	0.011833	0.065934	0.029006	-0.0083057	-0.015222
starwars.jpg	0.96297	0.067832	0.072921	0.018732	0.009219	-0.017585	0.060556
starwarspoint.jpg	0.93729	0.042715	0.0282	-0.016718	-0.023205	-0.0084037	0.045394
white.png	1	-0.036349	-0.003394	0.016495	-0.035934	-0.067344	-0.026969
whitepoint.png	1	-0.023665	-0.033953	0.0081872	0.0063876	-0.0092984	0.021315

Table A.9 Vertical correlation results for different reduction iteration values

File name	Vertical correlation						
	Plain	**R1**	**R2**	**R4**	**R8**	**R16**	**R32**
armiehammer.jpg	0.95632	-0.009507	-0.02977	-0.048733	0.078402	-0.027458	-0.0026868
armiehammerpoint.jpg	0.94457	-0.029672	0.019821	0.028399	-0.0046512	0.050028	-0.002568
black.png	1	-0.039084	-0.016661	-0.038564	-0.016005	0.075325	-0.066879
blackpoint.png	1	-0.090028	-0.058838	0.0057683	0.039164	0.028157	-0.026301
blue.png	0.9976	0.044619	-0.029208	0.018473	0.051897	-0.010691	0.0038677
bluepoint.png	1	0.013964	0.0076955	-0.024834	0.044838	-0.01965	-0.0027437
circle.png	0.93134	-0.06003	-0.035525	0.010263	-0.027022	-0.0175	0.023203
circlepoint.png	0.77614	0.0053372	0.00066225	0.0033007	0.038759	0.022152	0.059701
clarkgable.jpg	0.97928	-0.017521	0.023176	-0.0087862	0.0048705	-0.016671	0.021851
clarkgablepoint.jpg	0.98209	0.032581	-0.065886	-0.033263	0.0023971	0.0082535	0.040922
green.png	0.99551	-0.033692	0.034712	0.035131	0.032023	0.0085266	0.020029
greenpoint.png	1	0.012334	0.040944	-0.062467	0.029135	-0.022586	0.0069765
red.png	1	0.053575	-0.0015665	-0.034606	-0.044189	0.013882	0.02386
redpoint.png	1	0.003056	-0.020291	-0.059984	-0.058777	-0.017514	-0.027918
square.png	0.99318	0.058219	0.12416	0.0041984	0.020664	0.0082385	0.0044187
squarepoint.png	0.99322	-0.012269	0.044204	0.097445	-0.053082	-0.030391	-0.0073755
starwars.jpg	0.89457	-0.065962	0.042893	0.041504	-0.0007008	-0.035233	-0.065797
starwarspoint.jpg	0.89205	0.016477	0.0068192	0.02446	-0.0020076	0.028585	0.062435
white.png	1	0.012401	-0.023026	0.032639	0.032829	0.03097	0.025105
whitepoint.png	1	-0.027617	0.02399	-0.010553	-0.05866	-0.0014862	0.018682

Table A.10 Diagonal correlation results for different reduction iteration values

File name	Diagonal correlation						
	Plain	R1	R2	R4	R8	R16	R32
armiehammer.jpg	0.96768	-0.009507	0.013864	-0.024197	0.027009	0.038471	-0.053878
armiehammerpoint.jpg	0.9648	-0.029672	0.029044	-0.051299	0.011701	0.020905	0.015816
black.png	1	-0.039084	-0.013065	-0.048685	-0.0015426	0.036201	0.027732
blackpoint.png	1	-0.090028	0.029424	-0.045008	0.067622	0.032771	0.0049667
blue.png	0.99548	0.044619	0.02068	-0.019395	-0.058747	-0.031025	0.017109
bluepoint.png	1	0.013964	0.015854	0.038125	-0.019815	-0.012909	0.0046639
circle.png	0.97418	-0.06003	-0.001596	-0.022289	-0.034401	-0.014962	-0.039915
circlepoint.png	0.90105	0.0053372	0.026826	0.0086642	-0.049917	0.011964	-0.033184
clarkgable.jpg	0.98728	-0.017521	0.0092626	0.036282	0.030406	0.018368	-0.05064
clarkgablepoint.jpg	0.989	0.032581	0.030941	0.014171	-0.014021	0.019179	-0.054103
green.png	1	-0.033692	0.0020041	0.028116	0.032113	-0.024449	-0.014872
greenpoint.png	0.99772	0.012334	-0.053163	0.0025362	-0.035356	0.02098	-0.0012484
red.png	1	0.053575	-0.0054842	0.01716	0.031101	-0.010341	0.0042987
redpoint.png	0.99767	0.003056	-0.060794	-0.02121	0.0009422	0.013859	-0.029902
square.png	0.99536	0.058219	0.039759	0.024599	0.017364	-0.049266	-0.028395
squarepoint.png	0.99102	-0.012269	0.04002	0.073855	0.01915	0.0085832	-0.038634
starwars.jpg	0.94586	-0.065962	0.0021592	-0.011478	-0.024346	-0.0027723	-0.0072665
starwarspoint.jpg	0.94777	0.016477	-0.035228	-0.024741	-0.048843	0.027253	-0.002195
white.png	1	0.012401	0.032675	0.0029797	-0.043695	-0.027166	-0.0066384
whitepoint.png	1	-0.027617	-0.032126	0.067533	0.03097	0.0070524	0.010671

Table A.11 Correlation between plain and cipher image results for different reduction iteration values

File name	Correlation between plain – cipher images						
	Plain	R1	R2	R4	R8	R16	R32
armiehammer.jpg	0.96768	-0.012936	0.033095	0.058011	0.037177	-0.010488	0.049132
armiehammerpoint.jpg	0.9648	0.016761	0.025496	0.021832	-0.01374	-0.026264	0.022306
black.png	1	NaN	NaN	NaN	NaN	NaN	NaN
blackpoint.png	1	NaN	NaN	NaN	NaN	NaN	NaN
blue.png	0.99548	-0.010791	-0.0038305	0.0097315	0.042435	-0.010947	0.025292
bluepoint.png	1	0.025307	-0.016216	-0.030594	-0.021899	-0.013554	0.059515
circle.png	0.97418	0.03498	0.0054536	0.0015972	0.039222	0.065599	-0.017673
circlepoint.png	0.90105	-0.03202	0.0063169	0.0038933	-0.038995	-0.017881	0.059286
clarkgable.jpg	0.98728	-0.0091124	0.00066171	-0.017949	0.0091151	0.026435	-0.027689
clarkgablepoint.jpg	0.989	0.028396	0.031757	-0.035531	0.018702	-0.038702	-0.021519
green.png	1	0.016774	-0.015054	-0.030926	0.019219	-0.023834	0.0074724
greenpoint.png	0.99772	-0.010957	0.013164	-0.000539	-0.0038648	-0.024186	0.0062483
red.png	1	0.02434	0.044527	-0.0033309	-0.032066	-0.012356	0.045169
redpoint.png	0.99767	-0.0005738	0.015839	0.064281	0.007307	-0.021502	-0.035722
square.png	0.99536	0.069721	0.096749	0.039923	0.015049	-0.0009951	-0.012753
squarepoint.png	0.99102	0.045297	0.059711	0.051262	0.010626	-0.02217	-0.013518
starwars.jpg	0.94586	-0.021218	0.0065591	0.043271	0.047088	0.0068748	-0.0002169
starwarspoint.jpg	0.94777	-0.010651	0.011213	0.041956	0.042924	-0.018074	-0.010587
white.png	1	NaN	NaN	NaN	NaN	NaN	NaN
whitepoint.png	1	NaN	NaN	NaN	NaN	NaN	NaN

Table A.12 Correlation between cipher original image and cipher one pixel changed image results for different reduction iteration values

File name	Correlation between original – pointed images						
	Plain	**R1**	**R2**	**R4**	**R8**	**R16**	**R32**
armiehammer.jpg	0.96768	0.5371	0.50402	0.47145	0.47257	0.49285	0.49292
armiehammerpoint.jpg	0.9648	0.5371	0.50402	0.47145	0.47257	0.49285	0.49292
black.png	1	1	0.99947	1	0.97526	0.49153	0.50783
blackpoint.png	1	1	0.99947	1	0.97526	0.49153	0.50783
blue.png	0.99548	1	1	1	0.98143	0.67975	0.7085
bluepoint.png	1	1	1	1	0.98143	0.67975	0.7085
circle.png	0.97418	1	0.99848	1	0.99267	0.64761	0.68282
circlepoint.png	0.90105	1	0.99848	1	0.99267	0.64761	0.68282
clarkgable.jpg	0.98728	0.98447	0.98608	0.89866	0.71488	0.64524	0.6672
clarkgablepoint.jpg	0.989	0.98447	0.98608	0.89866	0.71488	0.64524	0.6672
green.png	1	1	1	1	0.98401	0.66134	0.67854
greenpoint.png	0.99772	1	1	1	0.98401	0.66134	0.67854
red.png	1	1	1	0.9996	0.99285	0.71247	0.69175
redpoint.png	0.99767	1	1	0.9996	0.99285	0.71247	0.69175
square.png	0.99536	1	1	1	0.99241	0.67697	0.66621
squarepoint.png	0.99102	1	1	1	0.99241	0.67697	0.66621
starwars.jpg	0.94586	0.7026	0.55824	0.51458	0.50683	0.48465	0.49019
starwarspoint.jpg	0.94777	0.7026	0.55824	0.51458	0.50683	0.48465	0.49019
white.png	1	1	1	1	0.97548	0.63734	0.62946
whitepoint.png	1	1	1	1	0.97548	0.63734	0.62946

Table A.13 Encryption time for different Lorenz and reduction iteration values

File name	Cipher time (seconds)					
	Image Size	**R1 L1**	**R16 L4**	**R32 L4**	**R16 L8**	**R32 L8**
armiehammer.jpg	550×440	4.752	24.029	39.183	32.741	81.585
armiehammerpoint.jpg	550×440	4.735	24.082	54.575	34.325	90.621
black.png	594×514	5.792	31.372	49.674	40.841	113.248
blackpoint.png	594×514	5.775	29.574	48.676	41.169	111.451
blue.png	594×514	5.845	30.07	49.164	42.106	107.618
bluepoint.png	594×514	5.91	30.55	49.612	42.015	118.611
circle.png	594×514	5.914	29.892	48.358	41.845	95.947
circlepoint.png	594×514	5.735	30.095	47.662	40.05	63.105
clarkgable.jpg	600×400	4.595	23.517	38.346	33.11	48.756
clarkgablepoint.jpg	600×400	4.585	23.88	37.421	31.922	91.414
green.png	594×514	5.853	29.396	47.632	47.06	116.456
greenpoint.png	594×514	7.808	29.648	47.952	76.402	118.201
red.png	594×514	5.798	30.271	50.036	78.268	120.776
redpoint.png	594×514	5.899	30.581	48.489	76.38	117.593
square.png	594×514	5.974	30.018	48.666	78.361	102.689
squarepoint.png	594×514	5.857	30.335	50.577	77.65	99.657
starwars.jpg	540×352	3.474	18.46	31.333	48.792	36.303
starwarspoint.jpg	540×352	3.506	18.433	30.772	47.831	36.617
white.png	594×514	5.763	30.352	50.01	75.399	59.199
whitepoint.png	594×514	5.699	29.843	49.044	76.778	59.682

Table A.14 Horizontal correlation results for different Lorenz and reduction iteration values

File name	Horizontal correlation					
	Plain	**R1 L1**	**R16 L4**	**R32 L4**	**R16 L8**	**R32 L8**
armiehammer.jpg	0.98321	-0.022833	-0.034536	0.048638	-0.013944	-0.035252
armiehammerpoint.jpg	0.97048	-0.045262	0.018789	0.01184	-0.0060183	0.013163
black.png	1	-0.043936	-0.0090728	-0.00024259	-0.025702	-0.0037037
blackpoint.png	1	0.018913	-0.00815	-0.010353	0.020486	-0.010324
blue.png	1	0.015982	-0.0092672	0.040303	-0.030561	-0.058874
bluepoint.png	1	-0.0091526	0.053895	0.044847	0.011537	0.034297
circle.png	0.93911	0.042148	-0.018182	0.014306	0.0084524	-0.01533
circlepoint.png	0.91315	-0.044096	0.029197	0.02569	-0.0080372	0.002365
clarkgable.jpg	0.99186	-0.022829	-0.010894	0.020105	-0.0078399	0.022577
clarkgablepoint.jpg	0.98802	0.012882	0.031556	-0.017924	-0.0041408	-0.076282
green.png	1	-0.023181	-0.039918	-0.0082785	-0.0074962	0.056212
greenpoint.png	1	-0.0069084	-0.00384	0.0040922	0.0082995	-0.036978
red.png	1	-0.068136	-0.033151	0.018905	-0.027891	0.031919
redpoint.png	1	-0.05848	0.031713	-0.052572	0.005331	-0.033189
square.png	0.99082	0.14241	-0.0078204	-0.0066693	-0.006736	0.020821
squarepoint.png	1	0.12217	0.031166	0.005669	0.001205	0.044466
starwars.jpg	0.96297	0.067832	-0.023384	-0.002707	-0.0054858	-0.02897
starwarspoint.jpg	0.93729	0.042715	0.014125	-0.0069917	-0.039887	0.0082197
white.png	1	-0.036349	-0.030236	0.016512	-0.036731	0.035107
whitepoint.png	1	-0.023665	-0.011311	0.015633	-0.03595	-0.01116

Table A.15 Vertical correlation results for different Lorenz and reduction iteration values

File name	Vertical correlation					
	Plain	R1 L1	R16 L4	R32 L4	R16 L8	R32 L8
armiehammer.jpg	0.95632	-0.009507	0.0010577	0.036172	0.0055503	0.046554
armiehammerpoint.jpg	0.94457	-0.029672	0.015951	0.0058533	0.011609	0.030453
black.png	1	-0.039084	-0.0067532	0.10645	-0.047344	-0.0068621
blackpoint.png	1	-0.090028	-0.013794	0.013415	-0.019612	0.0065913
blue.png	0.9976	0.044619	-0.048601	0.017861	-0.023238	-0.039534
bluepoint.png	1	0.013964	-0.012289	0.0034428	-0.04482	-0.037615
circle.png	0.93134	-0.06003	0.013704	-0.038755	-0.0081498	-0.027752
circlepoint.png	0.77614	0.0053372	-0.0098134	0.035027	-0.032398	0.059091
clarkgable.jpg	0.97928	-0.017521	-0.0052007	0.022446	-0.032837	-0.015887
clarkgablepoint.jpg	0.98209	0.032581	-0.033276	0.0024592	0.060596	-0.015574
green.png	0.99551	-0.033692	-0.0086472	0.023638	-0.055838	0.030993
greenpoint.png	1	0.012334	-0.0080423	-0.014563	0.068273	-0.016894
red.png	1	0.053575	0.0071499	-0.037529	-0.056159	0.012435
redpoint.png	1	0.003056	0.034123	0.023859	-0.010792	0.002539
square.png	0.99318	0.058219	-0.021969	0.072879	0.00036518	-0.0032655
squarepoint.png	0.99322	-0.012269	-0.0013508	0.0044403	0.041685	0.041548
starwars.jpg	0.89457	-0.065962	0.012616	0.037108	0.040196	-0.0097071
starwarspoint.jpg	0.89205	0.016477	0.0092271	0.015002	-0.036723	-0.013595
white.png	1	0.012401	-0.035798	-0.046019	-0.056867	0.01161
whitepoint.png	1	-0.027617	-0.0021178	0.01271	-0.040577	0.034182

Table A.16 Diagonal correlation results for different Lorenz and reduction iteration values

File name	Diagonal correlation					
	Plain	**R1 L1**	**R16 L4**	**R32 L4**	**R16 L8**	**R32 L8**
armiehammer.jpg	0.96768	-0.009507	0.023892	0.033337	0.016667	0.013884
armiehammerpoint.jpg	0.9648	-0.029672	0.04535	0.050746	-0.0036287	0.021877
black.png	1	-0.039084	-0.0094551	-0.017274	0.011638	-0.0296
blackpoint.png	1	-0.090028	-0.060841	-0.0028129	0.00066461	0.00034049
blue.png	0.99548	0.044619	-0.026893	0.0063296	0.02297	-0.0413090
bluepoint.png	1	0.013964	-0.01235	0.035451	-0.0051413	0.035548
circle.png	0.97418	-0.06003	-0.067441	-0.0661	-0.087327	-0.0026801
circlepoint.png	0.90105	0.0053372	0.039066	0.041722	-0.020471	-0.016755
clarkgable.jpg	0.98728	-0.017521	-0.076593	-0.061427	0.044199	-0.036117
clarkgablepoint.jpg	0.989	0.032581	-0.0094444	-0.036805	-0.042715	0.030285
green.png	1	-0.033692	0.010916	0.0033064	0.0063996	0.0093297
greenpoint.png	0.99772	0.012334	0.020766	0.030318	-0.013827	0.020098
red.png	1	0.053575	-0.019138	0.029185	0.033726	0.0093219
redpoint.png	0.99767	0.003056	0.050552	0.0042605	0.0076269	-0.01698
square.png	0.99536	0.058219	0.0030057	0.041621	0.00067334	0.0053444
squarepoint.png	0.99102	-0.012269	0.032576	0.010746	-0.031991	-0.025595
starwars.jpg	0.94586	-0.065962	0.063418	0.01446	-0.014531	-0.00074958
starwarspoint.jpg	0.94777	0.016477	0.040883	-0.0074928	-0.039443	-0.0095675
white.png	1	0.012401	0.0081949	0.048084	-0.021178	0.036014
whitepoint.png	1	-0.027617	0.042036	-0.0025259	-0.032448	0.01224

Table A.17 Correlation between plain and cipher image results for different Lorenz and reduction iteration values

File name	**Correlation between plain – cipher images**					
	Plain	**R1 L1**	**R16 L4**	**R32 L4**	**R16 L8**	**R32 L8**
armiehammer.jpg	0.96768	-0.012936	0.020305	0.024883	0.0060137	-0.043453
armiehammerpoint.jpg	0.9648	0.016761	-0.062331	0.042994	0.039412	0.017927
black.png	1	NaN	NaN	NaN	NaN	NaN
blackpoint.png	1	NaN	NaN	NaN	NaN	NaN
blue.png	0.99548	-0.010791	0.017769	0.00066421	0.043456	-0.027429
bluepoint.png	1	0.025307	-0.021526	-0.030704	0.0017209	-0.011705
circle.png	0.97418	0.03498	-0.027538	-0.015097	0.027274	0.0026059
circlepoint.png	0.90105	-0.03202	-0.0086741	-0.031874	0.022296	0.025138
clarkgable.jpg	0.98728	-0.0091124	-0.010326	-0.03673	-0.023413	-0.013283
clarkgablepoint.jpg	0.989	0.028396	0.053919	0.020311	0.0012155	-0.0087354
green.png	1	0.016774	-0.02707	0.013095	-0.023614	-0.024123
greenpoint.png	0.99772	-0.010957	-0.045113	0.057999	-0.054424	0.019152
red.png	1	0.02434	0.0051881	0.052721	0.031271	-0.044213
redpoint.png	0.99767	-0.00057383	-0.021671	-0.027901	-0.0035804	0.0027865
square.png	0.99536	0.069721	0.0028025	-0.018279	-0.006962	0.036696
squarepoint.png	0.99102	0.045297	-0.021537	0.087692	-0.067688	-0.019512
starwars.jpg	0.94586	-0.021218	0.038358	-0.00098637	-0.026597	0.0040371
starwarspoint.jpg	0.94777	-0.010651	0.029645	-0.032959	-0.048909	0.029505
white.png	1	NaN	NaN	NaN	NaN	NaN
whitepoint.png	1	NaN	NaN	NaN	NaN	NaN

Table A.18 Correlation between cipher original image and cipher one pixel changed image results for different Lorenz and reduction iteration

File name	Correlation between original – pointed images					
	Plain	**R1 L1**	**R16 L4**	**R32 L4**	**R16 L8**	**R32 L8**
armiehammer.jpg	0.96768	0.5371	0.015267	0.028768	-0.030162	-0.0011326
armiehammerpoint.jpg	0.9648	0.5371	0.015267	0.028768	-0.030162	-0.0011326
black.png	1	1	NaN	NaN	NaN	NaN
blackpoint.png	1	1	NaN	NaN	NaN	NaN
blue.png	0.99548	1	0.029258	-0.0082635	-0.060055	0.042641
bluepoint.png	1	1	0.029258	-0.0082635	-0.060055	0.042641
circle.png	0.97418	1	-0.048172	-0.067046	0.021385	0.0055362
circlepoint.png	0.90105	1	-0.048172	-0.067046	0.021385	0.0055362
clarkgable.jpg	0.98728	0.98447	-0.042448	0.029643	0.020411	-0.03698
clarkgablepoint.jpg	0.989	0.98447	-0.042448	0.029643	0.020411	-0.03698
green.png	1	1	-0.021102	0.075039	0.034551	0.033908
greenpoint.png	0.99772	1	-0.021102	0.075039	0.034551	0.033908
red.png	1	1	-0.03287	0.0095098	-0.045258	0.031225
redpoint.png	0.99767	1	-0.03287	0.0095098	-0.045258	0.031225
square.png	0.99536	1	0.051776	-0.036482	0.0085466	0.040617
squarepoint.png	0.99102	1	0.051776	-0.036482	0.0085466	0.040617
starwars.jpg	0.94586	0.7026	0.0090348	-0.033806	-0.013197	0.0071253
starwarspoint.jpg	0.94777	0.7026	0.0090348	-0.033806	-0.013197	0.0071253
white.png	1	1	NaN	NaN	NaN	NaN
whitepoint.png	1	1	NaN	NaN	NaN	NaN

B
Encryption Times and Correlation Test Results of Second Dataset

This appendix contains the results of some tests applied to the second dataset. These are process time changing according to iteration numbers test, the relationships tests between; correlation and Lorenz and reduction iterations, correlation of plain and cipher images and Lorenz and reduction iterations.

Lorenz and reduction iteration values selected for the test are the Reduction Iteration=16 and Lorenz Iteration=4, and Reduction Iteration=32 and Lorenz Iteration=8 values determined as optimal values. The data below are the values of the Lorenz parameters obtained from the switch and the key used in the test. The first key is used for the values Reduction Iteration=16 and Lorenz Iteration=4, and the second for the values Reduction Iteration=32 and Lorenz Iteration=8.

Key: E8-79-61-80-2D-68-96-26-85-CD-A4-91-C5-BD-77-DC
Lorenz Sigma: 10.27805244
Lorenz Beta: 2.409767347
Lorenz Rho: 27.12021164

Key: 28-90-84-C5-19-53-6A-B2-0D-71-32-4D-2E-37-66-9A
Lorenz Sigma: 9.440943538
Lorenz Beta: 2.572226881
Lorenz Rho: 27.20918576

Table B.1 Encryption times and correlation analysis results of one hundred sample image for Lorenz Iteration = 4 and Reduction Iteration = 16

#	File name	Image size	Cipher time (seconds)	Horizontal		Vertical		Diagonal		Corr between plain - cipher
				Plain	Cipher	Plain	Cipher	Plain	Cipher	
1	airplane.png	512×512	26.439	0.95793	-0.006434	0.95565	0.012387	0.93747	-0.045427	-0.0070332
2	apple.jpg	460×460	20.784	0.99567	0.024657	0.99715	0.018112	0.99429	-0.014556	-0.0067896
3	armiehammer.jpg	550×440	28.805	0.96333	-0.001928	0.96411	-0.016545	0.94566	0.012841	0.0250270
4	baboon.png	512×512	24.711	0.90609	0.045000	0.85347	-0.030188	0.82679	-0.010636	0.0293970
5	bacteria.jpg	750×500	35.005	0.97579	-0.050559	0.97647	0.036441	0.96492	0.021439	0.0426040
6	balls.jpg	800×600	45.322	0.99447	0.012588	0.99396	0.009421	0.98827	0.022432	-0.0209990
7	barbara.png	512×512	24.853	0.88087	-0.007051	0.96115	-0.052622	0.86824	0.007828	-0.0192520
8	bee.jpg	728×424	29.069	0.99294	0.051102	0.99191	-0.035933	0.98809	-0.044190	-0.0238390
9	bird.jpg	716×478	32.049	0.99473	-0.034288	0.98602	-0.035685	0.98460	0.002699	-0.0112830
10	black.png	594×514	28.448	1.00000	0.023011	1.00000	-0.012498	1.00000	-0.027602	NaN
11	blue.png	594×514	30.271	1.00000	0.032565	1.00000	-0.040292	1.00000	-0.050033	0.0436630
12	book.jpg	620×412	25.846	0.86994	0.002438	0.88761	-0.000375	0.78324	0.048252	-0.0223080
13	calibration.jpg	900×674	61.366	0.99475	-0.016841	0.99362	0.041591	0.99378	-0.001681	-0.0266100
14	candy.jpg	700×420	29.309	0.99100	0.017176	0.99253	0.012577	0.98872	0.026544	0.0289550
15	capuchin.jpg	640×426	26.962	0.97808	-0.044456	0.97729	-0.020686	0.96793	-0.036836	-0.0436600
16	cat.png	490×732	35.608	0.98219	0.020523	0.96966	0.042311	0.95164	-0.005390	-0.0002289
17	chair.jpg	710×710	50.582	0.98481	-0.033700	0.98923	0.010051	0.97373	-0.008607	-0.0240860
18	china.jpg	800×400	31.301	0.98579	-0.010261	0.98580	0.021194	0.97246	-0.040535	-0.0095828
19	chocolate.jpg	650×488	31.553	0.99265	-0.013788	0.98941	0.008711	0.98346	0.032085	-0.0124830
20	circle.png	594×514	30.589	0.85496	0.021399	0.89958	0.024744	0.88841	-0.010881	-0.0476020
21	city.jpg	590×350	19.878	0.95804	-0.016899	0.94687	0.061116	0.90590	0.019943	-0.0221420
22	clarkgable.jpg	600×400	23.462	0.98355	0.003239	0.98806	-0.029481	0.97746	0.052456	-0.0059357

Table B.1 Encryption times and correlation analysis results of one hundred sample image for Lorenz Iteration = 4 and Reduction Iteration = 16 (continues)

#	File name	Image size	Cipher time (seconds)	Horizontal		Vertical		Diagonal		Corr between plain - cipher
				Plain	Cipher	Plain	Cipher	Plain	Cipher	
23	coral.jpg	618×410	23.97	0.95462	-0.001566	0.95899	0.030289	0.93248	0.016936	-0.0340830
24	david.jpg	760×985	72.954	0.99357	0.022418	0.99315	0.004860	0.98707	0.026588	-0.0524930
25	desert.jpg	800×600	46.388	0.99002	0.051594	0.98219	0.044623	0.97439	-0.034433	0.0074091
26	dinosaur.jpg	662×366	23.13	0.93381	-0.002952	0.93470	-0.045180	0.87457	-0.010661	0.0348270
27	dolphin.jpg	768×514	38.249	0.98327	-0.025868	0.97681	-0.000986	0.97098	-0.011356	0.0123230
28	drop.jpg	916×610	52.696	0.98801	0.003029	0.99143	-0.003353	0.98350	0.013833	-0.0456680
29	duck.jpg	500×500	23.096	0.99823	-0.010363	0.99562	-0.038362	0.99584	0.004180	0.0375870
30	eagle.jpg	600×600	36.803	0.99006	0.015319	0.99357	0.004733	0.98416	-0.023733	-0.0129860
31	eggplant.jpg	400×400	15.824	0.97944	-0.013268	0.99361	-0.010490	0.98413	0.000508	0.0529670
32	egypt.jpg	800×600	54.06	0.84135	-0.000716	0.73864	0.076846	0.70613	0.021817	-0.0334420
33	food.jpg	538×360	19.73	0.98981	-0.008839	0.98782	-0.067629	0.97365	0.036237	-0.0311390
34	forrest.jpg	550×366	19.995	0.83399	0.016188	0.85741	-0.036000	0.74886	-0.035451	0.0322240
35	fruits.png	512×512	26.799	0.98270	0.043937	0.98162	-0.045301	0.97374	-0.068751	0.0063428
36	frymire.jpg	1100×1100	123.963	0.91490	-0.031004	0.88414	0.021568	0.81599	-0.072076	-0.0124520
37	girl.png	768×512	39.612	0.98390	0.057945	0.99450	0.050963	0.98481	0.046778	0.0011088
38	girlface.bmp	512×512	26.464	0.98275	-0.023147	0.98890	-0.063021	0.97393	0.009791	0.0232870
39	grass.jpg	590×350	20.499	0.96823	-0.007566	0.96757	-0.017711	0.94263	-0.021429	-0.0044638
40	green.png	594×514	30.284	0.99775	0.071528	1.00000	-0.028788	1.00000	-0.003753	0.0396230
41	horse.jpg	640×640	42.106	0.99556	0.043242	0.99698	0.040761	0.99550	0.034040	-0.0189490
42	hotdog.jpg	780×438	34.4	0.98764	-0.020413	0.97725	-0.016017	0.99409	-0.001716	0.0565490
43	india.jpg	800×436	34.504	0.92392	0.055076	0.91162	0.028277	0.88587	0.006666	0.0005569
44	juice.jpg	749×468	35.211	0.99560	0.040306	0.99228	0.048289	0.99192	-0.019925	0.0031366

Table B.1 Encryption times and correlation analysis results of one hundred sample image for Lorenz Iteration = 4 and Reduction Iteration = 16 (continues)

#	File name	Image size	Cipher time (seconds)	Horizontal		Vertical		Diagonal		Corr between plain - cipher
				Plain	Cipher	Plain	Cipher	Plain	Cipher	
45	lena.png	512×512	25.573	0.97340	0.008862	0.99084	0.008521	0.97113	0.027309	-0.0045884
46	library.jpg	300×458	13.373	0.70861	-0.003247	0.88700	0.004964	0.58112	0.056065	-0.0067344
47	lichtenstein.png	512×512	26.119	0.96530	-0.004982	0.96850	0.010007	0.94913	-0.011943	-0.0265800
48	lion.jpg	860×644	56.154	0.98875	0.035514	0.98902	0.004206	0.98000	0.041948	-0.0104020
49	map.jpg	642×600	38.484	0.92388	0.070277	0.91500	0.009099	0.86382	-0.003617	-0.0001988
50	marilyn.jpg	798×904	72.819	0.99748	-0.000661	0.99583	0.000925	0.99601	-0.009508	-0.0186050
51	monalisa.jpg	686×1024	130.423	0.94671	-0.010559	0.94440	0.031404	0.92877	0.008088	-0.0057367
52	monarch.png	768×512	76.127	0.96199	0.029732	0.97340	0.003360	0.96512	0.024696	0.0080283
53	monk.jpg	600×400	46.465	0.96884	-0.007381	0.97627	0.024231	0.94787	0.041006	-0.0236630
54	mountain.png	640×480	56.017	0.86045	-0.033837	0.86242	0.004769	0.81820	0.013622	-0.0561460
55	musketeers.jpg	590×350	39.458	0.93414	0.052271	0.90582	0.060681	0.88099	-0.001622	0.0159110
56	northlights.jpg	750×472	67.465	0.99541	-0.043842	0.98926	0.026355	0.98974	-0.007397	-0.0069723
57	norvegia.jpg	480×328	30.784	0.95880	-0.022653	0.93762	-0.044249	0.91272	0.020384	-0.0989550
58	orchestra.jpg	900×598	105.506	0.91702	-0.062556	0.94233	0.031279	0.88351	0.033870	-0.0216100
59	orchid.jpg	726×408	57.397	0.99144	0.021257	0.99156	0.027458	0.98545	-0.029065	-0.0241460
60	orhun.png	598×346	25.329	0.97330	0.010271	0.93362	-0.024244	0.93200	0.073242	-0.0442460
61	ottoman.jpg	768×432	29.555	0.92429	0.004489	0.91582	-0.028513	0.88355	-0.017429	0.0233930
62	owl.jpg	550×366	17.648	0.98246	0.029141	0.98148	-0.005271	0.97424	-0.010662	0.0012667
63	paint.jpg	530×376	18.142	0.91333	0.002744	0.90861	-0.021743	0.90383	-0.016911	0.0326960
64	parrotfish.jpg	363×420	23.741	0.94587	-0.017311	0.93901	-0.019893	0.92408	0.029986	0.0190070
65	peacock.jpg	1544×1368	194.936	0.91447	0.023501	0.90072	0.016331	0.85933	-0.022365	0.0096988
66	peas.jpg	1280×1024	117.938	0.98957	-0.021350	0.98823	0.060916	0.98846	-0.031331	0.0171490

Table B.1 Encryption times and correlation analysis results of one hundred sample image for Lorenz Iteration = 4 and Reduction Iteration = 16 (continues)

#	File name	Image size	Cipher time (seconds)	Horizontal		Vertical		Diagonal		Corr between plain - cipher
				Plain	Cipher	Plain	Cipher	Plain	Cipher	
67	pencil.jpg	1024×768	82.855	0.99720	0.010976	0.99620	0.045389	0.99455	-0.018665	-0.0140490
68	penguin.jpg	600×400	22.4	0.98493	0.008487	0.97266	0.009178	0.96321	0.013759	-0.0413370
69	peppers.png	512×512	24.566	0.97912	-0.035132	0.98587	-0.049621	0.98031	-0.023436	0.0560420
70	pharaoh.jpg	750×498	34.563	0.96143	-0.050503	0.94743	0.036169	0.93812	-0.033020	-0.0247550
71	pixar.jpg	740×416	28.795	0.98457	0.042483	0.99317	0.022934	0.98336	0.007800	0.0264320
72	pool.png	510×382	17.788	0.95908	-0.052235	0.98115	0.002204	0.95390	-0.014697	0.0271250
73	power.jpg	864×434	35.14	0.97358	0.012174	0.96399	-0.016889	0.95425	-0.026613	0.0015942
74	ratatouille.jpg	1558×1458	213.099	0.99551	-0.004680	0.99660	-0.002493	0.99507	0.048927	-0.0206510
75	red.png	594×514	27.088	0.99778	0.020216	1.00000	0.006251	1.00000	-0.002000	-0.0476420
76	river.png	532×346	16.813	0.95207	0.017322	0.95249	0.022280	0.91845	-0.011016	-0.0096595
77	rogue.jpg	606×850	46.074	0.98504	0.024535	0.98545	-0.049796	0.97626	-0.002881	-0.0299620
78	sails.png	768×512	37.392	0.91435	0.002090	0.92472	0.003954	0.88104	0.008835	0.0010806
79	sea.jpg	782×604	47.305	0.99423	0.010716	0.93005	-0.011307	0.92306	0.069184	-0.0078715
80	serrano.png	628×794	48.203	0.94700	0.033601	0.94355	0.013682	0.94523	0.004277	0.0601630
81	shawshank.jpg	928×522	46.224	0.98485	-0.019834	0.98810	-0.003287	0.97719	-0.041551	-0.0153360
82	ship.jpg	1024×656	63.705	0.88793	-0.038767	0.75330	-0.044983	0.71275	-0.020578	0.0181720
83	slave.jpg	768×512	36.971	0.97236	0.004964	0.98709	-0.007741	0.95959	0.050958	0.0221940
84	snake.jpg	652×436	26.684	0.99494	-0.040067	0.99502	0.010816	0.99125	-0.001319	-0.0533720
85	spaceship.jpg	450×336	14.427	0.98984	0.013035	0.99138	0.029783	0.98510	-0.044202	-0.0076570
86	square.png	594×514	30.049	0.99777	0.025176	0.99082	0.032781	0.98450	-0.020945	-0.0379670
87	starwars.jpg	540×352	18.233	0.94202	-0.046844	0.94884	-0.011643	0.88358	-0.015273	0.0774290
88	sunflower.jpg	720×480	34.94	0.99079	-0.019216	0.99101	-0.087610	0.97869	0.034940	0.0244890

Table B.1 Encryption times and correlation analysis results of one hundred sample image for Lorenz Iteration = 4 and Reduction Iteration = 16 (continues)

#	File name	Image size	Cipher time (seconds)	Horizontal		Vertical		Diagonal		Corr between plain - cipher
				Plain	Cipher	Plain	Cipher	Plain	Cipher	
89	tiger.jpg	640×480	29.516	0.97344	0.015511	0.96747	-0.003511	0.94706	0.022439	0.0110150
90	toddler.png	934×630	57.641	0.99765	0.005552	0.99758	-0.057516	0.99679	0.025535	0.0714440
91	town.jpg	758×422	31.099	0.93382	-0.042834	0.94350	0.076269	0.91233	-0.040798	0.0066932
92	toy.jpg	666×666	43.033	0.99780	0.002875	0.99580	0.013309	0.99395	0.014100	-0.0073478
93	tree.jpg	658×438	27.74	0.91659	0.002171	0.92289	-0.032563	0.90882	0.002891	0.0008062
94	triangle.png	650×304	19.182	0.79666	-0.032754	0.78544	-0.033624	0.58226	0.067564	-0.0004005
95	tukan.jpg	800×570	44.122	0.99246	0.017674	0.97896	-0.022171	0.97823	-0.041272	0.0017194
96	tulips.png	768×512	772.323	0.98257	-0.045139	0.98379	0.004233	0.97186	-0.022503	-0.0396740
97	victorhugo.jpg	628×304	19.384	0.98967	-0.034204	0.99414	0.075189	0.97890	0.026425	0.0107210
98	village.jpg	700×442	28.816	0.94153	-0.008038	0.93062	0.026116	0.90943	-0.008171	-0.0127610
99	white.png	594×514	28.209	1.00000	0.012499	1.00000	0.014360	1.00000	-0.047333	NaN
100	wine.jpg	650×366	21.979	0.95439	-0.014778	0.96765	-0.017715	0.93174	-0.012882	-0.0206840
Average:			48.7876	0.96045	0.001491	0.95922	0.001757	0.93669	0.000211	-0.0026736
Median:			31.099	0.98219	0.002171	0.97896	0.004233	0.97098	-0.001716	-0.0059357
Total time:			4878.76							

Table B.2 Encryption times and correlation analysis results of one hundred sample image for Lorenz Iteration = 8 and Reduction Iteration = 32

#	File name	Image size	Cipher time (seconds)	Horizontal		Vertical		Diagonal		Corr between plain - cipher
				Plain	Cipher	Plain	Cipher	Plain	Cipher	
1	airplane.png	512×512	48.648	0.94584	0.018926	0.95195	0.003964	0.92255	-0.053451	-0.0257330
2	apple.jpg	460×460	40.829	0.99706	0.026324	0.99739	0.041584	0.99434	0.048582	-0.0008920
3	armiehammer.jpg	550×440	44.354	0.96406	-0.011906	0.96748	-0.007837	0.94363	-0.014297	-0.0469850
4	baboon.png	512×512	48.245	0.90766	-0.015957	0.86668	0.019916	0.81752	-0.016742	-0.0025910
5	bacteria.jpg	750×500	70.615	0.97437	-0.025889	0.97798	0.007772	0.96919	-0.016560	-0.0147800
6	balls.jpg	800×600	89.261	0.99233	0.023852	0.99255	-0.026077	0.98697	-0.056551	-0.0168250
7	barbara.png	512×512	48.427	0.89593	-0.010317	0.95944	-0.037250	0.87763	-0.010789	-0.0273410
8	bee.jpg	728×424	61.333	0.99394	0.005457	0.99092	0.024061	0.98743	-0.020660	0.0241180
9	bird.jpg	716×478	64.298	0.99330	0.018071	0.99171	-0.040538	0.97247	-0.016948	-0.0324660
10	black.png	594×514	56.905	1.00000	-0.004973	1.00000	0.019539	1.00000	-0.058987	NaN
11	blue.png	594×514	56.445	1.00000	-0.059342	1.00000	-0.046727	0.99778	-0.004780	-0.0427060
12	book.jpg	620×412	47.153	0.85611	0.001413	0.86571	0.012035	0.77270	0.059961	0.0416530
13	calibration.jpg	900×674	107.511	0.99448	0.023455	0.99805	-0.005818	0.99172	0.027026	-0.0128270
14	candy.jpg	700×420	52.772	0.99584	-0.043279	0.99405	-0.017434	0.98732	0.001489	0.0009408
15	capuchin.jpg	640×426	49.553	0.98067	-0.049934	0.96716	-0.027462	0.97174	-0.016245	-0.0033946
16	cat.png	490×732	64.276	0.98223	0.050552	0.96818	0.014692	0.95863	0.034711	0.0250010
17	chair.jpg	710×710	89.607	0.98536	-0.012612	0.98138	-0.016599	0.98038	-0.002669	-0.0052269
18	china.jpg	800×400	56.776	0.98723	0.015936	0.98973	-0.024519	0.97867	0.013784	0.0255690
19	chocolate.jpg	650×488	60.998	0.99221	0.011319	0.98934	-0.003065	0.98350	0.001646	0.0080346
20	circle.png	594×514	57.671	0.85789	-0.010614	0.78700	-0.033174	0.89559	0.025097	-0.0670780
21	city.jpg	590×350	39.389	0.94558	0.028911	0.94317	-0.007764	0.91121	-0.012155	0.0502060
22	clarkgable.jpg	600×400	45.927	0.98583	-0.046643	0.98727	0.004690	0.96586	0.004542	-0.0047477

Table B.2 Encryption times and correlation analysis results of one hundred sample image for Lorenz Iteration = 8 and Reduction Iteration = 32 (continues)

#	File name	Image size	Cipher time (seconds)	Horizontal		Vertical		Diagonal		Corr between plain - cipher
				Plain	Cipher	Plain	Cipher	Plain	Cipher	
23	coral.jpg	618×410	47.6	0.94999	-0.004685	0.94118	0.015837	0.93681	0.022079	0.0160430
24	david.jpg	760×985	147.009	0.99476	0.014619	0.99494	-0.012202	0.98985	-0.038805	0.0529480
25	desert.jpg	800×600	94.524	0.99089	0.075086	0.98099	-0.053118	0.97712	-0.001890	-0.0131430
26	dinosaur.jpg	662×366	47.621	0.93947	-0.007181	0.95009	-0.049120	0.90430	0.048815	0.0316390
27	dolphin.jpg	768×514	78.247	0.98626	0.000643	0.97981	0.023364	0.95645	0.019502	0.0186720
28	drop.jpg	916×610	109.877	0.99353	-0.020939	0.98661	0.048337	0.98780	-0.055516	0.0198430
29	duck.jpg	500×500	48.718	0.99150	0.005030	0.99722	-0.018772	0.99429	0.003646	0.0033059
30	eagle.jpg	600×600	68.989	0.98878	-0.039286	0.98967	-0.021670	0.98765	-0.000862	-0.0652550
31	eggplant.jpg	400×400	31.842	0.98204	-0.027998	0.99256	0.023342	0.97835	0.008702	-0.0343630
32	egypt.jpg	800×600	92.202	0.83425	0.016598	0.72181	0.008154	0.70881	0.039069	0.0271040
33	food.jpg	538×360	38.254	0.98998	0.012827	0.98898	0.015468	0.97918	-0.025429	0.0145820
34	forrest.jpg	550×366	38.447	0.82421	-0.009643	0.86495	-0.034494	0.79629	0.009748	-0.0283120
35	fruits.png	512×512	49.821	0.98509	0.037028	0.97529	0.010428	0.97872	0.005201	0.0722370
36	frymire.jpg	1100×1100	235.56	0.89852	0.011719	0.85830	-0.040540	0.81056	-0.002079	-0.0138200
37	girl.png	768×512	80.346	0.98952	0.011772	0.99248	-0.027428	0.98673	-0.001441	0.0231490
38	girlface.bmp	512×512	52.948	0.98371	0.022021	0.98970	-0.004616	0.97122	-0.004705	0.0138530
39	grass.jpg	590×350	41.204	0.97094	-0.035242	0.96431	0.040776	0.94400	0.015024	-0.0019134
40	green.png	594×514	62.642	1.00000	0.012017	1.00000	-0.014408	1.00000	-0.047334	0.0467970
41	horse.jpg	640×640	82.913	0.99513	-0.031072	0.99693	0.001985	0.99095	-0.022709	0.0030326
42	hotdog.jpg	780×438	68.262	0.99581	0.049807	0.98702	0.006112	0.98320	0.020290	-0.0095107
43	india.jpg	800×436	68.988	0.90867	0.035328	0.94280	-0.031014	0.88687	0.032118	0.0034823
44	juice.jpg	749×468	69.48	0.99596	-0.014194	0.99352	0.007072	0.98775	0.000032	-0.0241710

Table B.2 Encryption times and correlation analysis results of one hundred sample image for Lorenz Iteration = 8 and Reduction Iteration = 32 (continues)

#	File name	Image size	Cipher time (seconds)	Horizontal		Vertical		Diagonal		Corr between plain - cipher
				Plain	Cipher	Plain	Cipher	Plain	Cipher	
45	lena.png	512×512	52.777	0.97761	-0.055165	0.98924	-0.074067	0.97601	0.040532	0.0342540
46	library.jpg	300×458	27.826	0.69678	0.051672	0.88076	0.057547	0.62666	-0.018996	-0.0154760
47	lichtenstein.png	512×512	51.753	0.97648	0.047056	0.98330	0.053928	0.94746	0.012227	-0.0053655
48	lion.jpg	860×644	122.499	0.98631	0.020711	0.98959	0.082717	0.97584	0.016941	0.0393100
49	map.jpg	642×600	75.56	0.92901	0.022895	0.91138	-0.019811	0.86736	-0.013899	0.0309330
50	marilyn.jpg	798×904	143.463	0.99748	-0.015327	0.99703	0.009321	0.99496	-0.032111	-0.0147450
51	monalisa.jpg	686×1024	126.031	0.94276	-0.027662	0.95237	0.020468	0.93076	-0.019815	0.0057992
52	monarch.png	768×512	70.036	0.96807	0.035163	0.95261	-0.021392	0.95316	0.022753	-0.0183710
53	monk.jpg	600×400	42.046	0.95474	0.039403	0.96424	0.026002	0.94221	0.006292	0.0096568
54	mountain.png	640×480	55.736	0.84477	-0.000445	0.85901	0.019266	0.80739	0.050959	-0.0659120
55	musketeers.jpg	590×350	36.247	0.94290	-0.034594	0.91648	0.027309	0.89202	0.045346	-0.0021551
56	northlights.jpg	750×472	65.84	0.99662	-0.017515	0.98657	0.026120	0.97942	-0.004995	0.0189690
57	norvegia.jpg	480×328	28.594	0.95680	-0.014406	0.94982	-0.002571	0.92160	-0.027711	0.0201080
58	orchestra.jpg	900×598	94.424	0.92846	-0.032681	0.94949	0.041127	0.86520	-0.001779	-0.0199880
59	orchid.jpg	726×408	52.809	0.99232	-0.034234	0.99345	-0.018990	0.98640	0.060436	0.0419890
60	orhun.png	598×346	35.952	0.97936	0.038090	0.93815	0.042616	0.93781	-0.082616	-0.0136460
61	ottoman.jpg	768×432	58.256	0.91692	-0.042170	0.90747	0.039638	0.88060	-0.038041	-0.0432820
62	owl.jpg	550×366	34.566	0.98118	-0.000838	0.97499	0.019959	0.97197	0.005873	0.0010162
63	paint.jpg	530×376	35.137	0.92821	-0.010739	0.90717	-0.015530	0.89116	0.052977	0.0324780
64	parrotfish.jpg	363×420	47.648	0.95219	-0.014442	0.94253	-0.027538	0.92050	0.003300	0.0013319
65	peacock.jpg	1544×1368	433.666	0.90144	-0.015410	0.91504	0.017474	0.86029	-0.044909	-0.0195130
66	peas.jpg	1280×1024	268.625	0.98447	0.004227	0.99510	0.025982	0.98944	0.030993	0.0581670

Table B.2 Encryption times and correlation analysis results of one hundred sample image for Lorenz Iteration = 8 and Reduction Iteration = 32 (continues)

#	File name	Image size	Cipher time (seconds)	Horizontal		Vertical		Diagonal		Corr between plain - cipher
				Plain	Cipher	Plain	Cipher	Plain	Cipher	
67	pencil.jpg	1024×768	159.392	0.99748	0.015106	0.99589	0.006490	0.99518	0.006660	-0.0053869
68	penguin.jpg	600×400	47.016	0.98843	0.017453	0.98585	-0.073941	0.96530	-0.000075	0.0065555
69	peppers.png	512×512	51.559	0.98087	-0.053260	0.97849	0.014425	0.96602	0.009071	-0.0212180
70	pharaoh.jpg	750×498	74.534	0.95625	-0.020017	0.95320	-0.014864	0.92212	-0.059670	-0.0223120
71	pixar.jpg	740×416	60.312	0.98848	-0.006964	0.99329	-0.001084	0.97842	-0.015990	0.0405360
72	pool.png	510×382	37.824	0.98957	-0.011905	0.97644	0.040721	0.97475	-0.015783	-0.0255270
73	power.jpg	864×434	74.344	0.95922	0.024706	0.97091	0.017679	0.94298	0.007610	-0.0225500
74	ratatouille.jpg	1558×1458	451.289	0.99649	0.030389	0.99566	0.025866	0.99098	0.004310	0.0606030
75	red.png	594×514	55.539	1.00000	-0.026423	1.00000	0.027523	1.00000	0.008268	-0.0248540
76	river.png	532×346	33.58	0.95476	0.035218	0.94663	0.026189	0.92934	0.037234	-0.0005069
77	rogue.jpg	606×850	98.942	0.98644	0.000588	0.98749	0.027657	0.97029	0.006120	-0.0262280
78	sails.png	768×512	75.773	0.93993	-0.029340	0.92161	-0.000274	0.87192	-0.014410	0.0468180
79	sea.jpg	782×604	91.235	0.99327	-0.047816	0.92667	-0.030082	0.92095	-0.046780	-0.0048165
80	serrano.png	628×794	95.758	0.95115	-0.014630	0.96269	0.021041	0.93749	0.026458	-0.0388350
81	shawshank.jpg	928×522	93.824	0.98979	-0.030993	0.99169	0.005932	0.98479	-0.017961	0.0607720
82	ship.jpg	1024×656	133.228	0.89518	-0.022572	0.75649	0.017478	0.69798	0.051341	0.0227980
83	slave.jpg	768×512	76.209	0.98480	0.024145	0.98773	-0.006998	0.97398	-0.027893	-0.0050700
84	snake.jpg	652×436	56.824	0.99585	-0.018451	0.99419	0.033978	0.99170	-0.018902	-0.0222790
85	spaceship.jpg	450×336	29.154	0.99210	-0.008344	0.99122	0.028269	0.98657	-0.002758	0.0380970
86	square.png	594×514	58.167	1.00000	0.011155	0.99323	-0.018804	0.99091	0.010942	0.0338060
87	starwars.jpg	540×352	35.874	0.93842	-0.015460	0.94774	-0.012078	0.88438	0.021998	-0.0358290
88	sunflower.jpg	720×480	67.063	0.99280	0.008622	0.98973	-0.047886	0.98655	-0.003636	-0.0073921

Table B.2 Encryption times and correlation analysis results of one hundred sample image for Lorenz Iteration = 8 and Reduction Iteration = 32 (continues)

#	File name	Image size	Cipher time (seconds)	Horizontal		Vertical		Diagonal		Corr between plain - cipher
				Plain	Cipher	Plain	Cipher	Plain	Cipher	
89	tiger.jpg	640×480	60.609	0.96917	-0.036952	0.97132	0.049845	0.94455	0.016245	0.0163000
90	toddler.png	934×630	110.421	0.99816	-0.006864	0.99830	-0.056507	0.99717	0.020104	0.0459260
91	town.jpg	758×422	58.05	0.94888	-0.031767	0.93182	-0.059133	0.91444	0.009823	-0.0062787
92	toy.jpg	666×666	85.526	0.99773	-0.025252	0.99388	-0.017445	0.98746	-0.024792	-0.0346420
93	tree.jpg	658×438	57.178	0.93216	0.053032	0.92806	-0.012967	0.88438	-0.009441	0.0510190
94	triangle.png	650×304	38.677	0.79449	0.050207	0.80979	0.003074	0.59247	-0.021777	0.0177220
95	tukan.jpg	800×570	90.165	0.98838	-0.005151	0.98229	-0.006472	0.97636	0.008271	-0.0122030
96	tulips.png	768×512	78.045	0.98076	0.009964	0.98678	0.018844	0.97647	-0.005376	-0.0181390
97	victorhugo.jpg	628×304	37.499	0.99173	-0.022823	0.99229	0.007215	0.98184	0.046993	-0.0012640
98	village.jpg	700×442	62.17	0.93034	-0.008710	0.93145	-0.061230	0.89910	-0.039162	-0.0180430
99	white.png	594×514	60.327	1.00000	-0.022625	1.00000	-0.037118	1.00000	0.024116	NaN
100	wine.jpg	650×366	47.48	0.93301	0.032685	0.97112	-0.071811	0.93288	0.022980	-0.0224390
Average:			76.59565	0.96088	-0.001625	0.95856	-0.000794	0.93672	0.000174	0.0016883
Median:			60.312	0.98297	-0.006007	0.98040	0.003519	0.97076	0.000760	-0.0020343
Total time:			7659.565							

References

[1] B. Jackson. (2019). How to optimize images for web and performance. (2019, April 01), [Online]. Available: `https://kinsta.com/blog/optimize-images-for-web/`.

[2] O. S. Faragallah, F. E. A. El-Samie, H. E. H. Ahmed, I. F. Elashry, M. H. Shahieen, E.-S. M. El-Rabaie, and S. A. Alshebeili, *Image encryption: a communication perspective*. CRC Press, 2013.

[3] F. Özkaynak, "Kaos tabanlı simetrik şifreleme sistemlerinin tasarım ve analizi," PhD thesis, YTU Graduate School of Natural and Applied Sciences, Istanbul, 2013.

[4] H. G. Liddell and R. Scott, *A greek-english lexicon*. New York: American Book Company, 1897.

[5] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[6] D. Kahn, *The Codebreakers: The comprehensive history of secret communication from ancient times to the internet*. Simon and Schuster, 1996.

[7] T. D. Vakfı. (2016). İslam ansiklopedisi - muamma. (2018, November 29), [Online]. Available: `https://islamansiklopedisi.org.tr/muamma`.

[8] T. D. Vakfı. (2016). İslam ansiklopedisi - kalkaşendî. (2018, November 29), [Online]. Available: `https://islamansiklopedisi.org.tr/kalkasendi`.

[9] T. D. Vakfı. (2016). İslam ansiklopedisi - ibnü'd-düreyhim. (2018, November 29), [Online]. Available: `https://islamansiklopedisi.org.tr/ibnud-dureyhim`.

[10] T. D. Vakfı. (2016). İslam ansiklopedisi - kindî, ya'kūb b. ishak. (2018, November 29), [Online]. Available: `https://islamansiklopedisi.org.tr/kindi-yakub-b-ishak`.

[11] T. D. Vakfı. (2016). İslam ansiklopedisi - siyâkat. (2018, November 29), [Online]. Available: `https://islamansiklopedisi.org.tr/siyakat`.

[12] D. Salomon, *Coding for data and computer communications*. Springer Science & Business Media, 2006.

[13] J. V. Leeuwen, A. Meyer, M. Nival, *et al.*, *Handbook of theoretical computer science: Algorithms and complexity*. MIT Press, 1990.

[14] N. L. Biggs, *Codes: an introduction to information communication and cryptography*. Springer, 2008.

[15] C. E. Shannon, “A mathematical theory of cryptography,” 1945, (Memorandum MM 45-110-02. Written by the Signal Security agency, “declassified” in 1957, a copy is held at the MIT Library).

[16] C. E. Shannon, “Communication theory of secrecy systems,” *Bell system technical journal*, vol. 28, no. 4, pp. 656–715, 1949.

[17] A. Webster and S. E. Tavares, “On the design of s-boxes,” in *Conference on the theory and application of cryptographic techniques*, Springer, 1985, pp. 523–534.

[18] H. Feistel, “Cryptography and computer privacy,” *Scientific american*, vol. 228, no. 5, pp. 15–23, 1973.

[19] C. M. Danforth. (2013). Mathematics of planet earth - chaos in an atmosphere hanging on a wall. (2018, November 29), [Online]. Available: `http://mpe.dimacs.rutgers.edu/2013/03/17/chaos-in-an-atmosphere-hanging-on-a-wall/`.

[20] E. N. Lorenz, “Deterministic nonperiodic flow,” *Journal of the atmospheric sciences*, vol. 20, no. 2, pp. 130–141, 1963.

[21] B. B. Mandelbrot, “The variation of certain speculative prices,” in *Fractals and scaling in finance*, Springer, 1997, pp. 371–418.

[22] B. Mandelbrot, “How long is the coast of britain? statistical self-similarity and fractional dimension,” *science*, vol. 156, no. 3775, pp. 636–638, 1967.

[23] B. B. Mandelbrot, *The fractal geometry of nature*. WH freeman New York, 1983, vol. 173.

[24] P.-L. Carmen and L.-R. Ricardo, “Notions of chaotic cryptography: Sketch of a chaos based cryptosystem,” in *Applied Cryptography and Network Security*, IntechOpen, 2012.

[25] E. N. Lorenz, *Predictability: does the flap of a butterfly’s wing in Brazil set off a tornado in Texas?* 3775. American Association for the Advancement of Science, 1972, vol. 156, pp. 636–638.

[26] D. Ruelle, *Chance and chaos*. Princeton University Press, 1993, vol. 11.

[27] G. Alvarez and S. Li, “Some basic cryptographic requirements for chaos-based cryptosystems,” *International journal of bifurcation and chaos*, vol. 16, no. 08, pp. 2129–2151, 2006.

[28] S.-J. Li, “Analyses and new designs of digital chaotic ciphers,” PhD thesis, Xi’an Jiaotong University, 2003.

[29] S. Farwa, T. Shah, N. Muhammad, N. Bibi, A. Jahangir, and S. Arshad, “An image encryption technique based on chaotic s-box and arnold transform,” *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 6, p. 360 364, 2017.

[30] J. Choi, S. Seok, H. Seo, and H. Kim, “A fast arx model-based image encryption scheme,” *Multimedia Tools and Applications*, vol. 75, no. 22, pp. 14 685–14 706, 2016.

[31] D. Hong, J.-K. Lee, D.-C. Kim, D. Kwon, K. H. Ryu, and D.-G. Lee, “Lea: A 128-bit block cipher for fast encryption on common processors,” in *International Workshop on Information Security Applications*, Springer, 2013, pp. 3–27.

[32] J. Li, Y. Xing, C. Qu, and J. Zhang, “An image encryption method based on tent and lorenz chaotic systems,” in *2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, IEEE, 2015, pp. 582–586.

[33] V. Rozouvan, “Modulo image encryption with fractal keys,” *Optics and lasers in engineering*, vol. 47, no. 1, pp. 1–6, 2009.

[34] Y. Sun, L. Chen, R. Xu, and R. Kong, “An image encryption algorithm utilizing julia sets and hilbert curves,” *PloS one*, vol. 9, no. 1, e84655, 2014.

[35] J. Scharinger and F. Pichler, “Efficient image encryption based on chaotic maps,” *Pattern Recognition*, vol. 96, pp. 159–170, 1996.

[36] J. Fridrich, “Image encryption based on chaotic maps,” in *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, IEEE, vol. 2, 1997, pp. 1105–1110.

[37] J. Fridrich, “Symmetric ciphers based on two-dimensional chaotic maps,” *International Journal of Bifurcation and chaos*, vol. 8, no. 06, pp. 1259–1284, 1998.

[38] G. Chen, Y. Mao, and C. K. Chui, “A symmetric image encryption scheme based on 3d chaotic cat maps,” *Chaos, Solitons & Fractals*, vol. 21, no. 3, pp. 749–761, 2004.

[39] N. Taneja, B. Raman, and I. Gupta, “Chaos based cryptosystem for still visual data,” *Multimedia Tools and Applications*, vol. 61, no. 2, pp. 281–298, 2012.

[40] A. Anto Steffi and D. Sharma, “An image encryption algorithm based on 3d lorenz map.,” *International Journal of Advanced Research in Computer Science*, vol. 4, no. 1, 2013.

[41] W. B. Pennebaker and J. L. Mitchell, *JPEG: Still image data compression standard*. Springer Science & Business Media, 1992.

[42] D. Salomon, *Data compression: the complete reference*. Springer Science & Business Media, 2004.

[43] S. Lian, J. Sun, and Z. Wang, “A novel image encryption scheme based-on jpeg encoding,” in *Proceedings. Eighth International Conference on Information Visualisation, 2004. IV 2004.*, IEEE, 2004, pp. 217–220.

[44] X. Niu, C. Zhou, J. Ding, and B. Yang, “Jpeg encryption with file size preservation,” in *2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, IEEE, 2008, pp. 308–311.

[45] D. Zhang and F. Zhang, “Chaotic encryption and decryption of jpeg image,” *Optik-International Journal for Light and Electron Optics*, vol. 125, no. 2, pp. 717–720, 2014.

[46] K. Jack, *Video demystified: a handbook for the digital engineer*. Elsevier, 2011.

Publications From the Thesis

Contact Information: yesimikikat@gmail.com

Conference Papers

F. Y. Ikikat and S. Yavuz, “Chaotic image encryption by using lorenz map and yuv color space transformation,” in 2018 International Conference on Data Science and Applications, ICONDATA, 2018, pp. 414–426.