Islamic University of Gaza
Deanery of Higher Studies
Faculty of Engineering
Computer Engineering Department

Novel Hybrid Cryptosystem Based on Quasi Group, Chaotic and ElGamal Cryptography

By
Eng. Heba Ali M. Abu Ghali

Supervisor
Prof. Mohammad A. Mikki

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Engineering

1432H (2011)

بسم الله الرحمن الرحيم

الجامعة الإسلامية – غزة
The Islamic University - Gaza

عمادة الدراسات العليا
هاتف داخلي: 1150

الرقم Ref ...ع/35/...
التاريخ Date ...14/06/2011...

نتيجة الحكم على أطروحة ماجستير

بناءً على موافقة عمادة الدراسات العليا بالجامعة الإسلامية بغزة على تشكيل لجنة الحكم على أطروحة الباحثة/ **هبه علي محمد أبو غالي** لنيل درجة الماجستير في **كلية *الهندسة*/ قسم هندسة الحاسوب** وموضوعها:

Novel Hybrid Cryptosystem Based on Quasi Group, Chaotic, and EIGamal Cryptography

وبعد المناقشة التي تمت اليوم الثلاثاء 12 رجب 1432هـ، الموافق 2011/06/14م الساعة الحادية عشرة صباحاً، اجتمعت لجنة الحكم على الأطروحة والمكونة من:

أ.د. محمد أمين مكّي	مشرفاً ورئيساً	.....
أ.د. حاتم محمود حماد	مناقشاً داخلياً	.....
د. أيمن أحمد أبو سمرة	مناقشاً داخلياً	.....

وبعد المداولة أوصت اللجنة بمنح الباحثة درجة الماجستير في **كلية *الهندسة*/قسم هندسة الحاسوب.**

واللجنة إذ تمنحها هذه الدرجة فإنها توصيها بتقوى الله ولزوم طاعته وأن تسخر علمها في خدمة دينها ووطنها.

والله ولي التوفيق،،،

عميد الدراسات العليا

د. زياد إبراهيم مقداد

P.O. Box 108, Rimal, Gaza, Palestine fax: +970 (8) 286 0800 فاكس Tel: +970 (8) 286 0700 ص.ب. 108 الرمال، غزة، فلسطين هاتف
public@iugaza.edu.ps www.iugaza.edu.ps


بسم الله الرحمن الرحيم

Dedication

To my beloved mother
To the soul of my father
To sisters and brothers
To all friends

Acknowledgements

Praise is to Allah, the Almighty for having guided me at every stage of my life.

This thesis is the result of years of work whereby I have been accompanied and supported by many people. It is wonderful that I now have the opportunity to express my gratitude to all of them.

This work would not have been possible without the constant encouragement and support I received from Prof. Mohammed Mikki, my advisor and mentor. I would like to express my deep and sincere gratitude to him. His understanding and personal guidance have provided a good basis for the present thesis.

I also extend my thanks to Prof. Hatem Hamad and Dr. Aiman Abusamra the members of the thesis discussion committee.

Also, I would like to take this opportunity to express my profound gratitude to my beloved family - my mother, brothers and sisters - without whom I would ever have been able to achieve so much. I especially wish to express my love for my mother, who did not only endure my manifold activities but also provided inspiration and support for my inclination to perfectionism. Only she knows how much I am indebted to her.

Last, but certainly not least, I want to thank my friends, for their moral support during this study.

TABLE OF CONTENTS

List of Figures

List of Tables

Novel Hybrid Cryptosystem Based on Quasi Group, Chaotic, and Elgamal Cryptography

By

Eng. Heba Ali M. Abu Ghali

Abstract

In this thesis, a hybrid cryptosystem as proposed, it consists of public and private key cryptosystems; it combines the advantages of these cryptosystem types. Elgamal public key cryptosystem is improved; its disadvantages have been eliminated. In the private key cryptosystem, we investigate the usage of chaotic maps in cryptography, their properties, such as sensitive dependency on initial conditions and system parameters, and random-like outputs, are similar to confusion and diffusion cryptography properties. Furthermore, quasi groups provide a powerful technique for generating a larger set of permutation transformations, so, we adopt the chaotic maps and quasi groups in our proposed cryptosystem.

Experimental results are explored in this thesis to demonstrate the efficiency of the proposed cryptosystem. We illustrate the security of the proposed cryptosystem using different data types, we confirm that the proposed cryptosystem is secure against each of the chosen ciphertext attacks, statistics analysis, plaintext attacks, differential attacks and brute force attacks, it destroys the plaintext characteristics. Furthermore, the sensitivity analysis proves the robustness and the high security level of the proposed cryptosystem. Also the scheme is highly secure against the entropy attacks.

We construct the private key cryptosystem to deal with a sound signal, it performs good results, and it gives high entropy and destroys the characteristics of the original signal.

We develop a novel image cryptosystem by using two color spaces, in order to work in image space components level, which means we change the values of red, green, blue, luminance and color-difference components, for each pixel. This method gives outstanding results; it maximizes the entropy to be very close to the theoretical value, also the results of other measures are very good.

It's clear from the measurements results and visual inspection, that the encrypted image contains new colors (which not existed in the original image), besides the new distribution of colors which appears in the encrypted image.

Key words: *Cryptography, Public key cryptosystem, Private key cryptosystem, Chaotic maps, Quasi groups' cryptography, Elgamal cryptosystem, Voice cryptosystem, Image cryptosystem.*

نظام تشفير جديد مهجن باستخدام أنظمة مختلفة

م. هبه علي محمد أبو غالي

ملخص البحث

أدى تطور الشبكات و تبادل المعلومات لظهور معضلة أمن المعلومات، حيث يتم إرسال بيانات مختلفة عبر الشبكات غير الآمنة، مما حث الباحثين على تطور خوارزميات جديدة في علم التشفير.

في هذه الأطروحة تقدم الباحثة طريقة تشفير جديدة ، فهي طريقة تشفير مهجنة من نظام التشفير المتناظر (تشفير المفتاح الخاص)و نظام التشفير غير المتناظر (نظام تشفير المفتاح العام)، و ذلك للاستفادة من خصائص النظامين، عبر نقل المفتاح الأساسي باستخدام التشفير غير المتناظر الذي يوفر تراسل آمن عبر الشبكات غير الآمنة، ثم إرسال باقي البيانات باستخدام التشفير المتناظر كونه أسرع.

أيضا طورت الباحثة نظام الجمل (Elgamal algorithm) بحيث تم التخلص من مساوئه، و استخدمناه في كنظام تشفير غير متناظر في النظام الهجين المقترح.

بالإضافة إلى ذلك اقترحت الباحثة نظام تشفير متناظر مبنى على النظريات الرياضية : الزمرة (Quasi group) و الخرائط العشوائية الفوضوية (Chaotic maps) ، و اظهر هذا النظام نتائج ممتازة، فهو مقاوم للهجمات الإحصائية و الهجمات التفاضلية و هجوم الانتروبي و الهجوم الوحشي (الكسر الأعمى) وكذلك هجوم النص المشفر المحدد و أيضا هجوم النص الصريح، حيث أن النظام يمحو خصائص النص الصريح (الأصلي).

تم تطبيق نظام التشفير المقترح على نصوص مختلفة و موجات صوتية و اثبت جدارته و فاعليته في كلتا الحالتين.

كما قدمت الباحثة نظام تشفير جديد خاص بالصور يتعامل و لأول مرة مع مكونات الصورة الدقيقة، حيث يقوم بتشفير كلا من المكونات التالية: اللون الأحمر و اللون الأخضر و اللون الأزرق و الفرق بين الألوان والإنارة (السطوع) ، كلا على حده باستخدام تقنية الزمرة والخرائط العشوائية الفوضوية، ثم يدمج المكونات بعد التشفير معا؛ هذا التشفير اظهر نتائج ممتازة حيث أن قيمة مقياس العشوائية الناتجة (entropy) مقاربة جدا للقيمة النظرية، أما باقي المقاييس فقيمها ممتازة. و من خلال التجارب و المقارنة البصرية بين الصور قبل و بعد التشفير، نلاحظ ظهور ألوان جديدة في الصور المشفرة لم تكون موجودة أصلا في الصور الأصلية، كما نلاحظ اختلاف توزيع الألوان في الصور المشفرة عما كانت عليه في الصور قبل التشفير، حيث تظهر الألوان مبعثرة و مشتتة تماما.

كلمات مفتاحية: *التشفير، التشفير المتناظر، التشفير غير المتناظر ، الخرائط العشوائية الفوضوية ، الزمرة، نظام تشفير الجمل، تشفير الصوت ، تشفير الصور.*

LIST OF ABBREVIATION

AES	Advanced Encryption Standard.
ATM	Automated teller machine.
CTAK	Ciphertext auto key.
DES	Data Encryption Standard.
ECC	Elliptic Curve Cryptography.
ECDLP	Elliptic Curve Discrete Logarithm Problem.
EVCS	Extended Visual Cryptography Scheme.
FPGA	Field-programmable Gate Array.
HAVAL	Hash of Variable Length.
HDTV	High-definition television.
IBM	International Business Machines.
IPSec	Internet Protocol Security.
MATLAB	Matrix Laboratory.
MD5	Message Digest Algorithm.
NPCR	Number of pixels change rate.
NSA	National Security Agency.
PGP	Pretty Good Privacy.
RC4	Rivest Cipher or alternatively, Ron's Code version 4.
RC5	Rivest Cipher or alternatively, Ron's Code version 5.
RGB	Red, Green, Blue color space.
RSA	Rivest, Shamir and Adleman public key encryption.
S/MIME	Secure Multipurpose Internet Mail Extensions.
SEA	Shuffle Encryption Algorithm.
SHA	Security Hash Algorithms.
SSH	Secure Shell.
SSL	Secure Sockets Layer.
SDTV	Standard Definition Television
TLS	Transport Layer Security.
TTP	Trusted Third Party.
VCS	Visual Cryptography Schema.
XOR	Exclusive-or operation.

YCbCr	Y is the luminance component and Cb And Cr are the blue-difference and red-difference chroma components.
$\mathbb{Z}_p^*$	$\mathbb{Z}$ denotes the set of integers. $\mathbb{Z}_p$ is the integers modulo p. $\mathbb{Z}_p^*$ multiplicative group of $\mathbb{Z}_p$

Chapter 1

Introduction

Security is one of fundamental importance in digital communication. Hence, Cryptography is one of the most important fields in computer security. It's a process of transmission data through unsecured channels, and only the authenticated receiver who has the legitimate key can read the encrypted messages which might be documents, phone conversations, images or other form of data, as demonstrated in Fig. 1.1

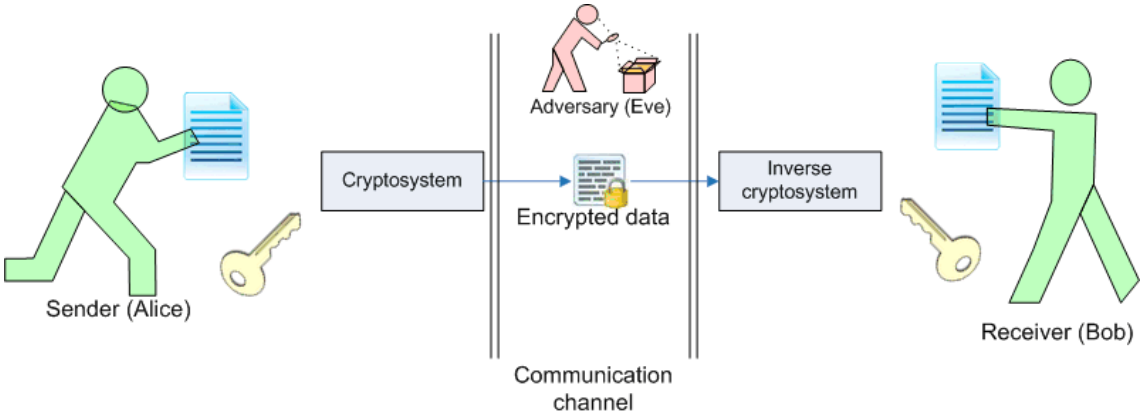


Fig. 1.1 General Cryptosystem.

Definition 1.1: *Cryptography* is the study of mathematical techniques related to aspects of information security such as confidentiality, data integrity, entity authentication, and data origin authentication. Cryptography is not the only means of providing information security, but rather one set of techniques.

Definition 1.2: *Cryptosystem* is a general term referring to a set of cryptographic primitives used to provide information security services. Most often the term is used in conjunction with primitives providing confidentiality, i.e., encryption.

In cryptosystems the information must be scrambled, so that other users will not be able to access the actual information. While providing privacy remains a central goal, the field has expanded to encompass many others, including not just other goals of communication security, such as guaranteeing integrity and authenticity of communications, but many more sophisticated and fascinating goals. Once largely the domain of the military, cryptography is now in widespread use, and you are likely to have used it even if you don't know it. When you shop on the Internet, for example to

buy a book, cryptography is used to ensure privacy of your credit card number as it travels from you to the shop's server. Or, in electronic banking, cryptography is used to ensure that your checks cannot be forged.

Cryptographic Goals

Cryptography services must guarantee the following:

1. *Confidentiality* is a service used to keep the content of information from all but those authorized to have it. *Secrecy* is a term synonymous with confidentiality and privacy. There are numerous approaches to providing confidentiality, ranging from physical protection to mathematical algorithms which render data unintelligible.

2. *Data integrity* is a service which addresses the unauthorized alteration of data. To assure data integrity, one must have the ability to detect data manipulation by unauthorized parties. Data manipulation includes such things as insertion, deletion, and substitution.

3. *Authentication* is a service related to identification. This function applies to both entities and information itself. Two parties entering into a communication should identify each other. Information delivered over a channel should be authenticated as to origin, date of origin, data content, time sent, etc. For these reasons this aspect of cryptography is usually subdivided into two major classes: *entity authentication* and *data origin authentication*. Data origin authentication implicitly provides data integrity (for if a message is modified, the source has changed), i.e. No one should be able to send a message to Bob and pretend to Alice (*data origin authentication*). When initiating a communication, Alice and Bob should be able to identify each other (*entity authentication*)

4. *Non-repudiation* is a service which prevents an entity from denying previous commitments or actions. When disputes arise due to an entity denying that certain actions were taken, a means to resolve the situation is necessary. For example, one entity may authorize the purchase of property by another entity and later deny such authorization was granted. A procedure involving a trusted third party is needed to resolve the dispute [1, 2].

1.1 Information Security and Cryptography

Information security is the procedure which expresses all measures taken to prevent unauthorized use of electronic data, whether this unauthorized use takes the form of destruction, use, disclosure, modification, or disruption. Information security and cryptography are interconnected and share the common services of keeping the confidentiality, integrity and availability of the information.

In the encryption process, information security uses cryptography to shift the information into the cipher form which does not allow it to be comprehensible by unauthorized personnel.

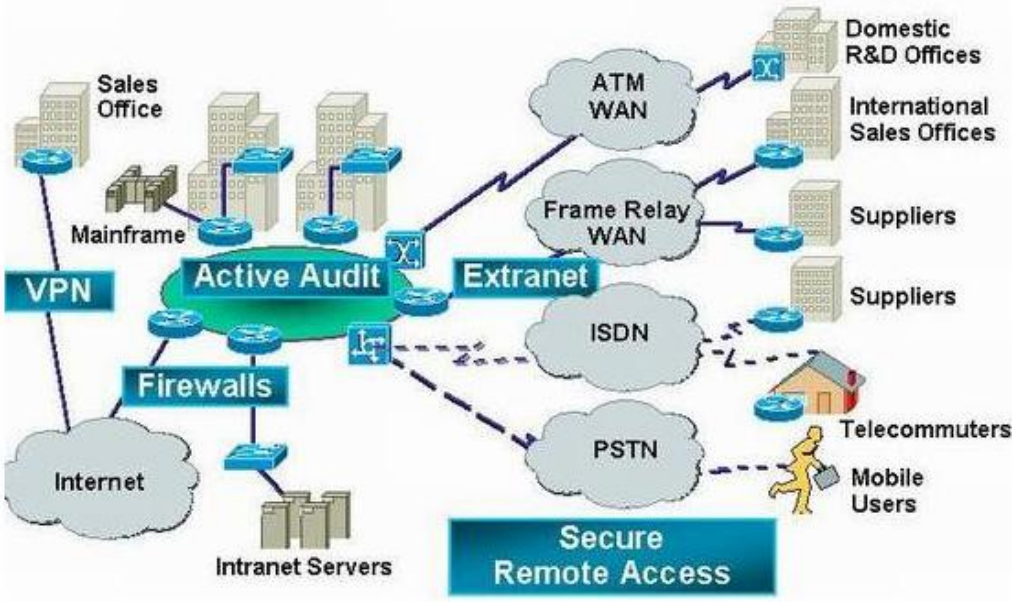


Fig. 1.2: Network security system

Cryptography provides the information security for other useful applications such as in encryption, message digests, and digital signatures. Cryptography is used in many applications encountered in everyday life such as: computer passwords, automated teller machine (ATM) cards, and electronic commerce as mentioned in Fig. 1.2. The request for cryptography system has increased recently for the public usage especially after the development of the Internet in the last years.

1.2 Motivation

With the huge growth of computer usage among modern people, there is more and more information transmitted over networks every day. This information may be banks transactions, business information or military information which should be

accessed only by authorized persons. In order to protect the confidential information secure cryptographic algorithms are required. There are two basic types of cryptosystems: *private key (symmetric key) systems* and *public key (asymmetric) systems.* Public key systems are much slower than private systems, but private systems require key agreement through an existing secure channel.

Private key systems use the same key for both encryption and decryption operations. In order to communicate securely using a private system, two parties must agree on the key using some pre-existing secure channel. When more than two parties are involved key distribution becomes even more complicated, and historically key distribution has been a major obstacle for practical uses of cryptography. Typical private ciphers use very convoluted transformations to obscure any patterns in the original message. The key controls how the transformations operate, and provides a map for reversing the transformations during decryption.

Public key cryptosystems help solve the key distribution problem by using separate keys for encryption and decryption operations, and making the encryption key public. Anyone can then encrypt a message, but only parties in possession of the private key can decrypt messages. Public key systems rely on one-way trap door functions, which are interesting mathematical functions that can be easily computed in one direction but are very difficult to reverse unless a secret key is known (the trapdoor). Since the encryption key is made public, finding the private decryption key from the public encryption key must be intractable. One application of public key cryptography is secure email. Public keys are typically published on a user's website. However if the user's website is compromised, a different public key corresponding to a malicious adversaries private key can be substituted.

The aim of this thesis is to propose new hybrid cryptosystem that provides benefits both private and public key cryptosystems, and can eliminate their previous drawbacks, In addition to be secure against statistical attack, differential attack, brute force attack and the plaintext attack.

1.3 Research Questions

The importance of this study can be summarized by four fundamental questions as follows:

- Why it is important to use both private and public cryptosystems together?

- What is the impact of the using of sophisticated approaches (chaotic and quasi group techniques) will be on the overall security?
- Which is the best way to combine chaotic and quasi group techniques in order to produce strong private key system?
- How much does the proposed hybrid cryptosystem increase the security? Could it holdup against the different types of attacks?

1.4 Problem Definition

As mentioned previously, most cryptosystems are either private key type which need a someway to exchange key, in addition to their private drawbacks , or its public key type which is mathematically more complicated and generally slower than private key cryptosystems , besides their private drawbacks.

Hence, we need a fast and secure cryptosystem; this can be achieved in a hybrid cryptosystem which consists of private key cryptosystem, which is fast and secured especially when we use a public key cryptosystem to exchange keys of private key cryptosystem.

The problem here is divided into three parts; first one is choosing the public key cryptosystem and improving it, if it needs enhancements. Second one is proposing a novel private key cryptosystem which involve chaotic maps and quasi group transpositions. The last one is building a hybrid cryptosystem that consists of the previous parts.

1.5 Research Scope

The research scope focuses on design an efficient private key cryptosystem using chaotic maps and quasi groups, in addition to solve the key distribution problem using public key cryptosystem that also provides user authentication before transmitting the confidential data.

1.6 Research Purpose

The objectives of the case studies are:

- To find an efficient public key cryptosystem
- To innovate a strong private key cryptosystem

In this thesis the security assessments of private key cryptosystem are based on:

1. The strength of the proposed scheme
    - Mathematical hard problem

- Randomize result
- Adequate key size

2. The performance of the proposed scheme**.**

1.7 Research Methodology

The purpose here is to generate a strong, secure and fast cryptosystem, using different techniques, and different strategies, comparing the result with previous systems to determine the best. So, a variety of schemas will be tested to obtain the best combination of chaotic and quasi group techniques, to be used as a private key system, and then a hybrid cryptosystem will be structured from the previous system and Elgamal cryptosystem as a public key cryptosystem.

In other words, to answer these research goals, this research employs the following research methodology as to:

- Show literature survey on Cryptography.
- Propose efficient cryptosystem by
    1. finding an efficient public key cryptosystem
    2. proposing a new private key cryptosystem based on chaotic maps and quasi groups
    3. Constructing a hybrid cryptosystem from the previous cryptosystem
- Compare our proposed cryptosystems with the others cryptosystems.
- Highlight the security results for the proposed cryptosystem.

1.8 Thesis Contributions

Practically everyone agrees that cryptography is an essential information security tool, and encryption can protect communications and stored information from unauthorized access and disclosure.

Cryptography allows people to carry over the confidence found in the physical world to the electronic world, thus allowing people to do business electronically without worries of deceit and deception. Everyday hundreds of thousands of people interact electronically, whether it is through e-mail, e-commerce, or ATM machines. The perpetual increase of information transmitted electronically has lead to an increased reliance on cryptography.

In this thesis, we involve mathematical theories in the encryption process to get the highest possible protection of encrypted data, and also a comparison between the available techniques - that are used in encryption - and our technique, is done, to see the improvement in the output.

The security protection of the proposed cryptosystem depends on the chaos properties and the added improvement of the output's entropy by involving the quasi group. In this way, cryptography benefits from the development in mathematics field.

We develop a hybrid cryptosystem that provides a secure way to distribute the key via the improved Elgamal cryptosystem, and then we use the proposed technique to encrypt the data in many forms: text, voice and images.

In this dissertation, we involve the image component colors and luminance in the encryption process; as a result of working with image in the color space level, the colors are distributed and new colors are created and existed, moreover the colors are distracted and are redistributed in the encrypted image.

The contributions of this study are highlighted below:

- We review several the cryptography techniques and types.
- We implement Elgamal cryptosystem and improve it.
- We produce a new private key cryptosystem based on chaotic maps and quasi groups.
- We investigate the impact of using chaotic maps and quasi groups in cryptography.
- We demonstrate through testing the efficiency and effectiveness of the proposed cryptosystem.
- We examine the encryption of image components level.
- We present a hybrid cryptosystem construct of private and public key cryptosystem.

1.9 Outline of the Thesis

The thesis is organized as follows:

In chapter 2, the fundamentals of cryptography are presented. The properties of private key, public key and hybrid cryptosystems are described. In addition, the concept of hash functions and digital signature are also introduced.

Chapter 3 surveys some related cryptosystems, and highlights its main shortcomings which are avoided in our work.

As pointed out before, the proposed hybrid cryptosystem is divided into primarily two category cryptosystems: first one is a public key cryptosystem namely Elgamal cryptosystem which is explained in the next chapter. The second part of hybrid cryptosystem is a private key cryptosystem; it's divided into 3 classes as shown in chapters 5, 6 and 7. The hybrid cryptosystem is revealed in chapter 8.

In chapter 4, the improved Elgamal cryptosystem is proposed and discussed. The security of the improved scheme is evaluated via both cryptanalysis and experiments.

In chapter 5, the methodology and implementation of the proposed private key cryptosystem is explained. Furthermore, security analysis of the proposed cryptosystem is made and the performance of the proposed cryptosystem is reported.

Chapter 6 explores the design and implementation of the proposed voice private key cryptosystem, In addition to investigation the simulation results and security analysis.

Chapter 7, illustrates the proposed image private key cryptosystem, and discusses the detailed security analysis of it including the performance evaluation of the proposed scheme.

In Chapter 8 we explain the idea of a proposed hybrid cryptosystem, also we demonstrate its implementation, and examine the results in order to present the overall security analysis.

Finally, concluding remarks of the thesis are given in chapter 9, this conclusion depends on all the results which we have obtained through the implementation of the study to be clarified and summarized for the reader. Future developments in the hybrid cryptosystem are suggested.

The appendices to the document provide additional relevant material. Appendix A provides an explanation of the mathematical basics used in the thesis. Appendix B gives you an idea about the color space used in the thesis, appendix C presents some hash functions, and appendix D elaborates some of the details of Diffie-Hellman.

Chapter 2

Overview of Cryptography

In this chapter, we introduce notation and basic principles in cryptography used in this thesis. The purpose of this chapter is to give a general idea of the principles, techniques, and algorithms which are required for understanding this thesis, but we rather assume the reader is familiar with cryptography.

2.1 Cryptography System Definition

The definitions of cryptography and cryptosystem are given in chapter 1. As mentioned, a cryptosystem is a system in which information can be made meaningless to all people except the intended reader. In a cryptosystem, the sender (usually called Alice) encrypts a plaintext to a ciphertext before sending and the receiver (usually called Bob) decrypts the ciphertext to obtain the plaintext, see Fig. 1.1

The term plaintext means the original message or information, which can be a company business plan or a personal medical record, etc., the encryption process make the original information unreadable to general readers. The ciphertext is the encryption process's output which is not recognizable. The decryption process converts the ciphertext back to plaintext for the intended readers.

In mathematical notation, the encryption and decryption transformations present hereunder:

- K denotes a set called the key space. An element of K is called a key.
- M denotes a set called the *message space*. M consists of strings of symbols from an alphabet of definition. An element of M is called a *plaintext message* or simply a *plaintext*. For example, M may consist of binary strings, English text, computer code, etc.
- C denotes a set called the *ciphertext space*. C consists of strings of symbols from an alphabet of definition, which may differ from the alphabet of definition for M. An element of C is called a ciphertext.
- Each element $e \in K$ uniquely determines a bijection from M to C, denoted by Ee, (i.e., $Ee : M \to C$). Ee is called an encryption function or an encryption transformation. Note that Ee must be a bijection if the process is to be reversed and a unique plaintext message recovered for each distinct ciphertext.

- For each $d \in K$, D_d denotes a bijection from C to M (i.e., $D_d : C \rightarrow M$). D_d is called a decryption function or decryption transformation.
- The process of applying the transformation Ee to a message $m \in M$ is usually referred to as encrypting m or the encryption of m.
- The process of applying the transformation D_d to a ciphertext c is usually referred to as decrypting c or the decryption of c.
- An encryption scheme consists of a set $\{Ee : e \in K\}$ of encryption transformations and a corresponding set $\{D_d : d \in K\}$ of decryption transformations with the property that for each $e \in K$ there is a unique key $d \in K$ such $D_d = Ee^{-1}$; that is, D_d $(Ee\ (m))$ = m for all $m \in M$. An encryption scheme is sometimes referred to as a cipher.
- To construct an encryption scheme requires one to select a message space M, a ciphertext space C, a key space K, a set of encryption transformations $\{Ee : e \in K\}$, and a corresponding set of decryption transformations $\{D_d : d \in K$.

Note: The function f is bijective if it is both one-to-one (injective) and onto (subjective).

***Definition* 2.1** A function (or transformation) is $1-1$ *(one-to-one)* if each element in the co-domain Y is the image of at most one element in the domain X.

***Definition* 2.2** A function (or transformation) is *onto* if each element in the co-domain Y is the image of at least one element in the domain. Equivalently, a function $f : X \rightarrow Y$ is onto $Im(f) = Y$.

***Definition* 2.3** If a function $f : X \rightarrow Y$ is 1−1 and $Im(f) = Y$, then f is called a *bijection* [1, 3].

Cryptographic techniques are typically divided into two general types: private key cryptosystems and public key cryptosystems. In brief, the private key cryptosystem use the same key in encryption and decryption process, whereas the public key cryptosystem utilize different keys in encryption and decryption process.

Encryption methods of these types will be discussed separately in section 2.2 and 2.3, other definitions and terminology will be introduced as required.

2.2 Private Key Cryptosystems

Private key cryptosystem is also known as secret key, or one-key cryptosystems. The plaintext is encrypted by key *ek* and the ciphertext is decrypted by key *dk*, where *ek* is the encryption key and *dk* is the decryption key. In private key scheme, key *dk* must be equal to key *ek*, refer to Fig. 2.1; here the two parties must agree on the key using some pre-existing secure channel. When more than two parties are involved key distribution becomes even more complicated, and historically key distribution has been a major obstacle for practical uses of cryptography. Examples of private key systems include the Data Encryption Standard (DES) [4], the Advanced Encryption Standard (AES).

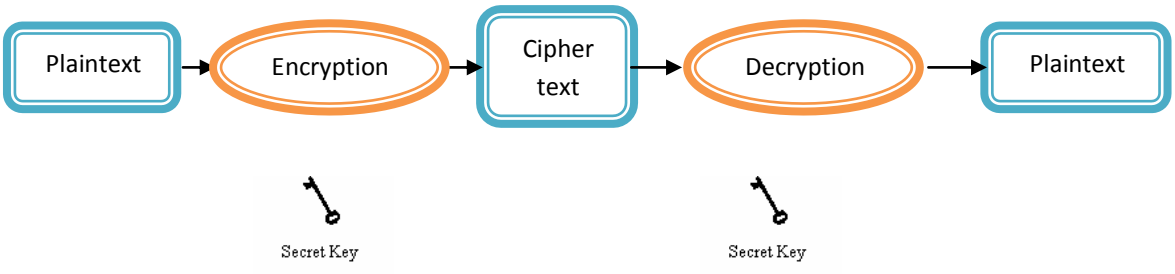


Fig. 2.1: Private key cryptosystem scheme

Definition 2.4 Consider an encryption scheme consisting of the sets of encryption and decryption transformations $\{ Ee : e \in K \}$ and $\{D_d : d \in K \}$, respectively, where K is the key space. The encryption scheme is said to be *private key* if for each associated encryption/decryption key pair $(e;d)$, it is computationally "easy" to determine d knowing only e, and to determine e from d.

Since $e = d$ in most practical private key encryption schemes, the term private key becomes appropriate. Other terms are used single-key, one-key, symmetric key, and conventional encryption.

Private key cryptosystems can be classified into stream ciphers and block ciphers. The stream ciphers encrypt the whole message at a time. While the block

ciphers split the message into blocks and then encrypt the blocks respectively. In general, the block size is normally 128 bits.

2.2.1 Stream Ciphers

A stream cipher is a type of private key encryption where the stream cipher algorithm generates key stream. The key stream bits and the plaintext bits are combined together, usually with the exclusive-or (XOR) operation, and then the result for this combination is the ciphertext. Stream ciphers are considered much faster with lower hardware cost than the block cipher algorithms [5].

Definition 2.5: Let K be the key space for a set of encryption transformations. A sequence of symbols $e_1\ e_2\ e_3$ ... $e_i \in K$ is called a key stream.

Definition 2.6 Let A be an alphabet of q symbols and let Ee be a simple substitution cipher with block length 1 where $e \in K$. Let $m_1\ m_2\ m_3$ ... be a plaintext string and let $e_1\ e_2\ e_3$ ... be a key stream from K. A *stream cipher* takes the plaintext string and produces a ciphertext string $c_1\ c_2\ c_3$ ... where $c_i = E_{ei}\ (m_i)$. If d_i denotes the inverse of e_i, then $D_{di}\ (c_i) = m_i$ decrypts the ciphertext string.[1]

In general the stream cipher can be classified (based on the key stream status) into two types as follows:

- Synchronous stream ciphers: the key stream in the synchronous stream ciphers is generated independently of the plaintext and ciphertext.
- Self-synchronizing stream ciphers: in the self-synchronous stream the key stream is generated dependent on the plaintext and the ciphertext. The self-synchronizing stream cipher scheme is known also as asynchronous stream ciphers or ciphertext auto key (CTAK).

There are many proposed stream cipher cryptosystems such as RC4 [6], RC5 [7], Helix [8], ISAAC [9], etc.

2.2.2 Block Ciphers

A block cipher is a private key encryption that encrypts a fixed-length block of plaintext into same length of ciphertext block. The encryption and decryption in the

block cipher are performed by using the same secret key. Typically, a block cipher that takes the input 64-bit block of plaintext must produce the output 64-bit block of ciphertext.

Definition A *block cipher* is an encryption scheme which breaks up the plaintext messages to be transmitted into strings (called *blocks*) of a fixed length t over an alphabet A, and encrypts one block at a time.

The Data Encryption Standard (DES) is one of the private key schemes, and was developed in 1977 by International Business Machines (IBM) and the NSA (National Security Agency). DES processes plaintext blocks of n = 64 bits, producing 64-bit ciphertext blocks. The effective size for the secret key in DES is 56 bits [4].

Two important classes of block ciphers are *substitution ciphers* and *transposition ciphers*.

Substitution ciphers are block ciphers which replace symbols (or groups of symbols) by other symbols or groups of symbols.

Definition 2.7: Let A be an alphabet of q symbols and M be the set of all strings of length t over A. Let K be the set of all permutations on the set A. Define for each $e \in K$ an encryption transformation Ee as:

$Ee\ (m) = (e(m_1)e(m_2)\ \ldots\ e(m_t)) = (c_1c_2\ \ldots.\ c_t) = c;$

Where $= (m_1\, m_2\ \ldots\ m_t) \in M$. In other words, for each symbol in a t-tuple, replace (substitute) it by another symbol from A according to some fixed permutation e. To decrypt $c = (c_1c_2\ \ldots.\ c_t)$ compute the inverse permutation $d = e^{-1}$ and $D_d\ (c) = (d\ (c_1)\ d\ (c_2) \ldots\ d\ (c_t)) = (m_1m_2\ \ldots.\ m_t)) = m$. Ee is called a *simple substitution cipher*.

Transposition ciphers are the simple transposition ciphers, which simply permutes the symbols in a block.

Definition 2.8: Consider a private key block encryption scheme with block length t. Let K be the set of all permutations on the set $\{1, 2, 3\ \ldots\ t\}$. For each $e \in K$ define the encryption function $Ee(m) = (m_{e(1)}\ m_{e(2)} \ldots\ m_{e(t)})$ where $=$

$(m_1\, m_2\ \ldots\ m_t) \in M$, the message space. The set of all such transformations is called a *simple transposition cipher*. The decryption key corresponding to e is the inverse permutation $d = e^{-1}$. To decrypt $c = (c_1 c_2\ \ldots.\ c_t)$ compute $D_d(c) = (d(c_1)d(c_2)\ldots\ d(c_t))$.

A simple transposition cipher preserves the number of symbols of a given type within a block, and thus is easily crypt analyzed.

Note: A substitution is said to add *confusion* to the encryption process whereas a transposition is said to add *diffusion*.

Confusion is intended to make the relationship between the key and ciphertext as complex as possible.

Diffusion refers to rearranging or spreading out the bits in the message so that any redundancy in the plaintext is spread out over the ciphertext [1, 10].

2.3 Public Key Cryptosystems

Public key cryptosystems help solve the key distribution problem by using separate keys for encryption and decryption, and making the encryption key public. Anyone can then encrypt a message, but only parties in possession of the private key can decrypt messages. Public key systems rely on one-way trap door functions, which are interesting mathematical functions that can be easily computed in one direction but are very difficult to reverse unless a secret key is known (the trap door) (Definitions 2.10 and 2.11) below. In these cryptosystems, there is a pair of keys, one of which is known to the public and used to encrypt the plaintext to be sent to the receiver who owns the corresponding decryption key, known as the private key (refer to Fig. 2.2). In general, a security protocol uses public key cryptosystem to exchange the secret key between communicating nodes and then uses private key cryptosystems with the agreed secret key as the password to ensure confidentiality on the data transferred.

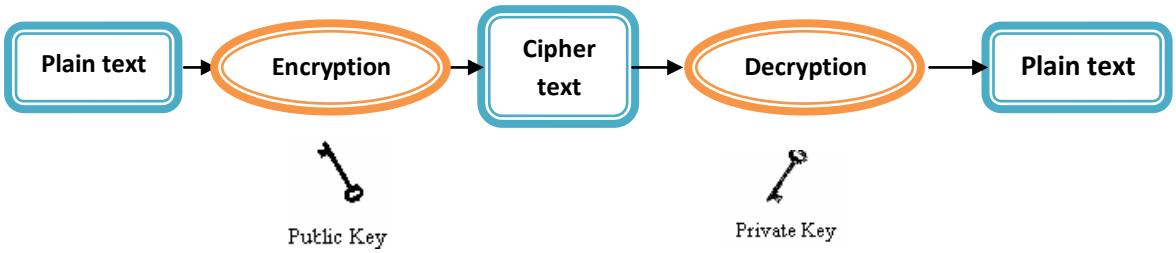


Fig. 2.2: Public key cryptosystem scheme

***Definition 2.9*:** A function f from a set X to a set Y is called a *one-way function* if $f(x)$ is "easy" to compute for all $x \in X$ but for "essentially all" elements $y \in Im(f)$ it is "computationally infeasible" to find any $x \in X$ such that $f(x) = y$

Definition 2.10: A *trapdoor one-way function* is a one-way function $f : X \to Y$ with the additional property that given some extra information (called the *trapdoor information*) it becomes feasible to find for any given $y \in Im(f)$ and $x \in X$ such that $f(x) = y$.

Definition 2.11: Consider an encryption scheme consisting of the sets of encryption and decryption transformations $\{ Ee : e \in K \}$ and $\{D_d : d \in K \}$, respectively. The encryption method is said to be a *public key encryption* scheme if for each associated encryption/decryption pair $(e; d)$, one key e (the public key) is made publicly available, while the other d (the private key) is kept secret. For the scheme to be secure, it must be computationally infeasible to compute d from e.

In more details: let $\{ Ee : e \in K \}$ be a set of encryption transformations, and let $\{ D_d : d \in K \}$ be the set of corresponding decryption transformations, where K is the key space. Consider any pair of associated encryption/decryption transformations $(Ee; D_d)$ and suppose that each pair has the property that knowing Ee it is computationally infeasible, given a random ciphertext $c \in C$, to find the message $m \in M$ such that $Ee(m) = c$. This property implies that given e it is infeasible to determine the corresponding decryption key d. (Of course e and d are simply means to describe the encryption and decryption functions, respectively.) Ee is being viewed here as a trapdoor one-way function with d being the trapdoor information necessary to compute the inverse function and hence allow decryption. This is unlike private key ciphers where e and d are essentially the same [1, 3].

Public key encryption, as mentioned before, assumes that knowledge of the public key does not allow computation of the private key. In other words, this assumes the existence of trapdoor one-way functions.

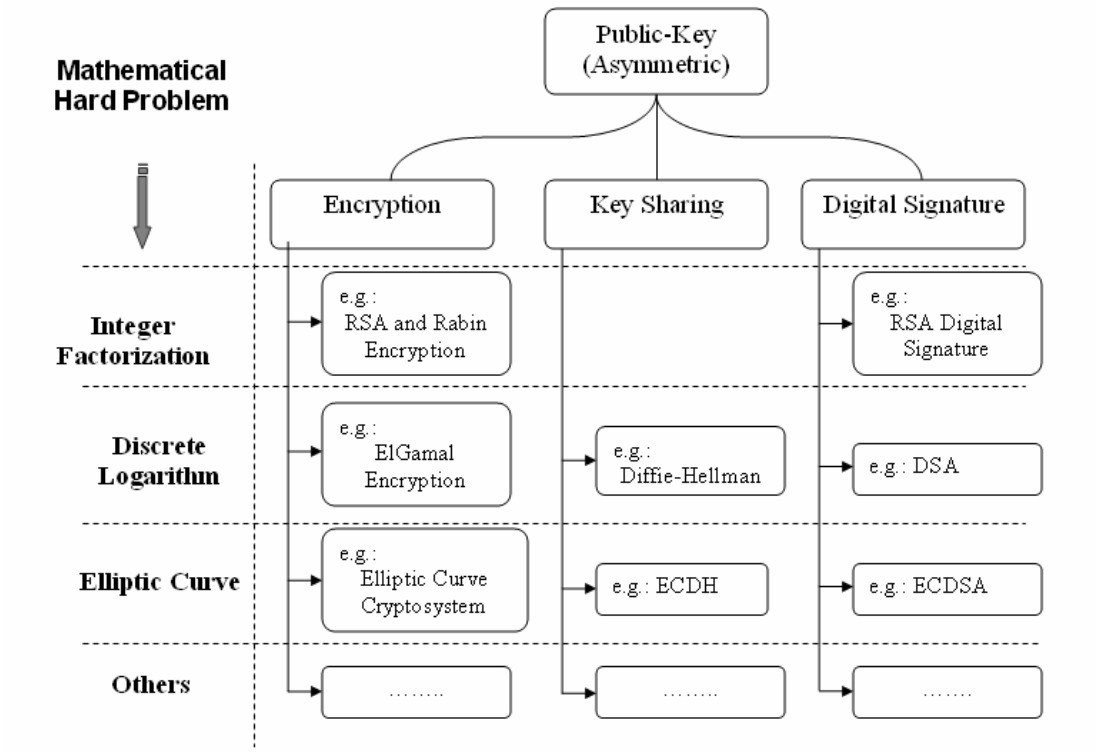


Fig. 2.3: Main branches of public key cryptosystem scheme

Fig. 2.3 shows the most important branches of public key cryptosystem which can be classified into three types: encryption, key sharing and digital signature. Each type is fragmented based on its mathematical hard problems (integer factorization, discrete logarithm and Elliptic Curve).

2.3.1 Digital Signatures

A major benefit of public key cryptography is that it provides a method for employing *digital signatures*. Digital signatures enable the recipient of information to verify the authenticity of the information's origin, and also verify that the information is intact. Thus, public key digital signatures provide *authentication* and *data integrity*. A digital signature also provides *non-repudiation*, which means that it prevents the sender from claiming that he or she did not actually send the information. These features are every bit as fundamental to cryptography as privacy, if not more.

A digital signature serves the same purpose as a handwritten signature. However, a handwritten signature is easy to counterfeit. A digital signature is superior to a handwritten signature in that it is nearly impossible to counterfeit, plus it attests to the contents of the information as well as to the identity of the signer.

Some people tend to use signatures more than they use encryption. For example, you may not care if anyone knows that you just deposited $1000 in your account, but you do want to be darn sure it was the bank teller you were dealing with. The basic manner in which digital signatures are created is illustrated in Fig. 2.4. Instead of encrypting information using someone else's public key, you encrypt it with your private key. If the information can be decrypted with your public key, then it must have originated with you.

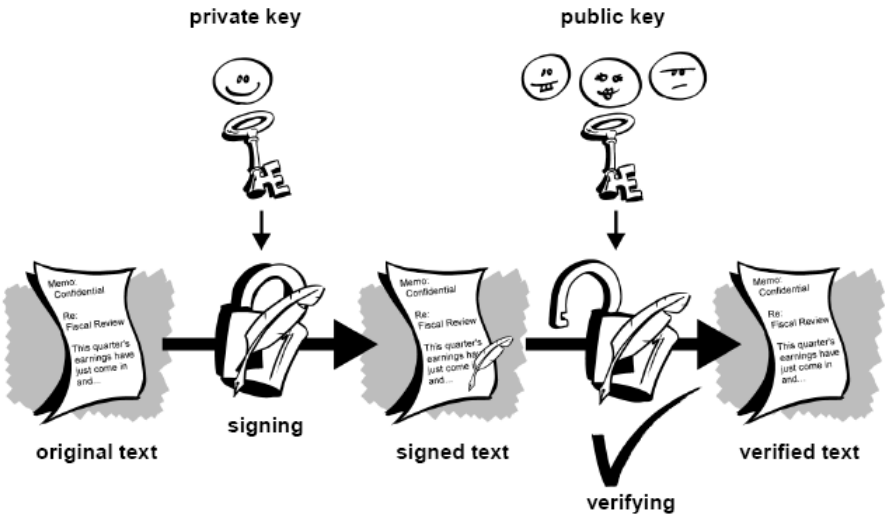


Fig. 2.4: Simple digital signatures.

Nomenclature and set-up**:**

- M is the set of messages which can be signed.
- S is a set of elements called *signatures*, possibly binary strings of a fixed length.
- S_A is a transformation from the message set M to the signature set S, and is called a *signing transformation* for entity A. The transformation S_A is kept secret by A, and will be used to create signatures for messages from M.
- V_A is a transformation from the set $M \rightarrow S$ to the set $\{true; false\}$. V_A is called a *verification transformation* for A's signatures, is publicly known, and is used by other entities to verify signatures created by A.

Note: The transformations S_A and V_A provide a *digital signature scheme* for A. Occasionally the term *digital signature mechanism* is used [1].

Signing procedure

Entity A (the signer) creates a signature for a message $m \to M$ by doing the following:

1. Compute $s = S_A(m)$.

2. Transmit the pair $(m; s)$. s is called the signature for message m.

Verification procedure

To verify that a signature s on a message m was created by A, an entity B (the verifier) performs the following steps:

1. Obtain the verification function V_A of A.

2. Compute $u = V_A(m; s)$.

3. Accept the signature as having been created by A if $u = true$, and reject the signature if $u = false$.

2.3.2 Public Key Encryption Based on Discrete Logarithm: Elgamal Scheme

Elgamal is a public key system which uses modular exponentiation as the basis for a one way trap door function. The reverse operation, the discrete logarithm, is considered intractable. Elgamal was never patented, making it an attractive alternative to the more well known RSA system. 1024 bits is the minimum recommended key size for Elgamal, and even larger keys are recommended for some applications.

An Elgamal cryptosystem operates in a finite cyclic group, which by convention is written multiplicatively. For simplicity we restrict our discussion to the two most common choices: the group of integers from 1 to $p - 1$ under multiplication mod p for some prime p, commonly called $\mathbb{Z}_p^*$, and subgroups of $\mathbb{Z}_p^*$ of prime order. We will use $|\alpha|$ to denote the order of an element α in $\mathbb{Z}_p^*$ and $\langle\alpha\rangle$ to denote the cyclic subgroup of $\mathbb{Z}_p^*$ generated by α. Unless otherwise noted, assume multiplications and exponentiations involving elements of $\mathbb{Z}_p^*$ are done $mod\ p$. I begin the discussion of Elgamal with the discrete log problem, since its intractability is central to the security of Elgamal [11].

2.3.2.1 The Discrete Log Problem

The standard logarithm is the inverse operation of standard exponentiation. Similarly we define the discrete logarithm to be the inverse of modular exponentiation: given a modular exponentiation $y = \alpha^a$ in $\mathbb{Z}_p^*$ and the base α, the discrete logarithm $\log_\alpha y$ is a. This is a discrete logarithm in the cyclic group $\langle\alpha\rangle$ which may or may not be all of $\mathbb{Z}_p^*$. When $|\alpha|$ = n is large and has at least one large prime factor, discrete log problems in $\langle\alpha\rangle$ are considered intractable.

There are three basic types of discrete log algorithms: "square-root" algorithms such as Pollard's rho algorithm, the Pohlig-Hellmen algorithm, and index calculus algorithms.

Pollard's rho algorithm can compute discrete logs in a cyclic group of prime order n in time $O(\sqrt{n})$ and negligible space. If n is not prime and the factorization of n is known, then the Pohlig-Hellman algorithm can be used. If $n = p_1^{e_1} p_2^{e_2} \ldots p_c^{e_c}$ is the prime factorization of n, then the Pohlig-Hellman algorithm computes partial solutions by computing discrete logs in subgroups of order p_i for $i = 1 \ldots c$. Typically Pollard's rho algorithm is used as a subroutine to compute these logarithms, and the partial solutions are combined to compute the requested discrete log. The runtime of Pohlig-Hellman is $O(\sum_{i=1}^{c} e_i(\log n + \sqrt{p_i}))$, assuming n has the prime factorization given above. In particular, if n is $B - smooth$, meaning that none of it's prime factors are greater than B, the runtime of the Pohlig-Hellman algorithm is $O(\ln\ln n\,(\log n + pi)$, since the average number of not necessarily distinct prime factors is $\sim ln\ ln\ n$. If n is at most 256 bits and has no factors of more than 16 bits, i.e. n is $(2^{16} - 1)$ -smooth, then we can expect the Pohlig-Hellman algorithm to require only $O(2^{12})$operations. When Pollard's rho algorithm is used with the Pohlig-Hellman algorithm, the combined algorithm also uses negligible space. If n has a large prime factor neither of these algorithms work well.

Index calculus algorithms do not work in a general cyclic group, but they do work in $\mathbb{Z}_p^*$ and they run in sub-exponential time. For example the number field sieve has an expected running time of $(e^{(1.923+O(1))+(\ln p)^{1/3}(\ln\ln p)^{2/3}})$. Index calculus methods do not work directly on subgroups of $\mathbb{Z}_p^*$; however it can be used to compute logs in subgroups by computing logs in $\mathbb{Z}_p^*$ .For this reason, if $n \ll p$ then a square-

root algorithm such as Pollard rho (or Pohlig-Hellman if n is composite) may be faster than index calculus methods, depending on the exact relationship between $n\ and\ p$.

2.3.2.2 Encryption and Decryption

An Elgamal cryptosystem can be described by a 4-tuple $(p, \alpha, \mathrm{a}, y)$, where p is a large prime and describes which group $\mathbb{Z}_p^*$ is used, α is an element of order n in $\mathbb{Z}_p^*$, a is a random integer with $1 \leq \mathrm{a} \leq n-1$, and $y = \alpha^{\mathrm{a}}$. If α is primitive, then $n = p - 1$. However this is not required, and n may be chosen to be much smaller than $p-1$ for efficiency reasons. The public key is (p, α, y), and the private key is a.

Before a message can be encrypted, it must be converted to an integer between 1 and $p-1$, i.e. an element of $\mathbb{Z}_p^*$. If the message is the key for a private cipher, it may already be a number. If the message is larger than $p-1$, it can be broken into blocks. Many textbooks impose a further restriction that the message is a member of $\langle\alpha\rangle$, however in practice implementations often ignore this restriction. If α is not primitive, then only n of the $p-1$ members of $\mathbb{Z}_p^*$ will be in $\langle\alpha\rangle$. In this case, it is not clear how to convert messages to an element of $\langle\alpha\rangle$. Raising g to the power of the message does not work, since a discrete log would be required to retrieve the original message.

The encryption function requires a random integer $k \in [1\,,\ n-1]$ in addition to the public key. The encryption and decryption functions are:

$$E_k(m) = (\alpha^k\ , my^k)\ and\ D(u,v) = u^{-\mathrm{a}}v. \qquad (2.1)$$

where all operations are done $mod\ p$ (in $\mathbb{Z}_p^*$). The decryption function will recover the original message:

$$u^{-\mathrm{a}} = \alpha^{-k\mathrm{a}} = (\alpha^{\mathrm{a}})^{-k} = y^{-k}\ , so\ D\big(E_k(m)\big) = u^{-\mathrm{a}}v = y^{-k}\ my^k = m$$

Algorithm Key generation for Elgamal public key cryptosystem:

Purpose: to produce a public key and a corresponding private key

Inputs: a large random prime number p.

Outputs: a public key and a private key.

Procedure:

Step 1: generate α of the multiplicative group $\mathbb{Z}_p^*$ of the integers modulo p (refer appendix A).

Step 2: Select a random integer a, $1 \leq a \leq p-2$, and compute $\alpha^a \, mod \, p$ (refer appendix A).

Step 3: A public key is (p, α, α^a) and a private key is a

Algorithm: Elgamal public key encryption:

Purpose: B encrypts a message m for A.

Inputs: The message m and public key (p; α; α^a)..

Outputs: The encrypted message.

Procedure:

Step 1: read the message as an integer m in the range $\{0, 1, \ldots, p-1\}$.

Step 2: Select a random integer k, $1 \leq k \leq p-2$.

Step 3: Compute $\gamma = \alpha^k \bmod p$ and $\delta = m \, . \, (\alpha^a)^k \bmod p$

Step 4: Send the ciphertext $c = (\gamma, \delta)$ to A.

Algorithm: Elgamal public key decryption:

Purpose: A decrypts a message c for A.

Inputs: The encrypted message c, private key (a) and the prime number p.

Outputs: The original message m.

Procedure:

Step 1: Use the private key a to compute $\gamma^{p-1-a} \bmod p$ (note: $\gamma^{p-1-a} = \gamma^{-a} = \alpha^{-ak}$).

Step 2: Recover m by computing $\gamma^{-a} \, . \, \delta \bmod p$.

Proof that decryption works.

$\gamma^{-a} \, . \, \delta \equiv \alpha^{-ak} \, m \, \alpha^{ak} \equiv m \pmod p$.

Example with artificially small parameters

Key generation: Entity A selects the prime p = 2357 and a generator α = 2 of $\mathbb{Z}_{2357}^*$

A chooses the private key a = 1751 and computes α^a mod p = 21751 mod 2357 = 1185, A's public key is (p = 2357; α = 2; α^a = 1185).

Encryption: To encrypt a message m = 2035, B selects a random integer k = 1520 and computes γ = 21520 mod 2357 = 1430 and δ = 2035. 1185^{1520} mod 2357 = 697, B sends γ = 1430 and δ = 697 to A.

Decryption: To decrypt, A computes γ^{p-1-a} = 1430^{605} mod 2357 = 872; and recovers m by computing m = 872. 697 mod 2357 = 2035

2.3.2.3 Security

If we recover the private key a, we can decrypt all past and future messages. Since the public key includes $y = \alpha^a$ and α, finding the private key from the public key amounts to computing a single discrete logarithm in $\langle\alpha\rangle$. For this reason, $n\ and\ p$ should be very large, n determines the runtime of square-root discrete log algorithms like Pollard's rho algorithm, and p determined the runtime of index calculus discrete log methods.

The key size typically refers to the size of p, 1024 is the recommended minimum. Legacy implementations running on limited hardware may use 768 bits or even less. If $n < p - 1$ is used, it should be large enough that the $O(\sqrt{n})$ discrete log algorithms take at least as long as the index calculus algorithms.

To break a single ciphertext (u, v) = (α^k, my^k), it would suffice to find y^k, since m = vy^k . Since inverses can be computed efficiently, we really just need y^k. We can find k by computing the discrete log of $u = \alpha^k$ base α . Cracking a single ciphertext is equivalent to the Diffie-Hellman problem: given α^k and $y = \alpha^k$, determine α^{kx} = y^k. Since a discrete log can be used to solve the Diffie-Hellman problem, it is not any more difficult than the discrete log problem; however it is not known whether or not it is less difficult.

2.3.2.4 Efficiency

Consider the Elgamal cryptosystem (p, α, a, y),where $n = |\alpha|$ is the order of α. Encryption requires two exponentiations and one multiplication in $\mathbb{Z}_p^*$. Since the exponent k is chosen between 1 and n - 1 using a large n will slow down exponentiation and therefore encryption. The single multiplication will not be significant compared to the two exponentiations.

Decryption requires one exponentiation, one inversion, and one multiplication in $\mathbb{Z}_p^*$. The private key a is the exponent, and it is also chosen between 1 and n - 1. Therefore choosing a smaller n could also increase decryption performance; however there are other approaches, like picking a with very few ones in its binary

representation. Because of the way modular exponentiation is implemented, this will reduce the computation time.

Decreasing p also increases performance, since multiplications in $\mathbb{Z}_p^*$ are faster for smaller p. This gives the implementer with performance constraints a choice: decrease p and use $n = p - 1$, or keep p large and choose $n \leq p - 1$. According to [12] the latter choice is not uncommon in actual implementations, likely because the faster index calculus discrete log methods depend on p and not n.

2.4 Private Key vs. Public Key Cryptography

Private key and public key encryption schemes have various advantages and disadvantages, some of which are common to both. This section highlights a number of these and summarizes features pointed out in previous sections.

(I) Advantages of private key cryptography

1. Private key ciphers can be designed to have high rates of data throughput. Some hardware implementations achieve encrypts rates of hundreds of megabytes per second, while software implementations may attain throughput rates in the megabytes per second range.
2. Keys for private key ciphers are relatively short.
3. Private key ciphers can be employed as primitives to construct various cryptographic mechanisms including pseudorandom number, hash functions and computationally efficient digital signature schemes.
4. Private key ciphers can be composed to produce stronger ciphers. Simple transformations which are easy to analyze, but on their own weak, can be used to construct strong product ciphers.
5. Private key encryption is perceived to have an extensive history, although it must be acknowledged that, despite the invention of rotor machines earlier, much of the knowledge in this area has been acquired subsequent to the invention of the digital computer, and, in particular, the design of the Data Encryption Standard in the early 1970s.

(II) Disadvantages of private key cryptography

1. In a two-party communication, the key must remain secret at both ends.

2. In a large network, there are many key pairs to be managed. Consequently, effective key management requires the use of an unconditionally trusted third party (TTP).

3. In a two-party communication between entities A and B, sound cryptographic practice dictates that the key be changed frequently and perhaps for each communication session.

4. Digital signature mechanisms arising from private key encryption typically require either large keys for the public verification function or the use of a TTP.

(III) Advantages of public key cryptography

1. Only the private key must be kept secret (authenticity of public keys must, however, be guaranteed).

2. The administration of keys on a network requires the presence of only a functionally trusted TTP as opposed to an unconditionally trusted TTP. Depending on the mode of usage, the TTP might only be required in an "off-line" manner, as opposed to in real time.

3. Depending on the mode of usage, a private key/public key pair may remain unchanged for considerable periods of time, e.g., many sessions (even several years).

4. Many public key schemes yield relatively efficient digital signature mechanisms. The key used to describe the public verification function is typically much smaller than for the private key counterpart.

5. In a large network, the number of keys necessary may be considerably smaller than in the private key scenario.

(IV) Disadvantages of public key encryption

1. Throughput rates for the most popular public key encryption methods are several orders of magnitude slower than the best known private key schemes.

2. Key sizes are typically much larger than those required for private key encryption, and the size of public key signatures is larger than that of tags providing data origin authentication from private key techniques.

3. No public key scheme has been proven to be secure (the same can be said for block ciphers). The most effective public key encryption schemes found to date

have their security based on the presumed difficulty of a small set of number-theoretic problems.

4. Public key cryptography does not have as extensive a history as private key encryption, being discovered only in the mid 1970s.

Summary of comparison

Private key and public key encryption have a number of complementary advantages. Current cryptographic systems exploit the strengths of each. An example will serve to illustrate. Public key encryption techniques may be used to establish a key for a private key system being used by communicating entities A and B. In this scenario A and B can take advantage of the long term nature of the public/private keys of the public-key scheme and the performance efficiencies of the private key scheme. Since data encryption is frequently the most time consuming part of the encryption process, the public key scheme for key establishment is a small fraction of the total encryption process between A and B. To date, the computational performance of public key encryption is inferior to that of private key encryption. There is, however, no proof that this must be the case. The important points in practice are:

1. Public key cryptography facilitates efficient signatures (particularly non-repudiation) and key management; and

2. Private key cryptography is efficient for encryption and some data integrity applications [1].

2.5 Hashing Functions

One of the fundamental primitives in cryptography is the cryptographic hash function, often informally called a one-way hash function. A simplified definition is presented follows.

Definition 2.12: A *hash function* is a computationally efficient function mapping binary strings of arbitrary length to binary strings of some fixed length, called *hash-values*.

For a hash function which outputs n-bit hash-values (e.g., n = 128 or 160) and has desirable properties, the probability that a randomly chosen string gets mapped to a particular n-bit hash-value (image) is 2^{-n}. The basic idea is that a hash-value serves as a

compact representative of an input string. To be of cryptographic use, a hash function h is typically chosen such that it is computationally infeasible to find two distinct inputs which hash to a common value (i.e., two *colliding* inputs x and y such that $h(x) = h(y)$), and that given a specific hash-value y, it is computationally infeasible to find an input (pre-image) x such that $h(x) = y$.

The most common cryptographic uses of hash functions are with digital signatures and for data integrity. With digital signatures, a long message is usually hashed (using a publicly available hash function) and only the hash-value is signed. The party receiving the message then hashes the received message, and verifies that the received signature is correct for this hash-value shown in Fig. 2.5. This saves both time and space compared to signing the message directly, which would typically involve splitting the message into appropriate-sized blocks and signing each block individually. Note here that the inability to find two messages with the same hash-value is a security requirement, since otherwise, the signature on one message hash-value would be the same as that on another, allowing a signer to sign one message and at a later point in time claim to have signed another.

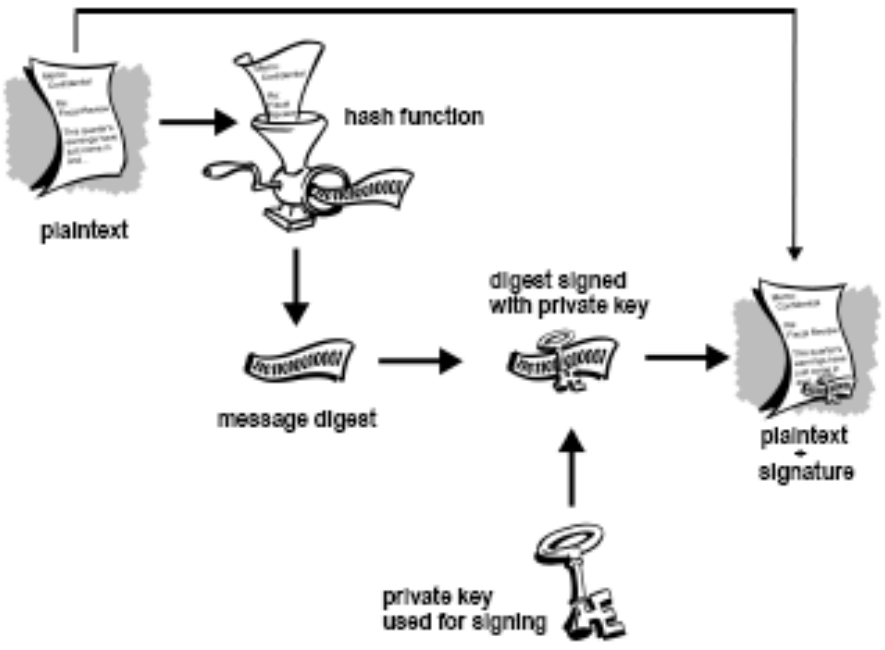


Fig. 2.5: Secure digital signatures.

Hash functions may be used for data integrity as follows. The hash-value corresponding to a particular input is computed at some point in time. The integrity of this hash-value is protected in some manner. At a subsequent point in time, to verify that the input data has not been altered, the hash-value is recomputed using the input at hand, and compared for equality with the original hash-value. Specific applications include virus protection and software distribution. As shown in Fig. 2.6

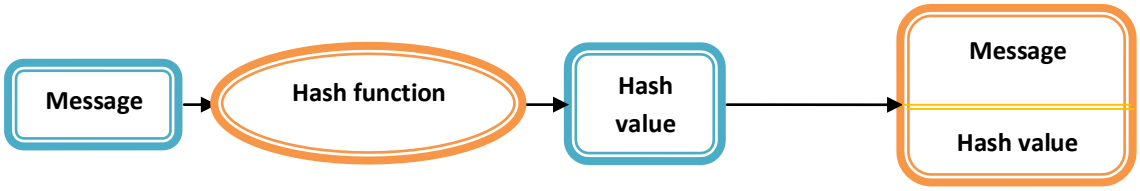


Fig. 2.6: Block diagram of Hashing operation.

A third application of hash functions is their use in protocols involving a priori commitments, including some digital signature schemes and identification protocols [1, 3, and 13].

That's why Hash function takes a random sized input message to produce a fixed sized output which is also known as the message digests. There are many Hash function algorithms such as MD5 [14], security Hash algorithms (SHA) family [15], HAVAL [16], etc.

For example, The Message Digest Algorithm (MD5) is a widely used cryptographic hash function with a 128-bit (16-byte) hash value. MD5 has been employed in a wide variety of security applications, and is also commonly used to check data integrity. However, it has been shown that MD5 is not collision resistant; as such, MD5 is not suitable for applications like Secure Sockets Layer (SSL) certificates or digital signatures that rely on this property. An MD5 hash is typically expressed as a 32-digit hexadecimal number.

MD5 was designed by Ron Rivest in 1991 to replace an earlier hash function, MD4. In 1996, a flaw was found with the design of MD5. While it was not a clearly fatal weakness [14].

On the other hand, the SHA-512 Hash function takes input messages of lengths up to 2128 bits and produces as output a 512-bit message. There are many Hash function techniques presented by the SHA family. Among these are SHA-1, SHA-224,

SHA-256, SHA-384, and SHA-512, which are designed by the National Security Agency (NSA) and published by the NIST as a U.S. government standard [15].

SHA-1 is a one of the security Hash algorithms (SHA), and it is employed in several security applications like SSL/TLS, PGP, SSH, S/MIME, and IPSec. SHA-1 is the successor to MD5. SHA-1 was cracked recently in 2005, by Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu [17].

2.6 Chaotic Cryptosystems

Chaos is the phenomenon of apparently random or unpredictable behavior in deterministic systems, it refers to the irregular output of deterministic system whose behavior is difficult to predict because there are so many variable or unknown factors. These systems' state evolves with time and exhibit dynamics and high sensitivity to initial conditions. As a result of this sensitivity, which manifests itself as an exponential growth of perturbations in the initial conditions, the behavior of chaotic systems appears to be random [18], [19]. This happens even though these systems are deterministic, meaning that their future dynamics are fully defined by their initial conditions, with no random elements involved. This behavior is known as deterministic chaos, or simply chaos [20]. This kind of systems can be a simple non-linear equation or a complex prediction mathematical model.

For a dynamical system to be classified as chaotic, it must have the following properties

1. it must be sensitive to initial conditions;
2. it must be topologically mixing; and
3. Its periodic orbits must be dense.

Sensitivity to initial conditions means that each point in such a system is arbitrarily closely approximated by other points with significantly different future trajectories. Thus, an arbitrarily small perturbation of the current trajectory may lead to significantly different future behavior. However, it has been shown that the last two properties in the list above actually imply sensitivity to initial conditions and if attention is restricted to intervals, the second property implies the other two (an alternative, and in general weaker, definition of chaos uses only the first two properties in the above list). It is interesting that the most practically significant condition, that of sensitivity to

initial conditions, is actually redundant in the definition, being implied by two (or for intervals, one) purely topological conditions, which are therefore of greater interest to mathematicians.

Sensitivity to initial conditions is popularly known as the "butterfly effect", so called because of the title of a paper given by Edward Lorenz in 1972 to the American Association for the Advancement of Science in Washington, D.C. entitled Predictability: Does the Flap of a Butterfly's Wings in Brazil set off a Tornado in Texas? The flapping wing represents a small change in the initial condition of the system, which causes a chain of events leading to large-scale phenomena. Had the butterfly not flapped its wings, the trajectory of the system might have been vastly different.

A chaotic map is a map that exhibits some sort of chaotic behavior. Maps may be parameterized by a discrete-time or a continuous-time parameter. Discrete maps usually take the form of iterated functions. Their properties, such as sensitive dependency on initial conditions and system parameters, and random-like outputs, are similar to confusion and diffusion cryptography properties, so they have been used to build good cryptosystems. Furthermore, these properties make the chaotic cryptosystems robust against any statistical attack, for more details see subsection 2.6.2.

The simplest class of chaotic dynamic systems is one-dimensional chaotic map which is a difference equation of the form

$$x_{n+1} = F(x_n, \delta) \qquad n = 0, 1, 2, 3, \ldots \tag{2.2}$$

Where the state variable x and the system parameter δ are scalars, i.e., x, $\delta \in \mathbb{R}$, and f is mapping function defiened in the real domain $\mathbb{R} \rightarrow \mathbb{R}$.

From Eq. (2.2), it can be seen that one-dimensional chaotic maps refer to those with the relation where the value of x_{n+1} is determined only by x_n. More specifically, this is known as recurrence relation. In chaotic dynamics, iteration is involved, which means to evaluate the map f over and over. [21, 22] For example, logistic chaotic map is one-dimensional chaotic map.

2.6.1 Logistic Chaotic Map

The logistic map is a polynomial mapping (equivalently, recurrence relation) of degree 2, often cited as an archetypal example of how complex, chaotic behavior can arise from very simple non-linear dynamical equations. The map was popularized in a seminal 1976 paper by the biologist Robert May, in part as a discrete-time demographic model analogous to the logistic equation first created by Pierre François Verhulst. Mathematically, the logistic map is written

$$x_{n+1} = r x_n (1 - x_n) \tag{2.3}$$

where:

x_n is a number between zero and one, and represents the ratio of existing population to the maximum possible population at year n, and hence x_0 represents the initial ratio of population to maximize population (at year 0).

r is a positive number, and represents a combined rate for reproduction and starvation.

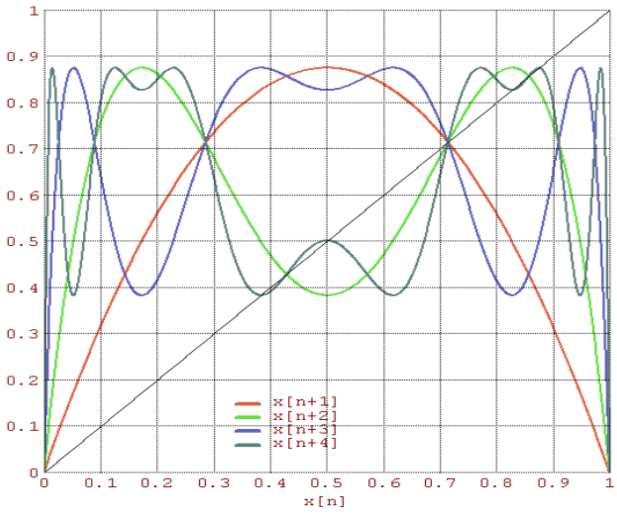


Fig. 2.7: Logistic function for r=3.5 after first 3 iterations

This nonlinear difference equation is intended to capture two effects.

- Reproduction where the population will increase at a rate proportional to the current population when the population size is small.
- Starvation (density-dependent mortality) where the growth rate will decrease at a rate proportional to the value obtained by taking the theoretical "carrying capacity" of the environment less the current population.

However, as a demographic model the logistic map has the pathological problem that some initial conditions and parameter values lead to negative population sizes. This problem does not appear in the older Ricker model, which also exhibits chaotic dynamics.

2.6.2 Properties of Chaotic Systems

The chaotic systems have three main properties:

1. sensitivity to initial conditions and system parameters
2. random-like outputs
3. unstable periodic orbits with long periods

These properties are explained in the following subsections. Using logistic map as an example to illustrates the following properties.

2.6.2.1 Local Instability

A chaotic system is sensitive to the initial conditions and parameters. Fig. 2.8 shows the output plot of two orbits of the logic map with red line (r=3.9995) and blue line (r=3.9994), starting with same initial condition $x_0 = 0.567021$

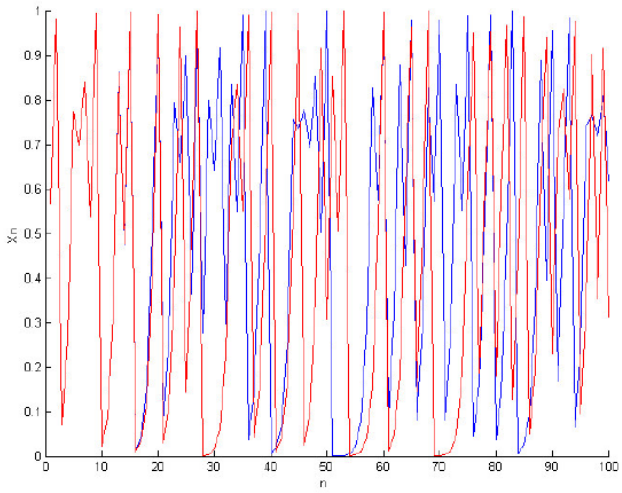


Fig. 2.8: Orbits 100 iterations of logistic map using two different system parameter (r)

Fig. 2.9 shows the output plot of two orbits of the logic map with red line ($x_0 = 0.567021$) and blue line ($x_0 = 0.567020$) , starting with same r= 3.9995.

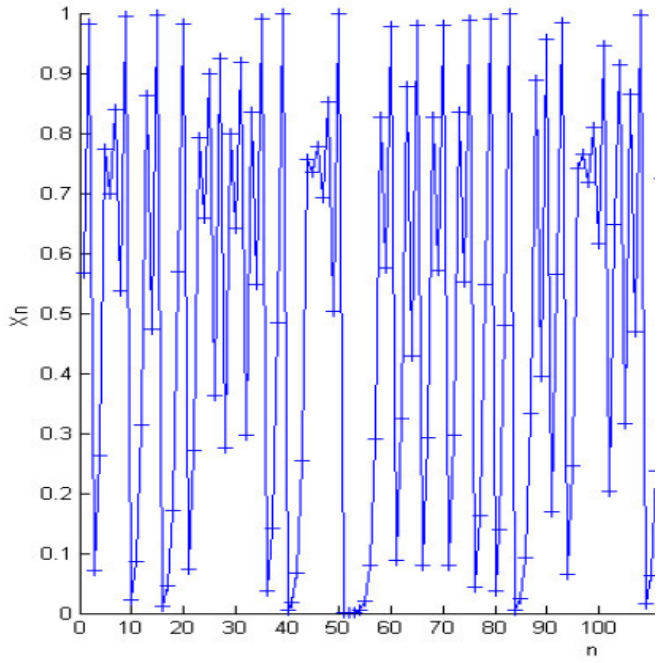


Fig. 2.9: Orbits 100 iterations of logistic map using two different initial condition with constant system parameter (r)

2.6.2.2 Stochasticity

The output of normal deterministic dynamic system is easily predictable. However, for a chaotic dynamic system, the output behavior is random-like and hard to predict in long term. Fig. 2.10 shows the random-like output of a logistic map $x_0 = 0.565 \; and \; r = 3.9995$.

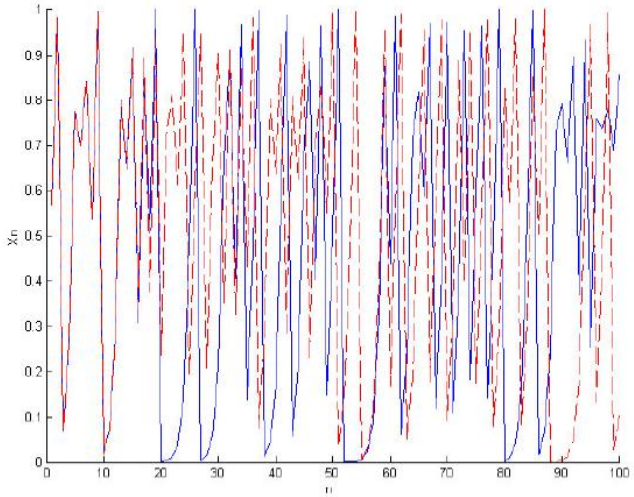


Fig. 2.10: Orbits 200 iterations of logistic map using the initial condition = 0.565 with constant system parameter (r= 3.9995)

2.6.2.3 Ergodicity

For a normal deterministic dynamic system, its output is usually periodicity. However, for a chaotic dynamic system, the system output is ergodic (non - periodic). The orbits of the logistic map at different number of iterations are plotted in Fig, 2.11 while the distribution of the output in 10000 iterations is shown in Fig. 2.12.

2.6.3 Chaotic System for Cryptography

There are some similar properties between chaotic systems and traditional cryptographic algorithms. In cryptographic algorithms, diffusion and confusion are applied on plaintext over the encryption rounds of the algorithm. In chaotic systems, similar things happen on the initial input parameters. After a sufficiently large number of iterations, an input parameter will be eventually spread over the entire phase space through the random-like orbit over iterations. The stochasticity property of a chaotic system is similar to the diffusion and confusion properties of cryptographic algorithms.

For cryptographic algorithms, in order to decrypt the ciphertext to the original plaintext correctly, the same key should be used in both encryption and decryption. This is just similar to the requirement that chaotic systems need the same input parameter to reproduce the same output orbit. In this case, the system parameters and initial conditions can be considered as the private key of a chaotic cryptosystem. The table 2.1 summaries the common relationship which promotes chaos theory into practical cryptographic design [23, 24].

Table 2.1: A comparison of some features characterized by chaotic system and traditional cryptosystems.

Chaotic system	Traditional cryptosystems
Ergodicity	Confusion
Sensitivity to initial condition and system parameters	Diffusion
Parameters	Encryption key
Iterations	Cipher rounds

In other words, confusion in traditional cryptosystems causes plaintext transforming to random ciphertext such that there should be no repeated pattern in the ciphertext. By the same token, the trajectories of chaotic systems pass through all points

of the phase space generally with uniform distribution, which means, it is very difficult to predict the final position of one point from its initial position. It is indeed the concept of ergodicity which can be associated with confusion in cryptosystems.

To develop a good cryptosystem, anther essential design principle is the property of diffusion. By doing so, a totally different ciphertext is resulted no matter how one bit of key or plaintext is changed. This implies that the system is sensitive to plaintext and its encryption key. On the other hand, recall that the chaotic systems highly depend on initial conditions and parameters. A small variation in any of the system parameters or initial points leads to the trajectories diverged significantly. In this regard, chaotic systems and cryptosystems can naturally benefit from each other.

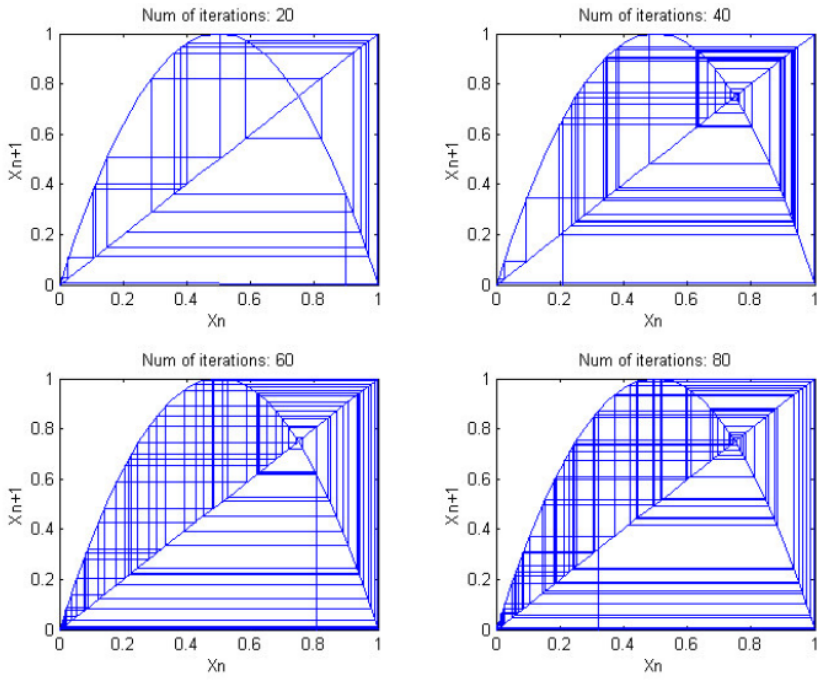


Fig. 2.11: The orbits of the logistic map at different number of iterations

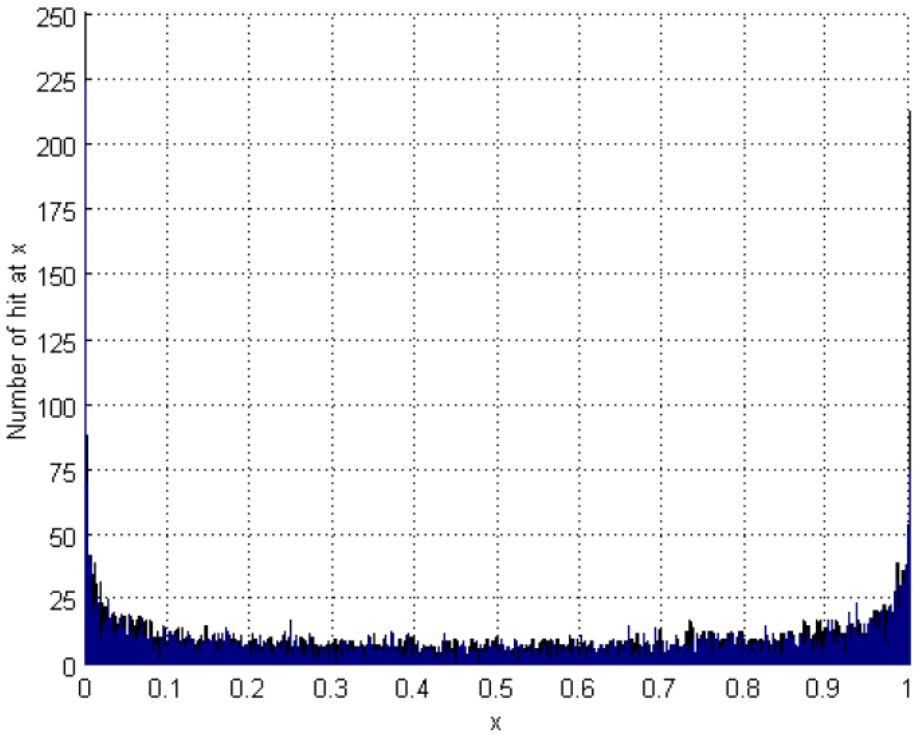


Fig. 2.12: The distribution of the output in 10000 iterations

2.7 Quasi group cryptosystems

Quasi groups (or Latin squares) provide a powerful technique for generating a larger set of permutation transformations by permuting not only the samples but also transforming the amplitudes themselves across their range [25]. By doing this, they provide an immensely large number of keys, even for small alphabets. Therefore, quasi group based ciphers can have a variety of applications, and in some cases can be competitive to number theory based systems in terms of the difficulty they offer to brute force attacks.

2.7.1 Quasi Group Definition

A quasi group Q may be defined [25] as a group of elements along with a multiplication operator such that for its elements x and y there exists a unique solution z, also belonging to Q, such that the following two conditions are obeyed:

$$(1)\ x * a = z \tag{2.4}$$

$$(2)\ x * b = z \tag{2.5}$$

A quasi group may be defined alternatively as a binary system $(Q,*)$ satisfying the two conditions:

(i) *For any a, b belonging to Q there exists a unique x belonging to Q such that*

$$a * x = b$$

(ii) *For any* a, b *belonging to* Q *there exists a unique* y *belonging to* Q *such that*

$$y * a = b$$

Definition 2.13: A *quasi group* is a set Q together with a binary operation * such that for all $a, b \in Q$ the equations $y * a = b$ *and* $a * x = b$ have unique solutions$y$ and x in Q respectively. A quasi group is said to be of order n if there are n elements in the set Q [26].

Table 2.2: Multiplication table for a quasi group of order 5

*	1	2	3	4	5
1	4	3	2	1	5
2	2	1	5	4	3
3	5	4	3	2	1
4	1	5	4	3	2
5	3	2	1	5	4

Example 1: An example of a quasi group Q of order 5 with elements $(1, 2, 3, 4, 5)$ is given by the element multiplications of Table 2.2. A multiplication operator in a quasi group behaves as a mapping between the row and the column indices. Suppose $x = 2$ *and* $a = 3$; the resulting z can be determined by looking up the element having the row index as 2 and the column index as 3 so, the obtained value of z is 5 $(z = 2 * 3)$[25] .

The multiplication table of a finite quasi group is a Latin square; that is, each element of Q occurs exactly once in each row and each column of the multiplication table defining '$*$'. Conditions (1) and (2) essentially postulate the existence of unique left and right divisors for each element in Q. This gives rise to an explicit definition of left and right division operations:

Let $(Q,*)$ be a quasi group, then two operations \ and / on Q can be defined as:

$$(3)\ x * (x\backslash y) = y \text{ and } x(x * y) = y \qquad (2.6)$$

$$(4)\ (y/x) * x = y \text{ and } (y * x)/x = y \qquad (2.7)$$

The following tables are an example of a quasi group Q of order 4 given in terms of multiplication (a), right (b) and left divisors (c) [27].

Table 2.3: Multiplication table for a quasi group of order 4 and operations / and \ tables

*	1	2	3	4
1	1	3	2	4
2	2	4	3	1
3	3	1	4	2
4	4	2	1	3

/	1	2	3	4
1	3	1	2	4
2	2	4	3	1
3	4	3	1	2
4	1	2	4	3

\	1	2	3	4
1	1	3	2	4
2	4	1	3	2
3	2	4	1	3
4	3	2	4	1

2.7.2 Quasi Group Encryption / Decryption

We can now define a quasi group cipher in terms of encryption and decryption function [28].

2.7.2.1 Quasi Group Encryption

Let $(Q,*,\backslash,/)$ be a quasi group and $a_1, a_2, a_3, \ldots, a_n \in Q$. We define the encryption function E with respect to the key $a \in Q$ as

$$E_a(a_1, a_2, a_3, \ldots, a_n) = b_1, b_2, b_3, \ldots, b_n \quad (2.8)$$

where $b_1, b_2, b_3, \ldots, b_n \in Q$ are computed by

(i) $b_1 = a * a_1$

(ii) $b_i = b_i - 1 * a_i, for\ i = 2, \ldots, n.$

Where i increments from 2 to the number of elements that have to be encrypted.

The next example illustrates the working of Eq. (2.8) with the help of the tables 2.3.a, using the seed a=2. This equation maps the initial input data vector $(a_1, a_2, a_3, a_4, a_5, a_6)$ = (2, 4, 1, 2, 3, 3) into the vector $(b_1, b_2, b_3, b_4, b_5, b_6)$ using the quasi group of order 4.

The following steps are used during the process of encryption:

$b_1 = a * a_1 = 2 * 2 = 4$

$b_2 = b_1 * a_2 = 4 * 4 = 3$

$b_3 = b_2 * a_3 = 3 * 1 = 3$

$b_4 = b_3 * a_4 = 3 * 2 = 1$

$b_5 = b_4 * a_5 = 1 * 3 = 2$

$b_6 = b_5 * a_6 = 2 * 3 = 3$

The resulted vector is (4, 3, 3, 1, 2, 3), it's noticed that the input vector has duplicated values such as (2, 3), whereas the resulted vector changes their values, i.e. the first 2 in the input vector become 1, while the second one become 3.

2.7.2.2 Quasi Group Decryption

The decryption process is similar to the encryption but the left division operation '\' is used as operation. The decryption function D is then defined as:

$$D_a(a_1, a_2, a_3, \ldots, a_n) = e_1, e_2, e_3, \ldots, e_n \quad (2.9)$$

where the original plaintext is computed by

(i) $e_1 = a \backslash a_1$

(ii) $ei = a_i - 1 \backslash a_i, for\ i = 2, \ldots, n.$

To perform the process of decryption we need to first generate the inverse matrix of a given quasi group (left division operation) and execute mapping procedure as described in the following algorithm.

Algorithm the Left Inverse of a Quasi Group

By executing the following algorithm, we can generate the left inverse of a given quasi group:

The Left Inverse of a Quasi Group:

Purpose: Getting the left inverse of a quasi group.

Inputs: a quasi group.

Outputs: left inverse of a quasi group.

Procedure:

Step 1: Scan the row of the quasi group

Step 2: Locate an element i (start the value of i from 1) and note the value of v in the corresponding location of the inverse matrix.

Step 3: Increment i.

Step 4: Go to step 1

Step 5: This process continues till all the elements in the row are exhausted.

Step 6: Then go to the next row and repeat steps 1 through 3

To recover the original vector from the encrypted vector (4, 3, 3, 1, 2, 3), we use the left operation table 2.3.c with the seed a=2. The following steps are used during the process of encryption:

$e_1 = a * a_1 = 2 * 4 = 2$

$e_2 = b_1 * a_2 = 4 * 3 = 4$

$e_3 = b_2 * a_3 = 3 * 3 = 1$

$e_4 = b_3 * a_4 = 3 * 1 = 2$

$e_5 = b_4 * a_5 = 1 * 2 = 3$

$e_6 = b_5 * a_6 = 2 * 3 = 3$

2.7.3 Properties of Quasi Group Cryptography.

(i) It is resist to the brute force attack

- The Hall algorithm: there is at least $n!\ (n-1)!\ \ldots 2!$ Latin squares. Let A= {0,…, 255} (i.e. data are represented by 8 bits), there are at least 256! 255! …2!>10^{58000} quasi groups.
- Numbers of reduced Latin square

  Latin square is said to be normalized or reduced if the elements of both its first row and its first column are in a natural order. The number of Latin squares LS(n) of order n increases very quickly with n and is indeed great, even for rather small n. It should be noted that the number of reduced Latin squares is exactly known for n < 10 , and $0.753 \cdot 10^{102805} < LS(256) < 0.304 \cdot 10^{101724}$.

- Since the number of Latin squares of order q (usually 32< q< 256), equal to the number of elements of an alphabet used in practice to represent the plaintext and the cryptogram, is immense, the key space is practically unlimited. Therefore, quasi group based ciphers are much more efficient and more secure than those based on regular algebraic systems such as fields or groups, or on deterministic algorithms (elliptic curve cryptosystems, conventional public-key cryptosystems) [29].

Table 2.4: Number of reduced Latin squares T(n) vs. n.

n	L_n
1	1
2	1
3	1
4	4
5	56
6	9,408
7	16,942,080
8	535,281,401,856
9	377,597,570,964,258,816
10	7,580,721,483,160,132,811,489,280

- Suppose that intruder knows a ciphertext $v=v_1v_2...v_k$, he has to recover the quasi group *(A,*)*. But there is no algorithm of the exhaustive search of all quasi groups that can be generated [30].

(ii) It is resist to the statistical attack.

- Let *(Q, *)* be a quasi group of *q* elements. Among the set of all possible cipher of certain length, all possible element of *Q* occurs with equal probability, i.e., each element of quasi group *Q* should occur as often as any other in each position.

- It is proved that each element occurs exactly *q* times among the products of two elements of *Q*, q^2 times among the products of three elements of *Q* and, generally q^{t-1} among the products of t elements of Q , its useful property when use a multilayer quasi group cryptosystem.

- The number (x) in quasi group of order 2 occurs twice, and in quasi group of order 3, it appears 3 times and so on.

2.8 Summary

This chapter covers the fundamentals and terminologies of cryptography, including the issues of private key cryptosystems and public key cryptosystems. Also the chaotic cryptography is explained, and its cryptographic properties are confirmed. In

addition to illustrate each of quasi groups' definition, quasi group cryptosystems and quasi group cryptographic properties.

We mention the advantages of private key cryptosystems and public key cryptosystems, which motivate us to use both of them. Furthermore, we highlight the strength of chaotic cryptography and quasi group cryptosystems in order to involve them in our proposal cryptosystem.

Chapter 3:

Literature Review

In this chapter numerous cryptography schemes are studied and their performance is evaluated, to avoid their weaknesses in our proposed cryptosystem.

3.1 Previous Work

Public key encryption protocols work using two keys. In these algorithms, there is a pair of keys, one of which is known to the public and used to encrypt the plaintext to be sent to the receiver who owns the corresponding decryption key, known as the private key.

Every public key cryptosystem is based on a mathematical problem that is in some sense difficult to solve. Basically, the three major types of mathematical hard problem that had been successfully being used in cryptography are the integer factorization problem, group factorization problem, the discrete logarithm problem and the Elliptic Curve discrete logarithm problem.

As example of public key cryptosystem, Elliptic Curve Cryptography (ECC) was invented by Neil Koblitz in 1987 and by Victor Miller in 1986. ECC has been proven to involve much less overheads when compared to RSA. The ECC has been shown to have many advantages due to ability to provide the same level of security as RSA yet using shorter keys [31]-[36].

However, its disadvantage is its luck of maturity, as mathematicians believe that not enough research has done in elliptic curve discrete logarithm problem (ECDLP) [37].

In [38] the author illustrates a public key cryptosystem which based on quantum systems. In quantum cryptography two parties can secure network communications by applying the phenomena of quantum physics. The security of these transmissions is based on the inviolability of the laws of quantum mechanics, which create the ability of the two communicating users to detect the presence of any third party trying to gain knowledge of the key. This result from a fundamental aspect of quantum mechanics: the process of measuring a quantum system in general disturbs the system. A third party

trying to eavesdrop on the key must in some way measure it, thus introducing detectable anomalies. So, we can use it only to produce and distribute a key, not to transmit any message data. However, it has a problem; the technology needed to build a quantum computer is currently beyond our reach. This is due to the fact that the coherent state, fundamental to a quantum computers operation, is destroyed as soon as it is measurably affected by its environment. Attempts at combating this problem have had little success, but the hunt for a practical solution continues [39].

While in Koblitz cryptosystems authors have been developed a cryptosystem based on Koblitz curves which is a family of elliptic curves offering faster computation than general elliptic curves. However, in order to enable these faster computations, scalars need to be reduced and represented using a special base-tau expansion. So, efficient conversion algorithms are necessary. Existing conversion algorithms require several complicated operations, such as multiprecision multiplications and computations with large rationales, resulting in slow and large implementations in hardware and microcontrollers with limited instruction sets. They build two algorithms, the first is designed to utilize only simple operations, such as additions and shifts, which are easily to implement on platforms, and to implement the second they used Altera Stratix II FPGAs which improved both computation speed and required area compared to the existing solutions [40, 41].

The basics of chaotic cryptography are discussed in the previous chapter; many researches in this field have been developed. In 2003, a cryptosystem with a private key block cipher algorithm used an external key of variable length (maximum 128-bits) to generate the system parameters and the initial conditions of the chaotic map. The ciphertext depends on the private key only [42].

The main weaknesses of this technique are the ciphertext depends on the private key only, so it is vulnerable to known plaintext attack. It uses small block size (8 bits), and easy to brute force because x_0 the initial condition only 256 values.

Later, [43] explained the weaknesses of using a 128-bit external key to derive the initial conditions and number of iterations. These weaknesses are summarized as follows: The interval chosen for a system parameter (λ), the small resolution used to calculate it, together with the deterministic nature of the algorithm, allow for a known plaintext attack. Also, the process to derive an initial condition (X) and number of

iterations (N) from the external key (K) is fundamentally flawed, allowing for chosen ciphertext and chosen plaintext attacks. It is concluded from these facts that the total lack of security along with the low encryption speed discourages the use of this algorithm for secure applications.

In 2005, another chaotic map based technique was proposed, in which multiple one-dimensional chaotic maps are used instead of a one-dimensional chaotic map. This algorithm uses an external secret key of variable length (maximum 128-bits). The plaintext is divided into groups of variable length (number of blocks in each group is different). These groups are encrypted by using randomly chosen chaotic map from a set of chaotic maps. The number of iterations and initial conditions for the chaotic maps depend on the randomly chosen session key and on the previous block of ciphertext. The encryption/decryption process is governed by two dynamic tables, which contain the number of iterations and initial conditions for the chaotic maps, these tables are updated from time to time during the encryption/decryption process [44].

In this cryptosystem, the ciphertext depends on the private key only, so it is vulnerable to known plaintext attack and also a small block size (8 bits) is used.

Although the previous algorithm is good regarding the confusion and diffusion as well as efficiency, in 2007, a cryptanalysis technique showed some weaknesses of it, which can be summarized as follows: The cipher generated looks like a block cipher, but it behaves as a stream cipher ,and equations of initial conditions and number of iterations are dependent on the secret key only, which results in the initial contents of the two dynamic tables exactly the same for different plaintext sequences as long as the secret key is fixed. The variable, the initial condition, which changes by iterating the maps, is used to update the two tables for encrypting the next plaintext block, and each plaintext block is encrypted with the last value of X . At the end of the paper, they make a straightforward modification to make the value of X dependent on both the key and the plaintext [45].

A private key cryptosystem is publicized in [46], an improved cryptosystem has been proposed to show the essential weaknesses and redundancies of the previous chaotic cryptosystems. An improved scheme is used to eliminate these weaknesses by different approaches. This scheme uses two skew tent maps instead of a logistic map. The redundant operations in previous systems are abandoned to simplify the cipher.

Permutation within ciphertext was implemented by using two independent chaotic variables to mask the plaintext.

Here, in the improved chaotic scheme, a one-dimensional chaotic map is used, which limits the degree of confusion and diffusion. Also, the number of iterations is calculated using the key indices (60 to 67), i.e. K_{60},K_{61},….,K_{67} stream is used to provide the number of iterations to be applied to the map, which reduces the range of the expected number of iterations (Maximum number of iterations = 2^8 (256) values).

On the other hand, quasi group is a non-associative group, has good scrambling properties. The approaches [47, 48, 49] using a quasi group multiplication operation as a permutations. However, these schemes need another reliable encryption algorithm is required to preserve the secrecy of the encryption. It is necessary to transmit the quasi group that is being used for encryption, which is one of the main drawbacks of the above approach. Once the eavesdropper breaks the encapsulating cipher he has access to the quasi group used for the encryption and all the other required information to get the data.

Images are widely used in various applications, that include military, legal and medical systems and these applications need to control access to images and provide the means to verify integrity of images. So, there are many image cryptosystems, for example, in [50] the algorithm, Shuffle Encryption Algorithm (SEA), uses nonlinear s-box byte substitution. Then, it performed a shuffling operation partially dependent on the input data and the given key. Statistical analysis showed that SEA is not vulnerable to statistical attacks. In addition, the huge number of possible keys makes SEA is not vulnerable to brute force attack, also. However, other types of attacks, not related to statistical or brute force attacks, are possible.

As another example of image encryption publicized in [51] the visual cryptography is applied for color images, for each pixel in the secret image, we first represent its value by binary bits with several bit-levels. Then, we encrypt the bits at each bit-level by applying the corresponding black and white visual cryptography schema (VCS) and extended visual cryptography scheme (EVCS) that has the same access structure and under the same visual cryptography model. This cryptosystem has problem which is how to determine the grey levels of the bit-levels, which is quite

complicated and depends on the different color model, the content of the secret image, the access structure, and the observers' experiences and so on.

Recently, an image encryption scheme based on a compound chaotic sequence was proposed in [52]. This scheme includes two procedures: substitutions of pixel values with XOR operations, and circular shift position permutations of rows and columns. The XOR substitutions are controlled by a compound pseudo-random number sequence generated from two correlated chaotic maps. And the row and column circular shift permutations are determined by the two chaotic maps, respectively. This schema suffers from the following, as mentioned in [53]: it can be broken by using only three chosen plain images. In addition to exist some weak keys and equivalent keys. Additionally, it is not sufficiently sensitive to the changes of plain images and the compound chaotic sequence is not random enough to be used for encryption.

Huang-PeiXiao , Guo-ji Zang[54] proposed scheme using two chaotic systems based on the thought of higher secrecy of multi-system. One of the chaotic systems is used to generate a chaotic sequence. Then this chaotic sequence was transformed into a binary stream by a threshold function. The other chaotic system was used to construct a permutation matrix. Firstly, the pixel values of a plain image have been modified randomly using the binary stream as a key stream. Secondly, the modified image was encrypted again by permutation matrix. This schema has two weaknesses, namely, low sensitivity to the change of plain image and Low encryption speed.

3.2 Research Issues

We notice that all the algorithms are either classified as a private key, they suffer from the disadvantages of private key cryptosystem, or classified as a public key and as a result of this, and they suffer from the disadvantages of public key cryptosystem in addition to its private shortcoming.

It's also observed that most of the previous schemes have one or more of the following weaknesses:

- Need to key distribution method.
- Unsatisfactory security.
- Insufficiently sensitive to the changes of plaintext or plain images.
- Low encryption speed.

- Limitation of the degree of confusion and diffusion.
- The encryption process is dependent on the secret key only.
- Initial conditions and number of iterations are dependent on the secret key
- Vulnerable to known plaintext attack
- Vulnerable to chosen plaintext attack
- Vulnerable to chosen ciphertext attack
- Vulnerable to brute force attack
- Vulnerable to statistical attacks
- For image cryptosystem: there is no work done previously
    - In the color components level.
    - With the luminance component level.

So, we need a system that has the advantage of both types, in another words, the system must be secure and fast, for this reason, we need a hybrid system that consist of public key cryptosystem and private key cryptosystem, moreover the proposed cryptosystem must overcome most of the previous shortcomings.

Chapter 4:

Improved Elgamal Cryptosystem

Elgamal cryptosystem is explained in chapter 2, as an example of a public key cryptosystem. Its security is based on the intractability of the discrete logarithm problem in a large prime modulus and the Diffie-Hellman problem. It is probabilistic, meaning that a single plaintext can be encrypted to many possible cipher texts [11].

The following diagrams illustrate the Elgamal cryptosystem key generation, encryption and decryption processes.

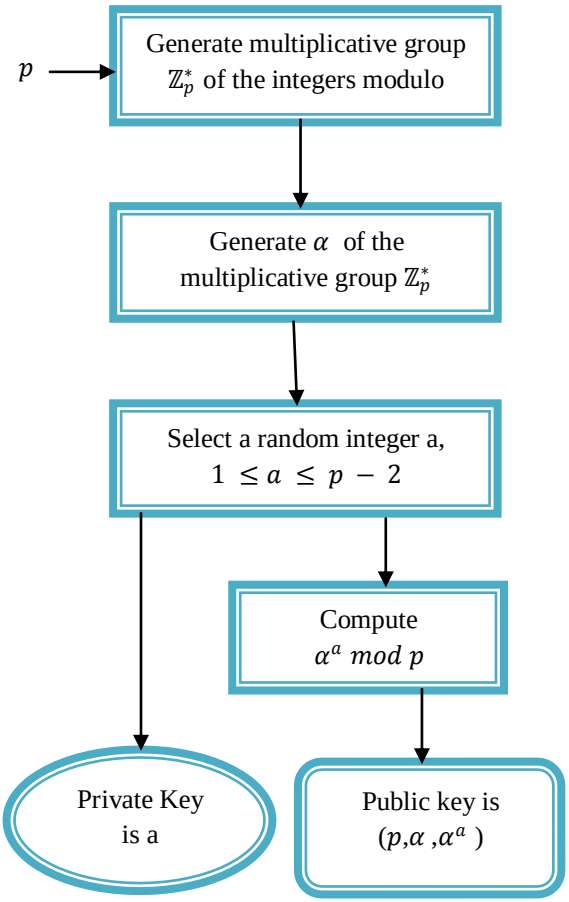


Fig. 4.1: Elgamal key generation diagram.

In section 2.3.2.2 Elgamal cryptosystem steps are demonstrated, even as the security of Elgamal is explored in 2.3.2.3, whereas the efficiency of Elgamal is explored in 2.3.2.4, it's noticed that Elgamal has the following weaknesses:

- Elgamal is a malleable cryptosystem
- Elgamal is not secure under chosen ciphertext attack
- Elgamal system which is the need for randomness

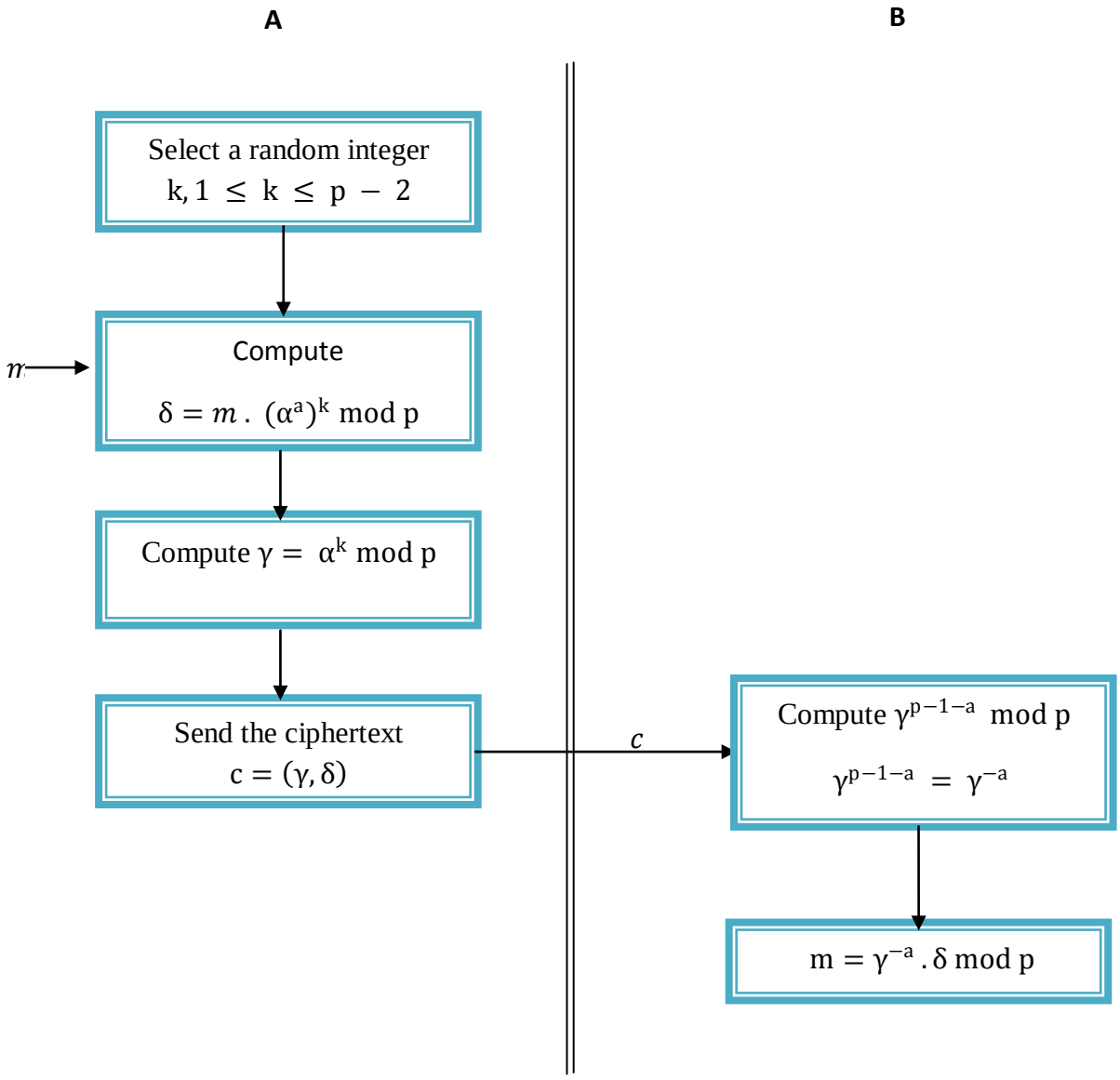


Fig. 4.2: Elgamal cryptosystem diagram.

The suggested improvements in the next section, used to eliminate Elgamal shortcomings

Elgamal-like cryptosystem requires a generation of two random numbers r1 and r2, then break the message into t pieces of length 512 bits, compute b1=g^{r1}mod p and b2=g^{r2}mod p , for each ease piece Cj= mj . $\left(pi^{r1}\ xor\ (pi^{r2})^{2j}\right) mod\ p$, send {b1, b2 ,Cj}, to decrypt it, mj=Cj . $(b1^{xi}\ xor\ b2^{xi\,2j})^{-1} mod\ p$ where pi=g^{xi} [78].

This system has a fundamental weakness; successful decryption can not be guaranteed.

4.1 The Proposed Improvements of Elgamal

The cryptosystem is improved and its disadvantages are eliminated using hash functions and chaotic maps.

4.1.1 Malleability

The first disadvantage is malleability, Elgamal is a malleable cryptosystem, and i.e. it is possible for an adversary to transform a ciphertext into another ciphertext which decrypts to a related plaintext.

For example, the message m is encrypted as E(m) = ($m\gamma^k$, α^k), where (a, γ) is the public key. Given such a ciphertext (c_1,c_2), an adversary can compute (c_1,t.c_2) , which is a valid encryption of tm, for any t.

E(tm) = (t. $m\gamma^k$, t. α^k)

The four parties P_1, P_2, P_3, P_4 execute a two party protocol in two concurrent sessions (see Fig. 4.3). The left session is between P_1 and P_2, and the right session is between $P_3 and\ P_4$. Both $P_2\ and\ P_3$ are controlled by an adversary, whereas P_1 and P_4 are honest and follow the protocol (and thus, P_1 is oblivious to the (P_3, P_4) interaction and $P4$ is oblivious to the (P_1, P_2) interaction). This means that P_2 and P_3 combined correspond to the adversary C in our notation, and that P_1 corresponds to sender (A) in our notation, where P_4 corresponds to receiver (B) in our notation [55].

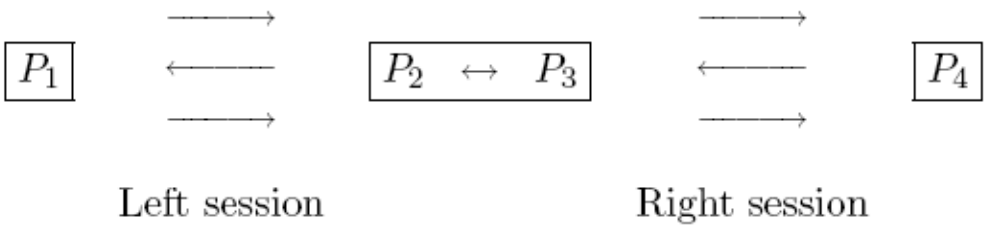


Fig. 4.3: Non-malleability: P2 and P3 are controlled by the adversary

To encroach this trouble, I used one-way functions, hash function to prevent adversary from change the ciphertext of original message with anther valid ciphertext.

Hash function is involved in Elgamal cryptosystem to make it not malleable. In encryption process we compute the hash value of ($E(m) = (m\gamma^k, \alpha^k)$) them the new $E(m) = (m\gamma^k, \alpha^k, \text{hash}\ (\ (m\gamma^k, \alpha^k))\ )$.

where hash (x) is one of the following MD2, MD5, SHA-1, SHA-256, SHA-384 and SHA-512.

With each new session, we use different hash function; it depends on the value of the key. Hence, the improved Elgamal is a nonmalleable cryptosystem. For more details, see appendix C.

4.1.2 Chosen Ciphertext Attack

Chosen ciphertext attack is an attack in which the cryptanalyst gathers information, at least in part, by choosing a ciphertext and obtaining its decryption under an unknown key. In the attack, an adversary has a chance to enter one or more known ciphertexts into the system and obtain the resulting plaintexts. From these pieces of information the adversary can attempt to recover the hidden secret key used for decryption [56].

The cryptosystem is not secure under chosen ciphertext attack, in this attack, an adversary has a chance to enter one or more known ciphertexts into the system and obtain the resulting plaintexts, by using the hash functions we also got rid from this weakness.

The adversary cannot send his/her cipher, not only because he don't know which hash function will be used, but because the hash function property, the inability to find two messages with the same hash value.

4.1.3 Randomness

By using chaotic map to generate the variable (k) in encryption process, we increased the confusion and diffusion of the cryptosystem, due to the chaotic system properties, this solve the main disadvantage of the Elgamal system which is the need for randomness mechanism.

The improved Elgamal cryptosystem is shown in Fig. 4.4, where the orange color is used to highlight the improvements.

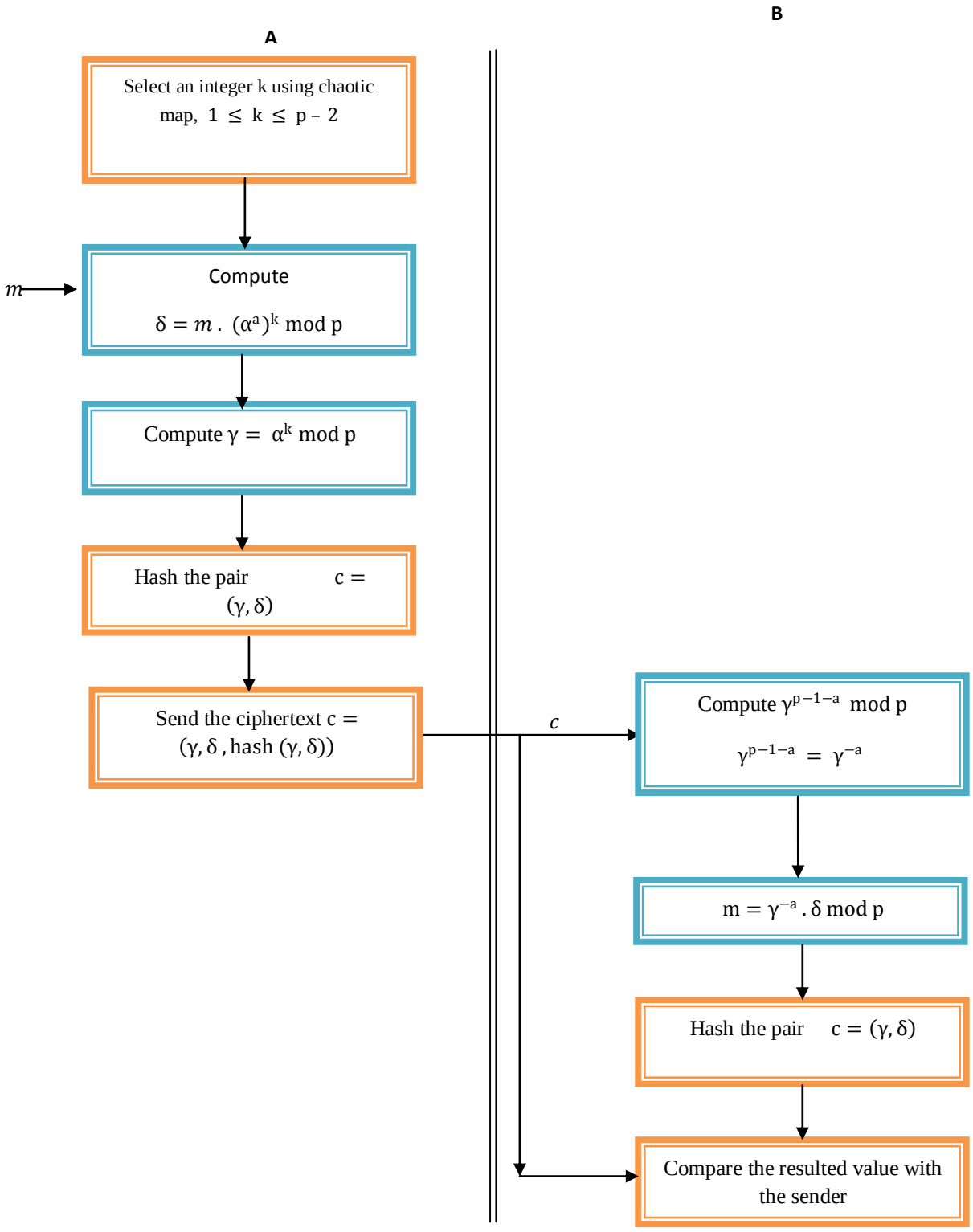


Fig. 4.4: The improved Elgamal cryptosystem diagram.

Chapter 5

The Proposed Text Private Key Cryptosystem

The proposed private key cryptosystem is based on chaotic maps and quasi groups, it has two types: stream cipher mode and block cipher mode, in the following sections those modes and their security analysis will be explored.

5.1 The Proposed Private Key Cryptosystem (Stream Cipher Mode)

Here, we use both of chaotic maps and quasi group to build an efficient cryptosystem. The stream cipher mode is adopted in this section.

5.1.1 Design

The general block diagram, illustrates the phases of the cryptosystem is shown in Fig. 5.1. The encryption process consists of five stages, the first phase is initialization, where the variables, chaotic maps and quasi groups are built, then preprocessing phase where the message is converted into numbers, while in chaotic transposition phase, the logistic map is used to rearrange the message, the quasi transposition phase mix up the massage using quasi group, the last phase is substitution in which the chaotic maps are used in order to change the cipher values to be unreadable date.

The steps involved in the proposed encryption/decryption process are given below.

Initialization Stage:

1. Getting the secret key
2. Build the quasi groups in the encryption process, and build the inverse quasi group in the decryption process.
3. Generate variables using chaotic maps

Encryption process:

Purpose: encrypting the data.

Inputs: the original message.

Outputs: the encrypted message.

Procedure:

Step 1: Initialize the required variables.

Step 2: Convert the message into numbers.

Step 3: Mix up the plaintext using chaotic maps transposition.

Step 4: Shuffle the plaintext via quasi groups transposition.

Step 5: Apply chaotic maps substitution on the plaintext.

Decryption process:

Purpose: decrypting the message.

Inputs: the encrypted message.

Outputs: the original message.

Procedure:

Step 1: Initialize the required variables.

Step 2: Apply inverse chaotic maps substitution on the ciphertext.

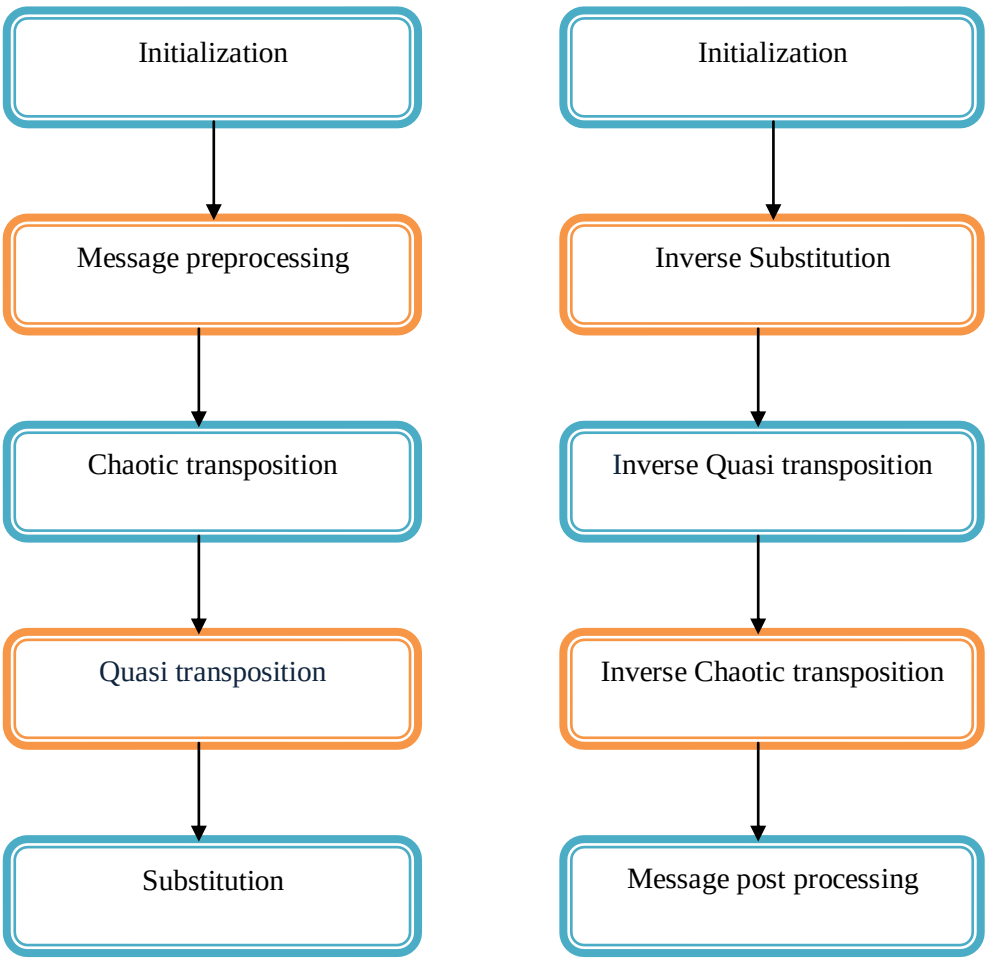
Step 3: Apply inverse quasi groups' transposition.

Step 4: Apply inverse chaotic map transposition.

Step 5: Message post processing.

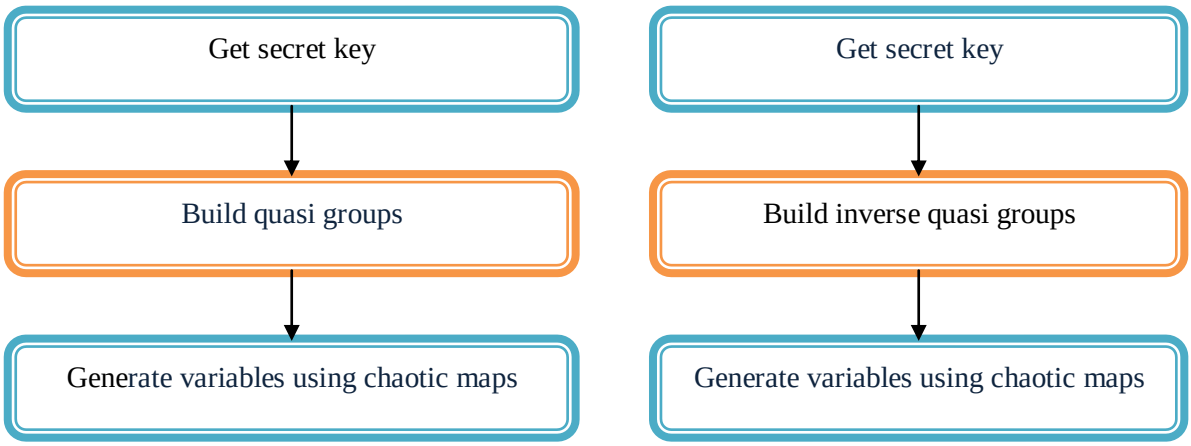
The initialization steps in encryption side are demonstrated in Fig. 5.2.a, while Fig. 5.2.b shows the initialization steps in decryption side.

The difference between initialization steps in encryption side and initialization steps in decryption side is the second step in the encryption side the construction of quasi groups, but in the decryption side the construction of inverse quasi groups.



a)Encryption b) Decryption

Fig. 5.1: a) Block diagram showing the private key cryptosystem encryption part, b) block diagram showing the private key cryptosystem decryption part



a)Encryption b) Decryption

Fig. 5.2: a) Block diagram showing the initialization steps in encryption part , b) Block diagram showing the initialization steps in decryption part .

5.1.2 Implementation and Results

The proposed cryptosystem has been implemented using Matlab and the simulation results were observed on Core2 Duo, 2.40GHz with 2 GB RAM , We have acquired outstanding results, as shown in the following figures the ciphertext and the plaintext are totally different.

To show the performance of the proposed cryptosystem, we compared between two plaintexts, the first one is consist of constant value particularly the letter 'E' and the other is ordinary text, Fig. 5.3 shows the first plaintext, its constant line, and its ciphertext shown in Fig. 5.4, its zigzag line, which means that the constant value in plaintext converts to multiple values.

The second plaintext shown in Fig. 5.5 while the resulted ciphertext shown in Fig. 5.6, here the original text is a recurring of some characters, but the resulted ciphertext doesn't demonstrate the repeated pattern.

As observed from these figures, the ciphertext shows a complete random behavior.

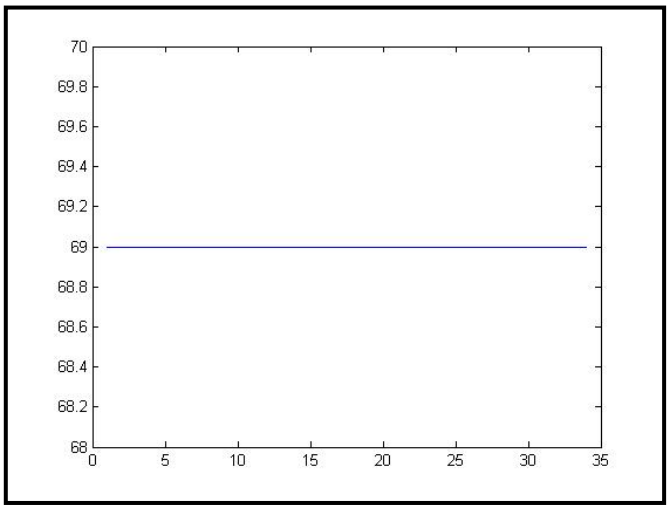


Fig. 5.3: The constant plaintext.

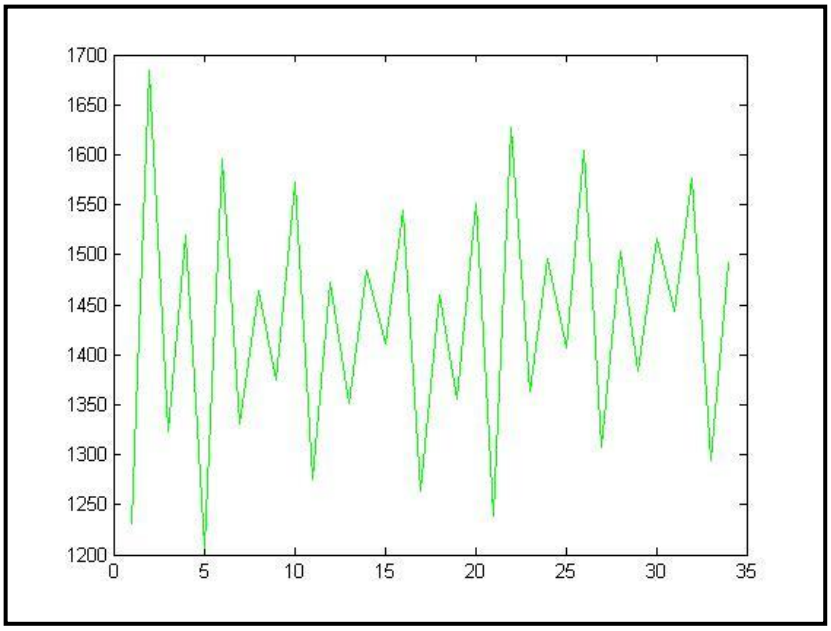


Fig. 5.4: The ciphertext of constant plaintext.

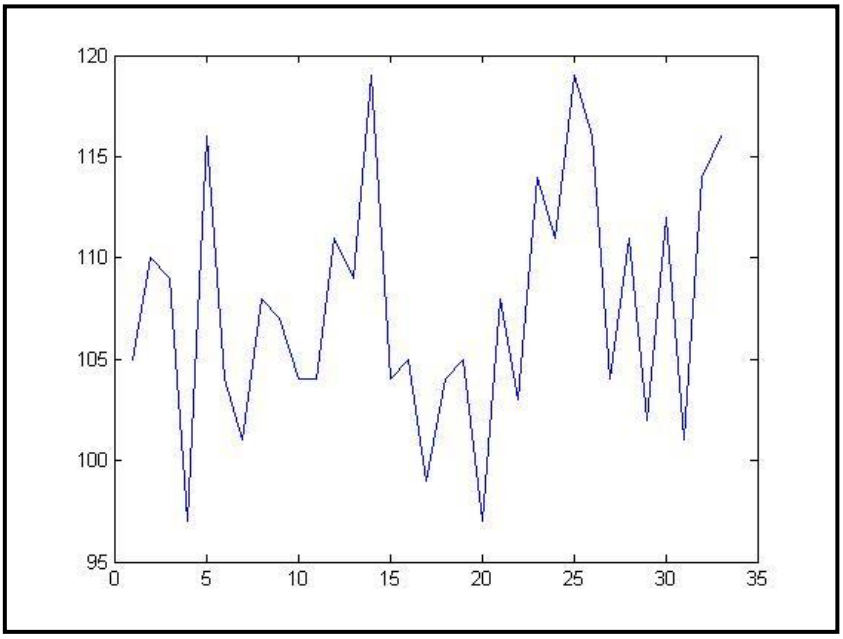


Fig. 5.5: The case 2 plaintext.

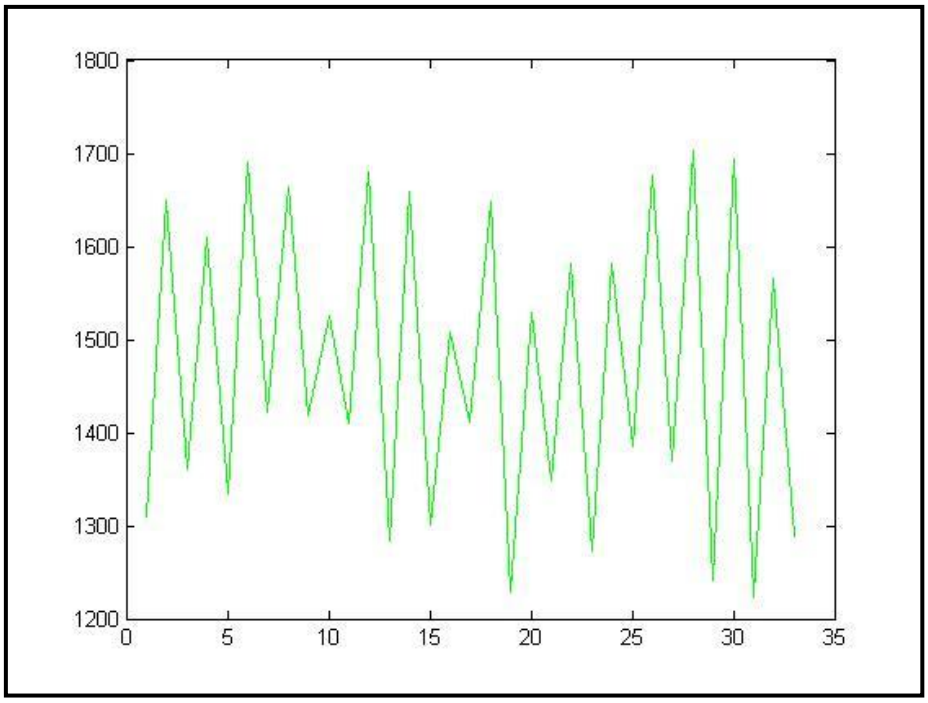


Fig. 5.6: the ciphertext of case 2.

5.2 Security Analysis (Stream Cipher Mode)

The following results demonstrate how much the entropy is maximized, and how the proposed cryptosystem does destroy any existing patterns in the input plaintext, which is desirable for a good cryptosystem.

5.2.1 Entropy

Entropy is a statistical measure of randomness that can be used to characterize the texture of the input. Entropy is a measure of disorder, or more precisely unpredictability.

English text has fairly low entropy. In other words, it is fairly predictable. Even if we don't know exactly what is going to come next, we can be fairly certain that, for example, there will be many more e's than z's, or that the combination 'qu' will be much more common than any other combination with a 'q' in it and the combination 'th' will be more common than any of them [57].

Case 1:

The plaintext:

"inmathelkhhomwhichialgrowthofpertuxcvrbationslijintheirfyhvnitialconditionsthebehia aviityudjkflaorofthat"

The entropy of the plaintext = 3.7600, while the entropy of the ciphertext = 6.5466.

Case 2:

The plaintext:

"EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE"

The entropy of the plaintext = 0, while the entropy of the ciphertext = 4.8580.

It's observed that the entropy is increased significantly, so the proposed cryptosystem destroys the characteristic of the sent message by increasing the randomness in it.

5.2.2 N-grams

N-grams model is a type of probabilistic model for predicting the next item in such a sequence. The N-gram analysis determines the frequency of different N-grams in a text. Especially the gaps between equal N-grams can potentially be very useful for cracking a cipher because they can point to the key length [58].

An n-gram of size 2 is referred to as a "bigram" (or, less commonly, a "digram"); size 3 is a "trigram" [59].

Case 1:

The plaintext:

"inmathelkhhomwhichialgrowthofpertuxcvrbationslijintheirfyhvnitialconditionsthebehia aviityudjkflaorofthat"

Table 5.1: The histogram, bigram and trigram of the case 1 plaintext.

Nr.	Histogram			Bigram			Trigram		
1.	**I**	14	13.46%	**TH**	5	4.81%	**THE**	3	2.88%
2.	**H**	11	10.58%	**TI**	3	2.88%	**IAL**	2	1.92%
3.	**T**	11	10.58%	**IT**	3	2.88%	**TIO**	2	1.92%
4.	**A**	8	7.69%	**HI**	3	2.88%	**ION**	2	1.92%
5.	**O**	8	7.69%	**ON**	3	2.88%	**ITI**	2	1.92%
6.	**N**	6	5.77%	**IA**	3	2.88%	**HIA**	2	1.92%
7.	**R**	5	4.81%	**HE**	3	2.88%	**ONS**	2	1.92%
8.	**L**	5	4.81%	**AT**	3	2.88%	**ORO**	1	0.96%
9.	**E**	5	4.81%	**RO**	2	1.92%	**NTH**	1	0.96%
10.	**F**	4	3.85%	**HO**	2	1.92%	**OFP**	1	0.96%
11.	**V**	3	2.88%	**IN**	2	1.92%	**NST**	1	0.96%
12.	**C**	3	2.88%	**IO**	2	1.92%	**OFT**	1	0.96%
13.	**S**	2	1.92%	**OF**	2	1.92%	**OND**	1	0.96%
14.	**B**	2	1.92%	**AL**	2	1.92%	**OMW**	1	0.96%
15.	**U**	2	1.92%	**NS**	2	1.92%	**NSL**	1	0.96%

16.	**W**	2	1.92%	**MW**	1	0.96%	**LKH**	1	0.96%
17.	**D**	2	1.92%	**OW**	1	0.96%	**LIJ**	1	0.96%
18.	**J**	2	1.92%	**MA**	1	0.96%	**LGR**	1	0.96%
19.	**K**	2	1.92%	**ND**	1	0.96%	**LCO**	1	0.96%
20.	**M**	2	1.92%	**OR**	1	0.96%	**MAT**	1	0.96%
21.	**Y**	2	1.92%	**OM**	1	0.96%	**MWH**	1	0.96%
22.	**X**	1	0.96%	**NM**	1	0.96%	**NMA**	1	0.96%
23.	**P**	1	0.96%	**NI**	1	0.96%	**NIT**	1	0.96%
24.	**G**	1	0.96%	**NT**	1	0.96%	**NDI**	1	0.96%
25.				**RF**	1	0.96%	**OWT**	1	0.96%
26.				**VR**	1	0.96%	**RBA**	1	0.96%
27.				**VN**	1	0.96%	**VNI**	1	0.96%
28.				**VI**	1	0.96%	**VII**	1	0.96%
29.				**WH**	1	0.96%	**UXC**	1	0.96%
30.				**WT**	1	0.96%	**UDJ**	1	0.96%

The ciphertext:

" s @Hfù ¬ ¬ -K±¾h8 ? ©à c v 5 N` o $ ¼ w6v> Hô%`6 ¸ uo½5¾MïZdûn ¸oâ¬2[â ¢

- à ¹H a+"

Table 5.2: The histogram, bigram and trigram of the case 1 ciphertext.

Nr.	Histogram			Bigram			Trigram		
1.	**H**	4	19.05%	**NO**	2	9.52%	**SHF**	1	4.76%
2.	**O**	3	14.29%	**SH**	1	4.76%	**OWV**	1	4.76%
3.	**N**	2	9.52%	**OW**	1	4.76%	**OMZ**	1	4.76%
4.	**V**	2	9.52%	**OM**	1	4.76%	**UOM**	1	4.76%
5.	**Z**	1	4.76%	**UO**	1	4.76%	**VHU**	1	4.76%
6.	**S**	1	4.76%	**VH**	1	4.76%	**ZDN**	1	4.76%
7.	**U**	1	4.76%	**ZD**	1	4.76%	**WVH**	1	4.76%
8.	**W**	1	4.76%	**WV**	1	4.76%	**VNO**	1	4.76%
9.	**M**	1	4.76%	**VN**	1	4.76%	**OHA**	1	4.76%
10.	**C**	1	4.76%	**OH**	1	4.76%	**NOW**	1	4.76%
11.	**D**	1	4.76%	**CV**	1	4.76%	**HCV**	1	4.76%
12.	**F**	1	4.76%	**HC**	1	4.76%	**FKH**	1	4.76%
13.	**K**	1	4.76%	**HA**	1	4.76%	**DNO**	1	4.76%
14.	**A**	1	4.76%	**FK**	1	4.76%	**HFK**	1	4.76%
15.				**HF**	1	4.76%	**HUO**	1	4.76%

16.				**HU**	1	4.76%	**NOH**	1	4.76%
17.				**MZ**	1	4.76%	**MZD**	1	4.76%
18.				**KH**	1	4.76%	**KHC**	1	4.76%
19.				**DN**	1	4.76%	**CVN**	1	4.76%

It observed from the previous tables, that the characteristics of the plaintext and ciphertext are totally different; mainly the occurrence of the characters is changed in the ciphertext from it in the plaintext.

Case 2:

The plaintext:

"EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE"

Table 5.3: The histogram, bigram and trigram of the case 2 plaintext.

Nr.	**Histogram**			**Bigram**			**Trigram**		
1.	**E**	29	100%	**EE**	14	48.28%	**EEE**	9	31.03%

The ciphertext:

"+5âQS¯«N)©x¨((³¤F"

Table 5.4: The histogram, bigram and trigram of the case 2 ciphertext.

Nr.	**Histogram**			**Bigram**			**Trigram**		
1.	**X**	1	20%	**XF**	1	20%	**SNX**	1	20%
2.	**S**	1	20%	**SN**	1	20%	**QSN**	1	20%
3.	**Q**	1	20%	**QS**	1	20%	**NXF**	1	20%
4.	**N**	1	20%	**NX**	1	20%			
5.	**F**	1	20%						

It is noticed that the letter 'E' didn't appear in the ciphertext and there is no character as much as a recurring character 'E' in the plaintext. So, the proposed cryptosystem is secure against the statistics analysis.

5.2.3 Autocorrelation

Autocorrelation is a mathematical tool for finding repeating patterns, such as the presence of a periodic signal. It refers to the correlation of a time series with its own past and future values [60].

It may be observed from Figs. 5.7 and 5.9 that the output characteristics of both cases are a very similar even though the inputs are very different.

In the other side Fig. 5.8 demonstrates the pattern in case 2 text, while Fig. 5.9 shows how does the proposed cryptosystem destroy the existing pattern, which is desirable for a good cryptosystem.

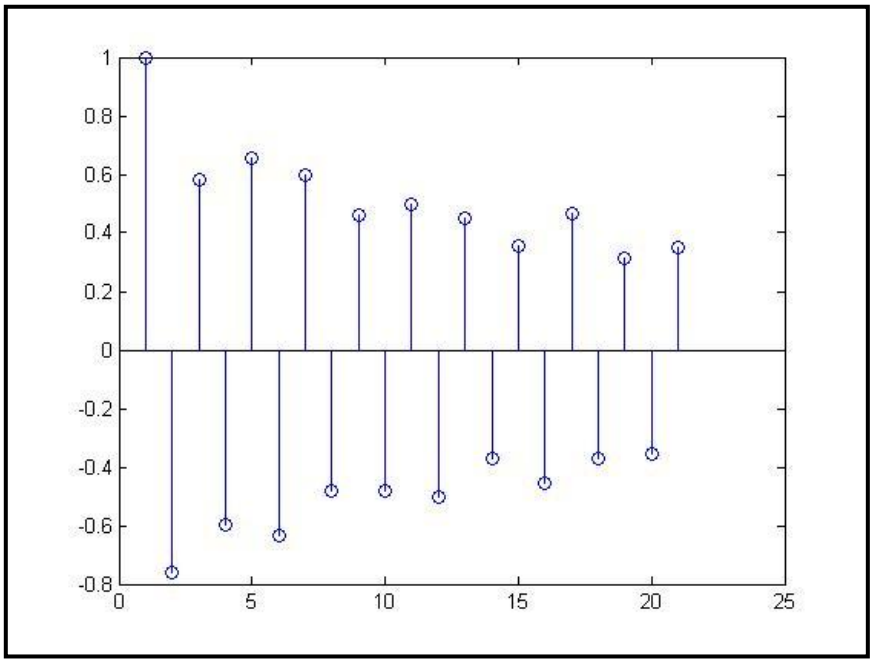


Fig. 5.7: The autocorrelation of case 1 ciphertext.

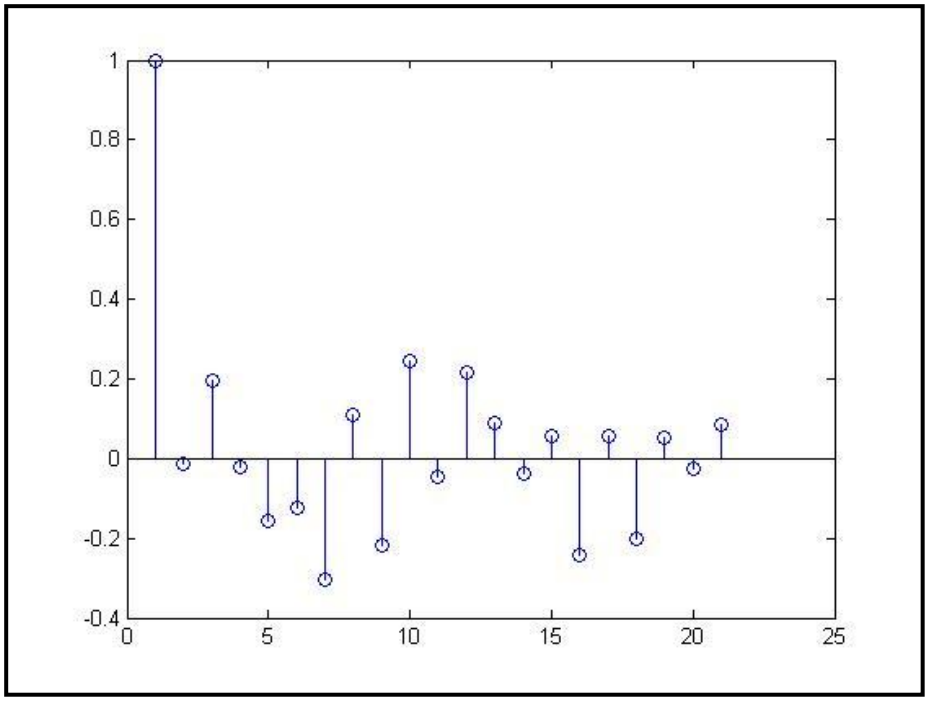


Fig. 5.8: The autocorrelation of case 2 plaintext.

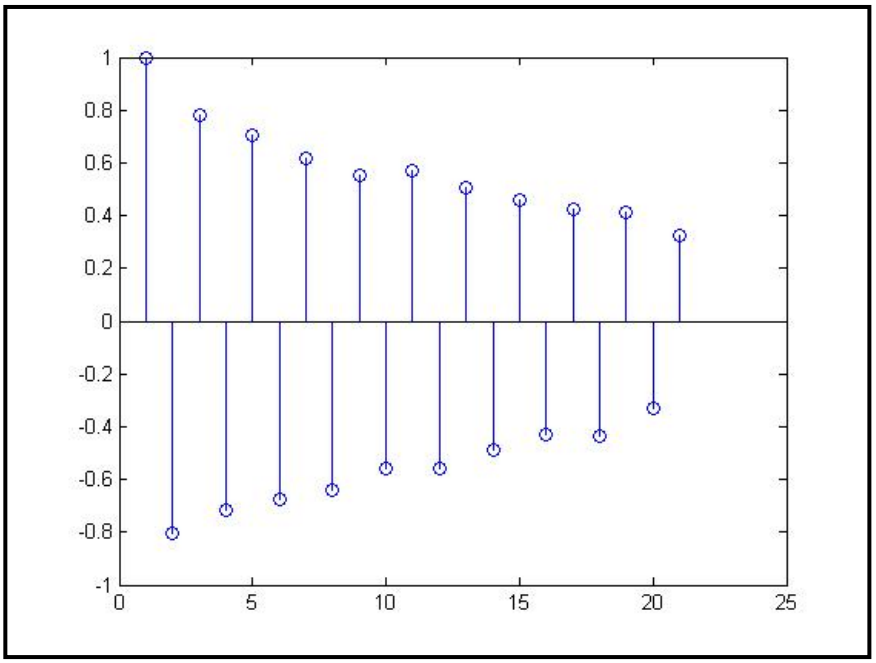


Fig. 5.9: The autocorrelation of case 2 ciphertext.

The autocorrelation of the ciphertext is decreasing as shown in Figs. 5.7 and 5.9, this pattern is nearly appear for all ciphertexts, that means whatever the plain text, the resulted autocorrelation of ciphertext will be meaningless for hacking purposes.

5.3 The Proposed Private Key Cryptosystem (Block Cipher Mode)

The proposed cryptosystem which mentioned in section 5.2 is stream cipher mode; here I examine the proposed cryptosystem in block cipher mode.

It's known that a stream cipher mode is extremely simple and fast, but requires synchronization, while a block cipher mode output is random-looking and has good statistical properties.

5.3.1 Design

The message is divided into 128 byte blocks and each block is encrypted separately using particular key, the user determines the number of encryption boxes, in the following experiments I have four encryption boxes with different keys as shown in Fig. 5.10.

Each encryption box works individually with different key, so more confusion and diffusion are obtained.

5.3.2 Implementation and Results

The experiments are applied in two plaintexts:

Case 1:

Plaintext:

"EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
E"

Ciphertext:

"]eêyµ!}±Mƒ¤1§nû0êQ<è=× ¸IT¯pŒG« {c K3·"

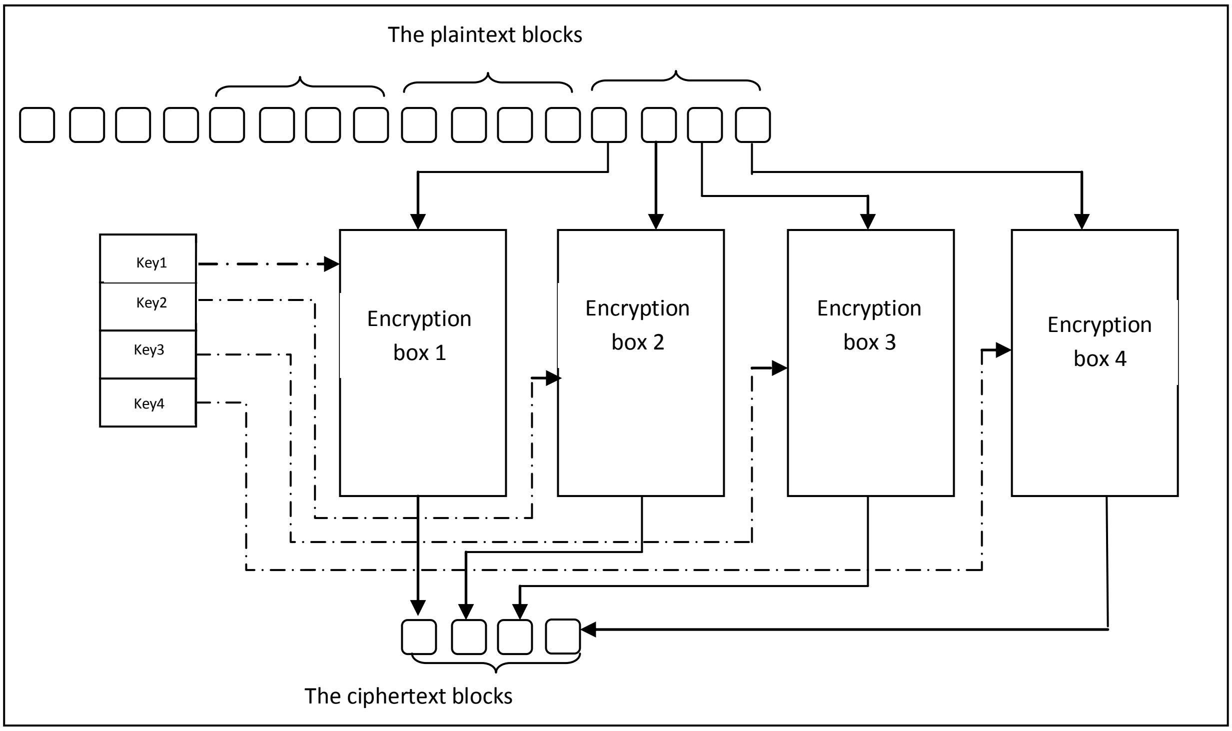


Fig. 5.10: The proposed private key cryptosystem in block cipher mode.

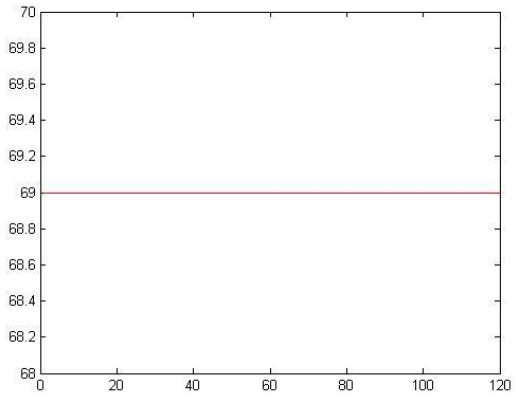


Fig. 5.11: Case 1 plaintext.

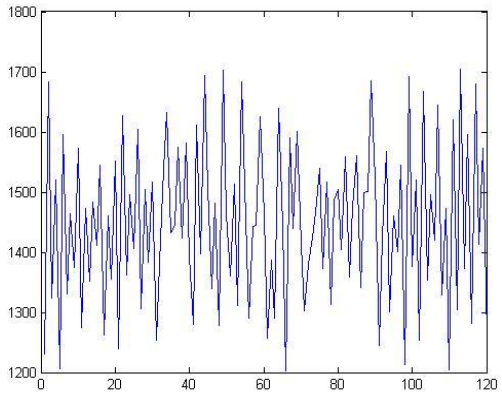


Fig. 5.12: Case 1 ciphertext.

As in previous section, although the plaintext is constant, the ciphertext has many values, this make the cryptanalysis difficult.

Case 2:

Plaintext:

"inmathelkhhomwchialgrowtfhofpeinmathelkhhomwchialgrowtfhofpeinmathelkhhomwchialgrowtfhofpeinmathelkhhomwchialgrowtfhofpe "

Ciphertext:

" ?{[®d I5 g YQ ' ïNü·¾× ƒµ ¨çWü ¾k¢="

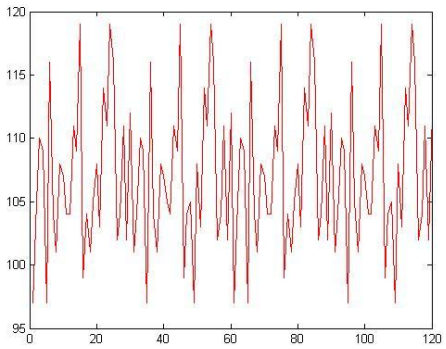


Fig. 5.13: Case 2 plaintext.

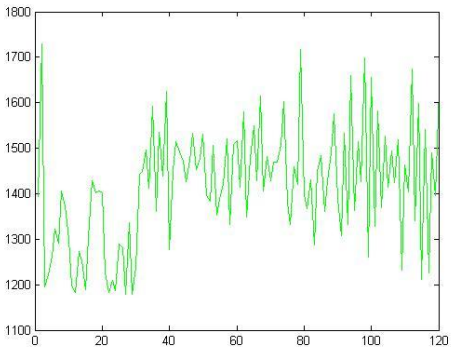


Fig. 5.14: Case 2 ciphertext.

Here the plaintext is a repeated of certain characters, but the resulted ciphertext doesn’t demonstrate the periodically pattern, it's give a totally different pattern.

It may be observed from the figures, that the encrypted sequence is essentially one value, even though the output sequence is very random. Study the characteristics of the output data in the next section.

5.4 Security Analysis (Block Cipher Mode)

The following results demonstrate how much the entropy is maximized, and how the proposed cryptosystem does destroy any existing patterns in the input plaintext, which is desirable for a good cryptosystem.

5.4.1 Entropy

Case 1:

The entropy of the plaintext = 0, while the entropy of the ciphertext = 6.6736.

Case 2:

The entropy of plaintext= 3.7600, while the entropy of the ciphertext= 6.6506.

5.4.2 N-grams

Case 1:

Table 5.5: The histogram, bigram and trigram of case 1 plaintext.

Nr.	Histogram			Bigram			Trigram		
1.	**E**	120	100%	**EE**	58	50 %	**EEE**	9	32.74%

Table 5.6: The histogram, bigram and trigram of case 1 ciphertext.

Nr.	Histogram			Bigram			Trigram		
1.	**P**	1	9.09%	**PG**	1	9.09%	**QIT**	1	9.09%
2.	**Q**	1	9.09%	**QI**	1	9.09%	**TPG**	1	9.09%
3.	**T**	1	9.09%	**TP**	1	9.09%	**YMN**	1	9.09%
4.	**Y**	1	9.09%	**YM**	1	9.09%	**PGC**	1	9.09%
5.	**N**	1	9.09%	**NQ**	1	9.09%	**NQI**	1	9.09%
6.	**M**	1	9.09%	**MN**	1	9.09%	**GCK**	1	9.09%
7.	**E**	1	9.09%	**EY**	1	9.09%	**ITP**	1	9.09%
8.	**G**	1	9.09%	**GC**	1	9.09%	**MNQ**	1	9.09%
9.	**I**	1	9.09%	**IT**	1	9.09%	**EYM**	1	9.09%
10.	**K**	1	9.09%	**CK**	1	9.09%			
11.	**C**	1	9.09%						

Case 2:

Table 5.7: The histogram, bigram and trigram of case 2 plaintext.

Nr.	Histogram			Bigram			Trigram		
1.	**H**	20	16.67%	**HO**	8	6.67%	**NMA**	4	3.33%
2.	**O**	12	10%	**NM**	4	3.33%	**OFP**	4	3.33%
3.	**A**	8	6.67%	**OF**	4	3.33%	**MWC**	4	3.33%
4.	**L**	8	6.67%	**MW**	4	3.33%	**MAT**	4	3.33%
5.	**M**	8	6.67%	**LK**	4	3.33%	**LGR**	4	3.33%
6.	**W**	8	6.67%	**MA**	4	3.33%	**LKH**	4	3.33%
7.	**I**	8	6.67%	**OM**	4	3.33%	**OMW**	4	3.33%
8.	**E**	8	6.67%	**OW**	4	3.33%	**OWT**	4	3.33%
9.	**F**	8	6.67%	**WC**	4	3.33%	**WCH**	4	3.33%

10.	**T**	8	6.67%	**WT**	4	3.33%	**WTF**	4	3.33%
11.	**R**	4	3.33%	**TH**	4	3.33%	**THE**	4	3.33%
12.	**P**	4	3.33%	**TF**	4	3.33%	**TFH**	4	3.33%
13.	**K**	4	3.33%	**PE**	4	3.33%	**ROW**	4	3.33%
14.	**C**	4	3.33%	**RO**	4	3.33%	**KHH**	4	3.33%
15.	**G**	4	3.33%	**LG**	4	3.33%	**INM**	4	3.33%
16.	**N**	4	3.33%	**KH**	4	3.33%	**FHO**	4	3.33%
17.				**FH**	4	3.33%	**FPE**	4	3.33%
18.				**FP**	4	3.33%	**ELK**	4	3.33%
19.				**EL**	4	3.33%	**CHI**	4	3.33%
20.				**CH**	4	3.33%	**ATH**	4	3.33%

Table 5.8: The histogram, bigram and trigram of case 2 ciphertext.

Nr.	**Histogram**			**Bigram**			**Trigram**		
1.	**Q**	1	12.5%	**WK**	1	12.5%	**QNW**	1	12.5%
2.	**W**	1	12.5%	**YQ**	1	12.5%	**YQN**	1	12.5%
3.	**Y**	1	12.5%	**QN**	1	12.5%	**NWK**	1	12.5%
4.	**N**	1	12.5%	**NW**	1	12.5%	**IGY**	1	12.5%
5.	**K**	1	12.5%	**GY**	1	12.5%	**GYQ**	1	12.5%
6.	**G**	1	12.5%	**IG**	1	12.5%	**DIG**	1	12.5%
7.	**I**	1	12.5%	**DI**	1	12.5%			
8.	**D**	1	12.5%						

It's clear that the ciphertexts have new characters which not exist in the original plaintexts; also, the characteristics of the plaintext and ciphertext are totally different; mainly the occurrence of the characters is changed in the ciphertext from it in the plaintext.

5.4.3 Autocorrelation

Case 1:

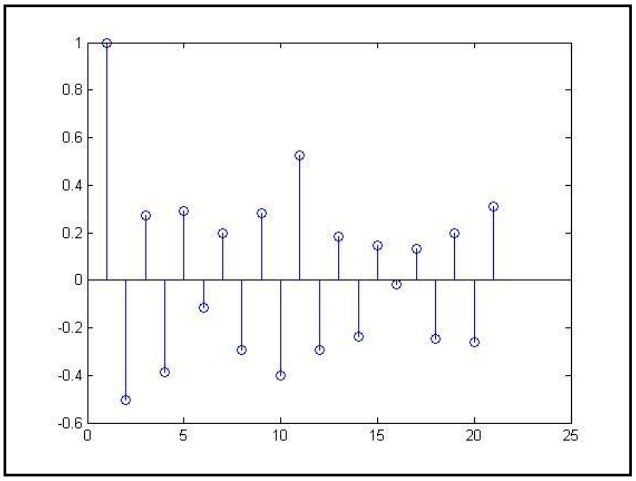


Fig. 5.15: Autocorrelation of case 1 ciphertext.

Case 2:

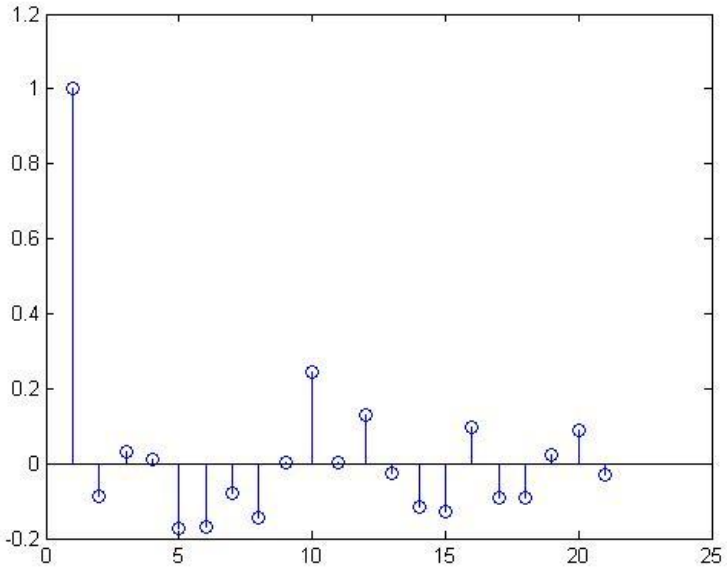


Fig. 5.16: Autocorrelation of case 2 plaintext.

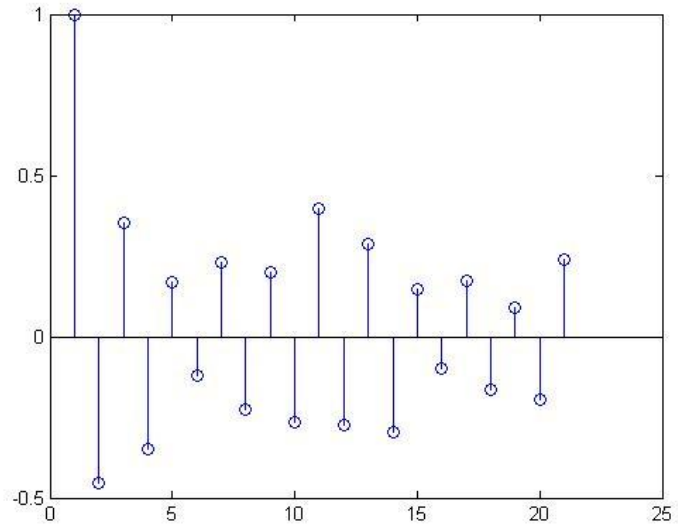


Fig. 5.17: Autocorrelation of case 2 ciphertext.

It may be noticed from figs. 5.15 and 5.17 that the output characteristics of both cases are a very similar even though the inputs are very different. In the other side Fig. 5.16 demonstrates the pattern in case 2 text, while Fig. 5.17 shows how does the proposed cryptosystem destroy the existing pattern, which is desirable for a good cryptosystem.

The result shows more randomness and confirms the diffusion property with block cipher more than it when using the stream cipher mode; on the other hand stream

ciphers advantage is that they are much faster than block cipher. In my work both choices are available.

5.5 Summery

In this chapter, the proposed private key cryptosystem is introduced, in two modes, stream cipher mode and block cipher mode.

The encryption process is divided into five phases, first of all, the parameters and quasi groups are initialized in the initialization phase, and the preprocessing phase is the preparing stage, where the plaintext is converted into numbers, which is more suitable to deal with the chaotic and quasi group's equations. The chaotic transposition stage, here involve the logistic map and the plaintext in one equation to eliminate the plaintext characteristics. The quasi transposition phase is used to shuffle the plaintext via quasi groups. The last stage is substitution where the chaotic maps are used to transform the cipher values to be unreadable date.

The implementation results show that the proposed cryptosystem has acquired outstanding results, the proposed cryptosystem destroys any existing patterns in the input, and also, it maximizes entropy. Moreover, the n-grams show that the proposed cryptosystem is secure against the statistics analysis.

The results show that the block cipher mode gives higher entropy than the steam cipher mode, the entropy of the constant plaintext = 0, while the entropy of the ciphertext using stream cipher mode = 4.8580, but the entropy of the ciphertext using block cipher mode = 6.6736.

For ordinary plaintext, the entropy of the plaintext = 3.7600, while the entropy of the ciphertext using stream cipher mode = 6.5466, but the entropy of the ciphertext using block cipher mode = 6.6506.

This result refers to encryption process use different keys, i.e. different quasi groups and different chaotic maps parameters, which means more and more randomness in the resulted ciphertexts.

Chapter 6

The Proposed Voice Private Key Cryptosystem

The proposed voice cryptosystem is based on chaotic maps and quasi groups, it deals with voice waves, its design, implementation and security analysis will be explored in the next sections.

6.1 The Proposed Voice Cryptosystem.

Here, we use both of chaotic maps and quasi group to construct a proficient voice cryptosystem.

6.1.1 Design

The speech encryption has always been a very important part of military communications. The proposed schema is suitable for this kind of data; the following figure illustrates the voice encryption/ decryption processes.

Initialization Stage:

1. Getting the secret key.
2. Build the quasi groups in the encryption process, and build the inverse quasi group in the decryption process.
3. Generate variables using chaotic maps.

Encryption process:

Purpose: encrypting the voice wave.

Inputs: the original voice wave.

Outputs: the encrypted wave.

Procedure:

Step 1: Initialize the required variables.

Step 2: Convert the voice wave into one-dimension array.

Step 3: Use the quasi group to shuffle the array.

Step 4: Apply chaotic maps on the array to eliminate the voice characteristics.

Step 5: Use the chaotic maps to transform the cipher values to be illegible date.

Decryption process:

Purpose: decrypting the voice wave.

Inputs: the encrypted wave. .

Outputs: the original voice wave.

Procedure:

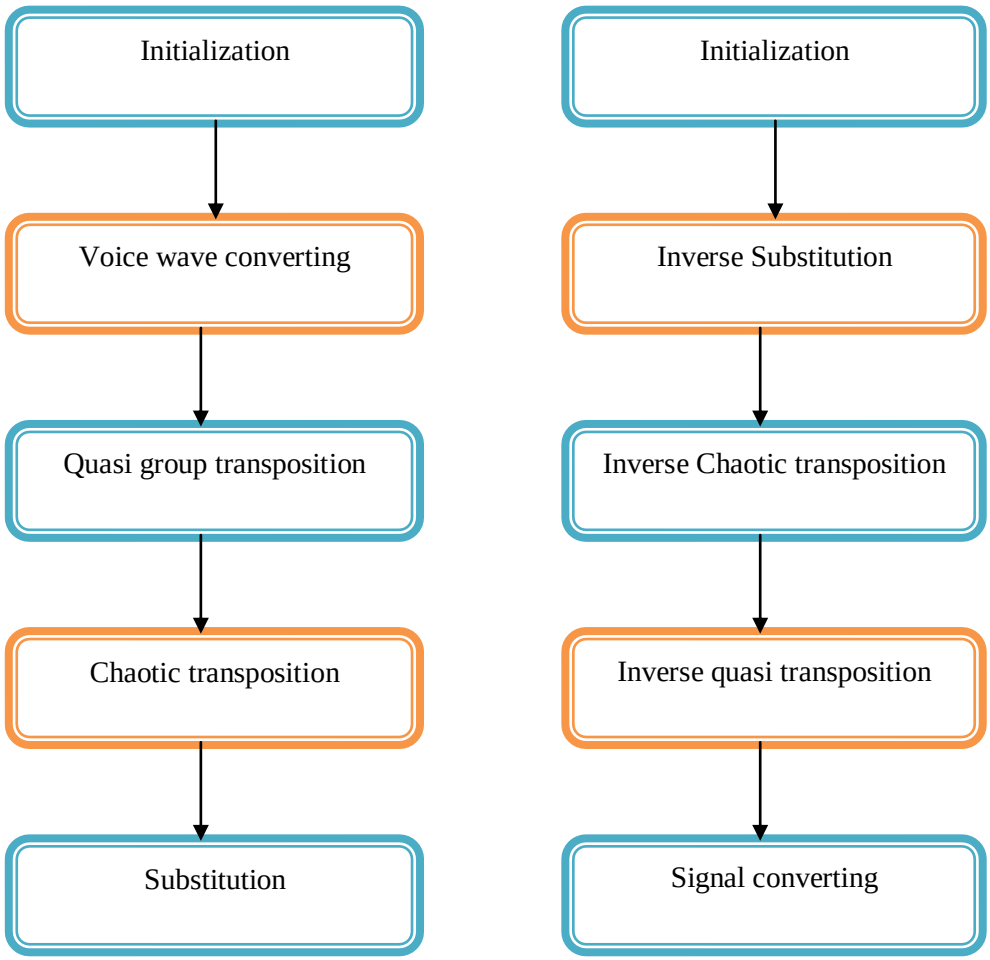
Step 1: Initialize the required variables.

Step 2: substitution phase: use the chaotic maps, in the inverse way.

Step 3: Apply chaotic maps on the array, in order to inverse the transposition effects.

Step 4: Use the inverse quasi group to reshuffle the array.

Step 5: transform the array into wave format.

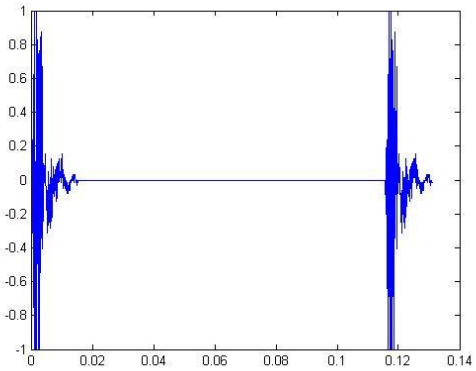


a) Encryption b) Decryption

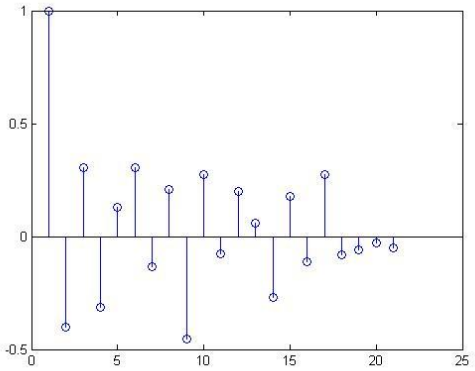
Fig. 6.1: a) Block diagram showing the proposed voice cryptosystem encryption part, b) block diagram showing the proposed voice cryptosystem decryption part.

6.1.2 Implementation and Results

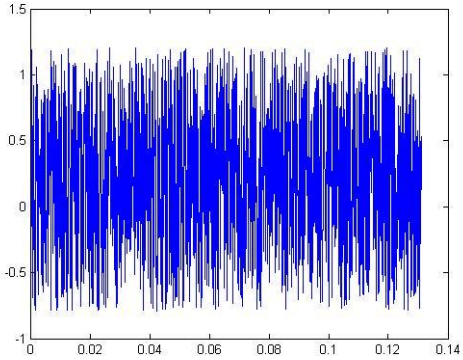
The proposed cryptosystem has been implemented using Matlab and the simulation results were observed on Core2 Duo, 2.40GHz with 2 GB RAM, The achieved results shown in the following figures.



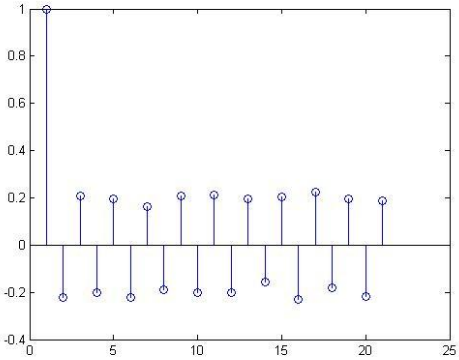
a) The original wave



b) The original autocorrelation

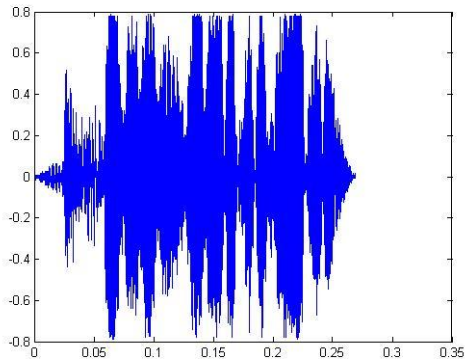


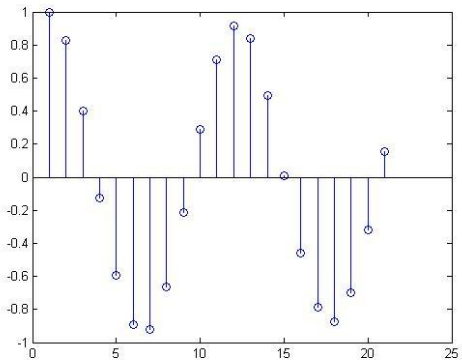
c) The encrypted wave



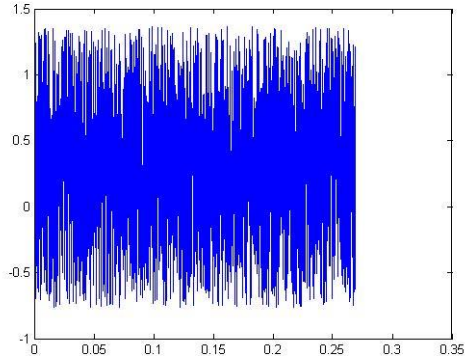
d)The encrypted autocorrelation

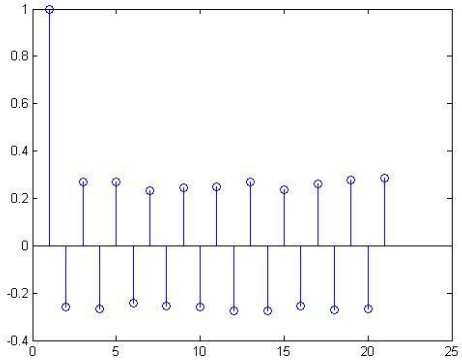
Fig. 6.2: Experiment 1





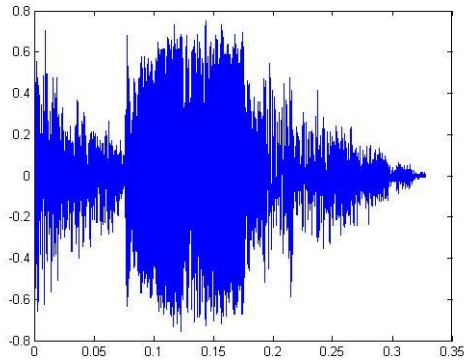
a) The original wave **b) The original autocorrelation**

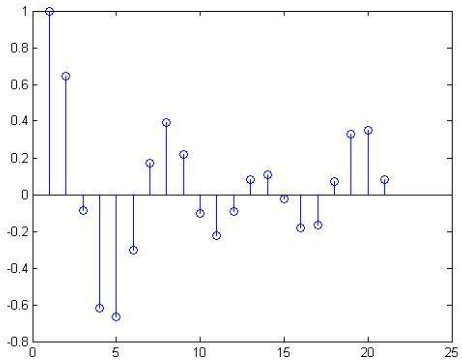




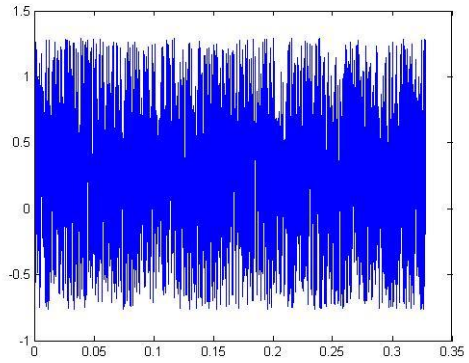
c) The encrypted wave **d) The encrypted autocorrelation**

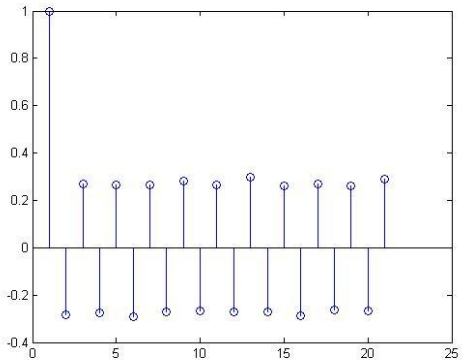
Fig. 6.3: Experiment 2





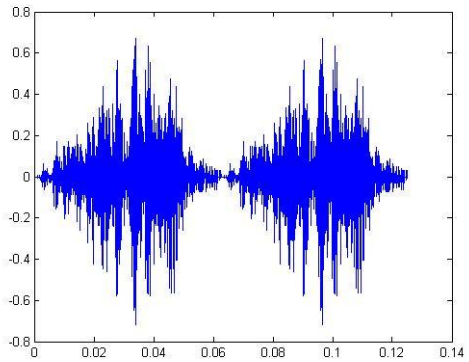
a) The original wave **b) The original autocorrelation**



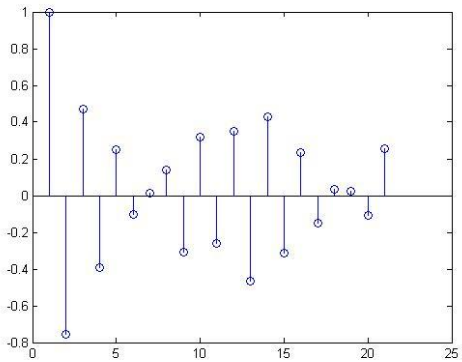


c) The encrypted wave **d) The encrypted autocorrelation**

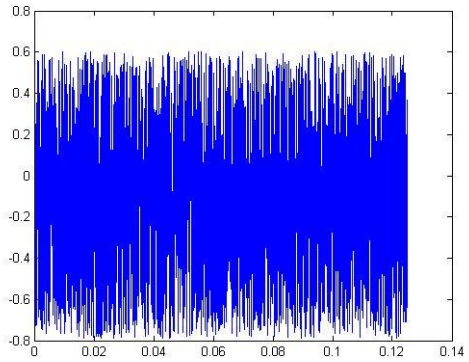
Fig. 6.4: Experiment 3



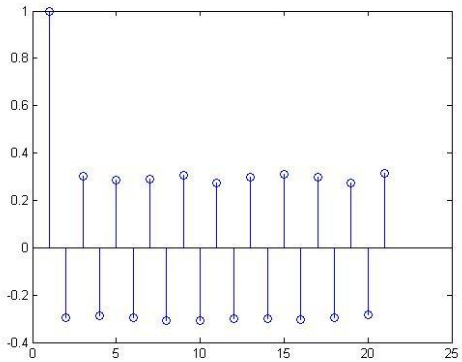
a) The original wave



b) The original autocorrelation



c) The encrypted wave



d) The encrypted autocorrelation

Fig. 6.5: Experiment 4

6.2 Security Analysis

As shown in the following subsections, the proposed voice cryptosystem is secure against statistical attacks; besides, it destroys the signal pattern, which is advantageous for a good cryptosystem.

6.2.1 Autocorrelation

It may be observed from the previous figures that the output autocorrelation of all cases are a very similar even though the inputs are very different and the encrypted waves give nearly the same sound, i.e. the original wave loses its own features, that is what was required.

6.2.2 Entropy

Table 6.1: The original and encrypted signal entropy values

Entropy of	Original signal	Encrypted signal
Experiment 1	1.7667	9.9985
Experiment 2	7.3172	10.4077
Experiment 3	6.9172	10.4382
Experiment 4	5.9511	10.2507

It's clear that the proposed cryptosystem maximizes the entropy, whatever the signal shape.

6.3 Summery

In this chapter the proposed private key cryptosystem is customized for voice signals, it shows good results, the characteristics of signal are eliminated and the dataset encrypted waves are nearly similar to each other, thus, the proposed cryptosystem is suitable for audio usage.

Chapter 7

The Proposed Image Private Key Cryptosystem

Using of images has increased recently, furthermore the internet and multimedia technologies are the most important ways for communication of digital information, so image security has become an important topic in the current computer world, for example, it is important to protect the diagrams of army emplacements, the diagrams of bank building construction, and the important data captured by military satellites. In addition, the number of computer crimes has increased recently.

7.1 Introduction

A digital image is defined by an array of individual pixels and each pixel has its own value. The array, and thus the set of pixels, is called a bitmap. If we have an image of 512 pixels × 512 pixels, it means that the data for the image must contain information about 262144 pixels [61].

We shall consider four basic types of images: An **indexed image** consists of a data matrix, X, and a color map matrix, map. Map is an m-by-3 array of class double containing floating-point values in the range [0, 1]. Each row of map specifies the red, green, and blue components of a single color. An indexed image uses "direct mapping" of pixel values to color map values. The color of each image pixel is determined by using the corresponding value of X as an index into map. The value 1 points to the first row in map, the value 2 points to the second row, and so on.

In a Binary image each pixel is just black or white. Since there are only two possible values for each pixel, we only need one bit per pixel. Such images can therefore be very efficient in terms of storage. Images for which a binary representation may be suitable include text (printed or handwriting), fingerprints, or architectural plans.

In a Grayscale image each pixel is a shade of grey, normally from (black) to (white). This range means that each pixel can be represented by eight bits, or exactly one byte. This is a very natural range for image file handling. Other grayscale ranges are used, but generally they are a power of 2. Such images arise in medicine (X-rays), images of printed works, and indeed 256 different grey levels are sufficient for the recognition of most natural objects.

True color, or RGB, here each pixel has a particular color, that color being described by the amount of red, green and blue in it. If each of these components has a range 0-255, this gives a total of 2553 = 16,777,216 different possible colors in the image. This is enough colors for any image. Since the total number of bits required for each pixel is 24, such images are also called 24-bit color images. Such an image may be considered as consisting of a stack of three matrices; representing the red, green and blue values for each pixel. This means that for every pixel there correspond three values.
A particular RGB color space is defined by the three chromaticities of the red, green, and blue additive primaries, and can produce any chromaticity that is the triangle defined by those primary colors [62].

The YCbCr color space is widely used for digital video. In this format, luminance information is stored as a single component (Y), and chrominance information is stored as two color-difference components (Cb and Cr). Cb represents the difference between the blue component and a reference value. Cr represents the difference between the red component and a reference value [63].

In the proposed schema, the encryption process is individually done on color spaces components, so the color values in encrypted images are randomly distributed, also luminance information values are scattered among the resulted image.

7.2 The Proposed Image Cryptosystem

Here we present a novel cryptosystem for secure transmitted images. The proposed cryptosystem overcomes the drawbacks of existing algorithms. Many standard test tools are used to quantify the security level of the proposed cryptosystem, and experimental results prove that the suggested cryptosystem has a high security level, lower correlation coefficients, and improved entropy.

7.2.1 Design

As a new method of encryption, we use quasi groups' transposition and chaotic maps transposition on image space components level; we work with two image spaces namely RGB color space and YCbCr color space.

The steps involved in the proposed encryption/decryption process are given below.

Initialization Stage:

1. Getting the secret key
2. Build the quasi groups in the encryption process, and build the inverse quasi group in the decryption process.
3. Generate variables using chaotic maps

Encryption process:

Purpose: encrypting the image.

Inputs: the original image matrix.

Outputs: the encrypted image matrix.

Procedure:

Step 1: Initialize the required variables.

Step 2: Split the RGB image color components

i. Use the quasi group to shuffle the R component array.
ii. Use the quasi group to shuffle the G component array.
iii. Use the quasi group to shuffle the B component array.

Step 3: Reconstruct the image.

Step 4: Spelt image into Y, Cr, and Cb components

i. Apply the quasi group transposition to shuffle the Y component array.
ii. Apply the quasi group transposition to shuffle the Cr component array.
iii. Apply the quasi group transposition to shuffle the Cb component array.

Step 5: Restructure the image.

Step 6: Apply chaotic maps substitution on the image array to eliminate the image characteristics.

Decryption process:

Purpose: decrypting the image.

Inputs: the encrypted image matrix.

Outputs: the original image matrix.

Procedure:

Step 1: Initialize the required variables.

Step 2: Apply chaotic maps on the array, in order to inverse the substitution effects.

Step 3: Spelt image into Y, Cr, and Cb components

i. Use the inverse quasi group to reshuffle the Y component array.

ii. Use the inverse quasi group to reshuffle the Cr component array.

iii. Use the inverse quasi group to reshuffle the Cb component array.

Step 4: Reconstruct the image.

Step 5: Split the RGB image color components

i. Use the inverse quasi group to reshuffle the R component array.

ii. Use the inverse quasi group to reshuffle the G component array.

iii. Use the inverse quasi group to reshuffle the B component array.

*Step 6***:** Restructure the image.

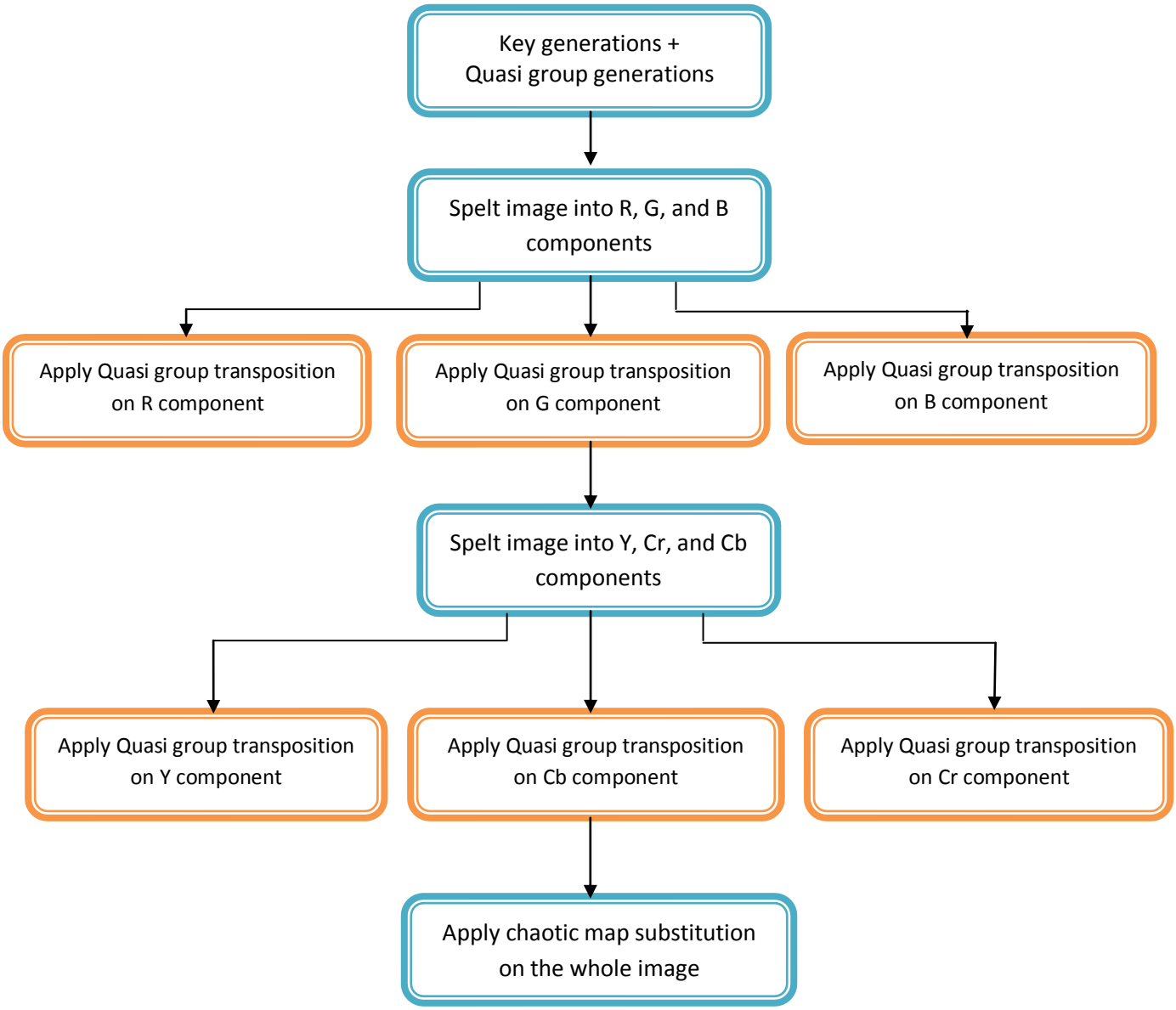


Fig. 7.1 The flowchart of the proposed image encryption process.

As shown in Fig. 7.1, the encryption process is divided into 4 main phases, first of all we initialize the quasi groups and generate the chaotic maps parameters from the

key, in second one we separate the RGB components of the image and apply a different quasi group transposition for each component, then the image is restructured in order to spelt the YCrCb components of the image and apply a different quasi group transposition for each component, finally, we reconstruct the image and perform the chaotic map substitution on it.

On the other hand, Fig. 7.2 shows the decryption process, also the first phase is initialization of parameters, but here we build the inverse quasi group, then we perform the chaotic map substitution on image, next we spelt image into YCrCb components to apply inverse quasi group transposition, after that the image is restructured in order to spelt the RGB components of the image and apply inverse quasi group transposition for each component, finally, we reconstruct the image.

The following schema is designed for colored images, while the following model is used with grayscale images. In grayscale encryption model, we use quasi groups' transposition and chaotic maps transposition on image level not on color space component level, which is the main difference between the two models.

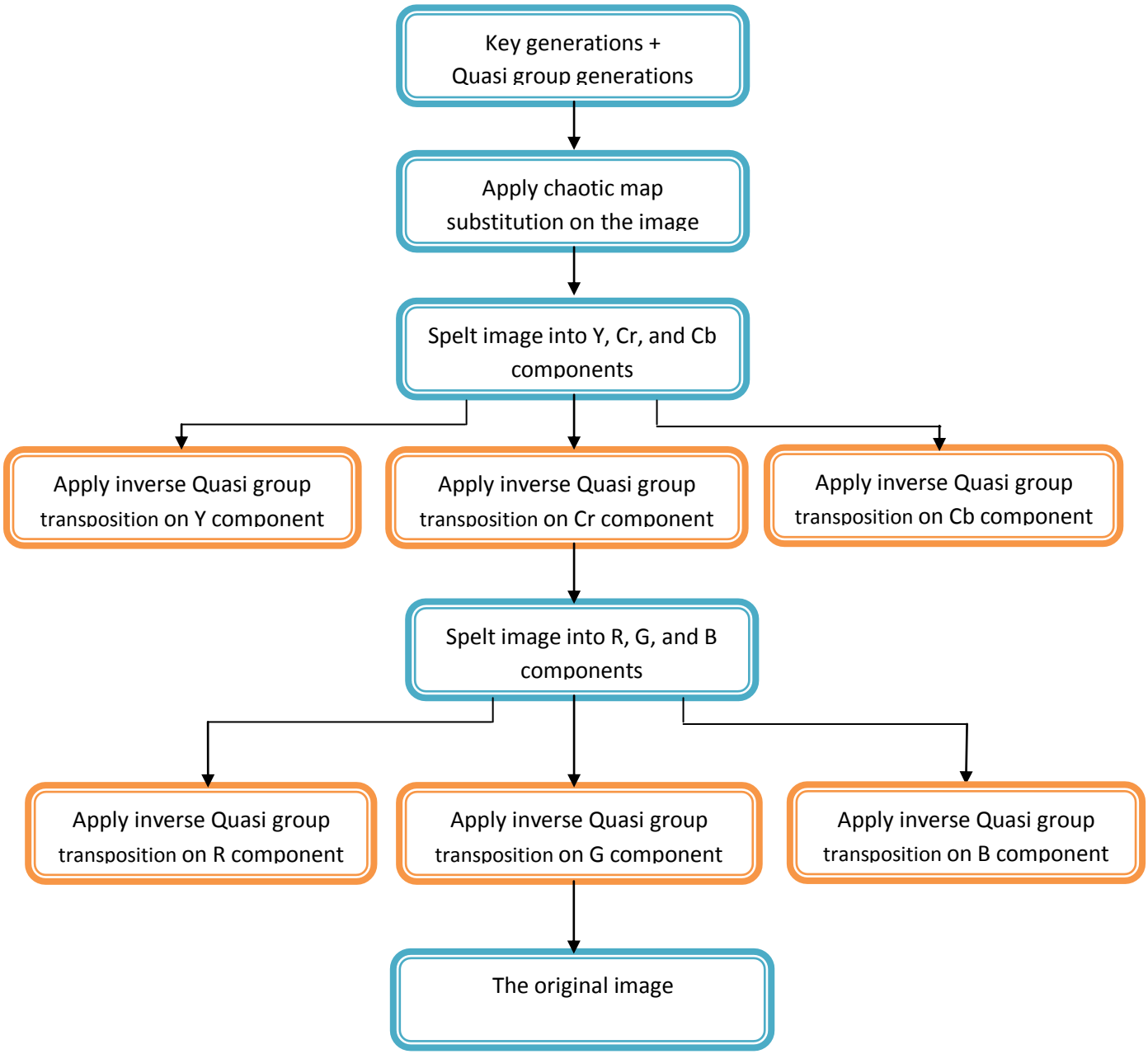


Fig. 7.2 The flowchart of the proposed image decryption process.

The following figures illustrate the encryption and decryption process for grayscale images.

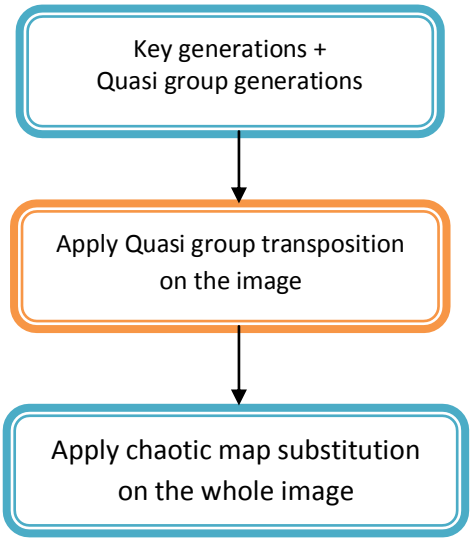


Fig. 7.3 The flowchart of encryption process for grayscale images.

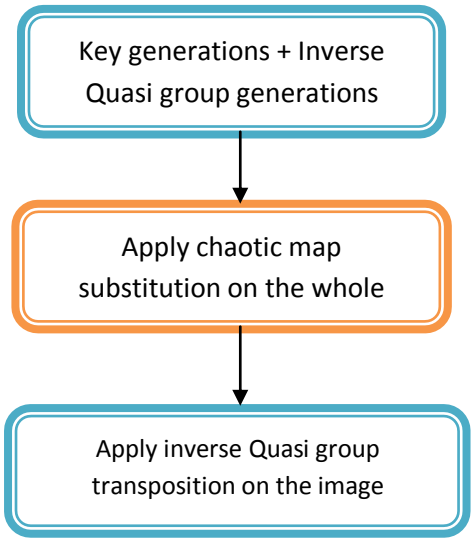


Fig. 7.4 The flowchart of decryption process for grayscale images.

The steps involved in the grayscale image encryption/decryption process are given below.

Initialization Stage:

1. Getting the secret key
2. Build the quasi groups in the encryption process, and build the inverse quasi group in the decryption process.

3. Generate variables using chaotic maps

Grayscale Encryption process:

Purpose: encrypting the image.

Inputs: the original image matrix.

Outputs: the encrypted image matrix.

Procedure:

Step 1: Initialize the required variables.

Step 2: Use the quasi group to shuffle the image array.

Step 3: Apply chaotic maps on the image array to eliminate the image characteristics

Grayscale Decryption process:

Purpose: decrypting the image.

Inputs: the encrypted image matrix.

Outputs: the original image matrix.

Procedure:

Step 1: Initialize the required variables.

Step 2: Apply chaotic maps on the array, in order to inverse the substitution effects.

Step 3: Use the inverse quasi group to reshuffle the image array.

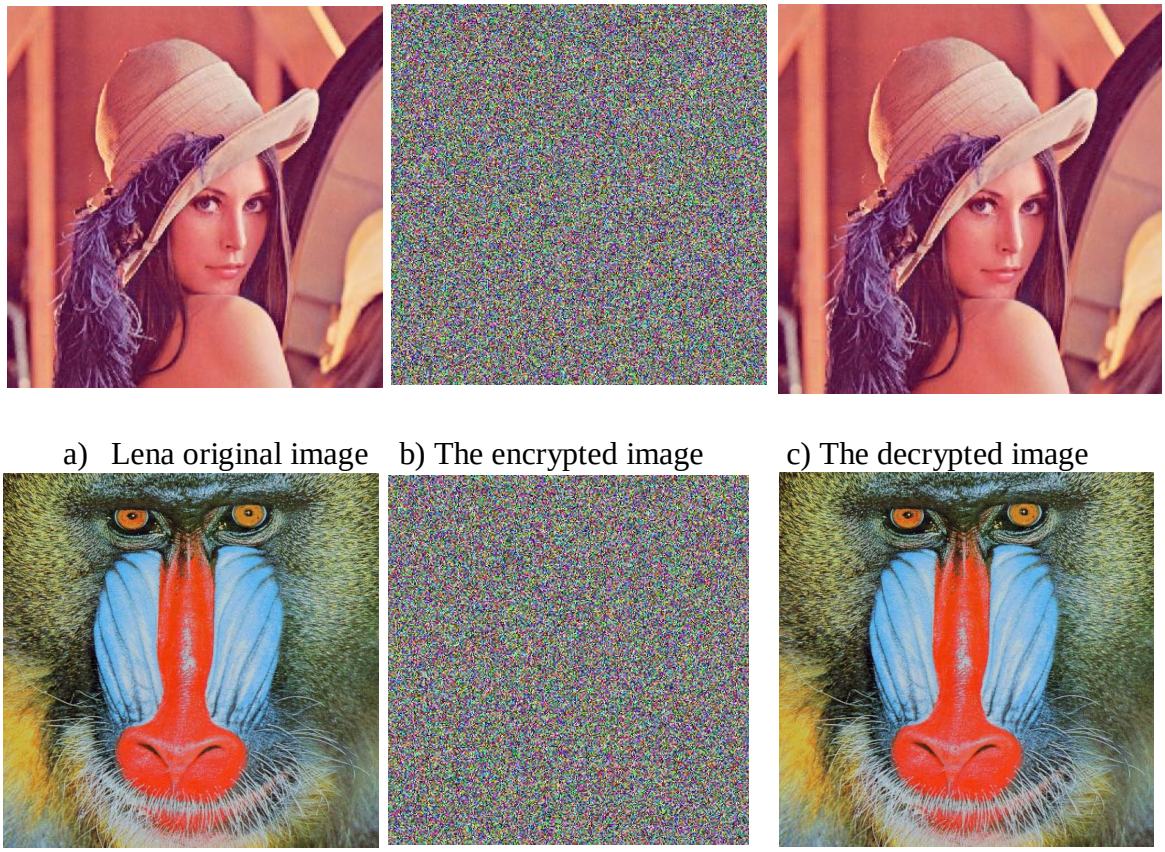
Chaotic maps present many desired cryptographic qualities such as simplicity of implementation that leads to high encryption rates, and excellent security. Their properties are similar to confusion and diffusion cryptography properties, so they have been used to build good cryptosystems. Furthermore, these properties make the chaotic cryptosystems robust against any statistical attack.

In addition to, the quasi group transformation is very effective in destroying the structure of the input signal and increases the entropy; therefore, it can be an excellent encryption technique.

As a result of working with image in the color space level, the colors distributed and new colors created and existed, moreover the colors distracted and redistributed in the encrypted image.

7.2.2 Implementation and Results

The proposed cryptosystem has been implemented using Matlab and the simulation results are observed on Core2 Duo, 2.40GHz with 2 GB RAM, The achieved results shown in the following figures.



a) Lena original image b) The encrypted image c) The decrypted image

d) Mandrill original image e) The encrypted image f) The decrypted image

Fig. 7.5 Application of proposed cryptosystem to Lena & Mandrill images.

The encrypted images are depicted in Figs. 7.5(b - e). As shown, the encrypted images are totally invisible. The decryption method takes as input the encrypted image, together with the same secret key1. The decrypted images are shown in Figs. 7.5(c - f). The visual inspection of Fig. 7.5 shows the possibility of applying the proposed cryptosystem successfully in both encryption and decryption. Also, it reveals its effectiveness in hiding the information contained in them.

In Fig. 7.5(b) there are green color pixels in the encrypted image, although the green color does not appear in the original image. This happened due to working in color components levels.

7.3 Security Analysis

Some experimental results are given in this section to demonstrate the efficiency of our scheme. The plain colored images of Lena and Mandrill in 512×512×3 format are used with various tests like visual testing, sensitivity analysis, and key space analysis, etc., to prove the effectiveness of the algorithm.

7.3.1 The Correlation Coefficients between Plain and Encrypted Images

The numerical measure of the correlation between two pixels of two images (plain image and cipher image) i.e., correlation coefficient (ρ) among the pixels are calculated by using the following formula

$$\rho = \frac{\sum(f-\bar{f})(\hat{f}-\bar{\hat{f}})}{\sqrt{\sum(f-\bar{f})^2 \sum(\hat{f}-\bar{\hat{f}})^2}} \qquad (7.1)$$

Where f and $\hat{f}$ are the pixel values of the original and encrypted images at the position (i, j) respectively [3].

Table 7.1: The correlation coefficient between original and encrypted image

Between	Lena	Mandrill
Plain & encrypted using key1	0.0022	0.0025
Plain & encrypted using key2	0.0005	0.0001

Table 7.1 shows the calculated values of the correlation coefficients between the plain image and the encrypted image using key1 and key2 correspondingly. The values of the correlation coefficients make it clear that there is no statistical resemblance between the plain image and the encrypted image and between two encrypted images produced by using two slightly different keys. The fact establishes the impossibility of statistical attack and robustness of the scheme.

The similar result obtained when we use gray grayscale image encryption model the correlation coefficients between the plain image and the encrypted image = 0.0021.

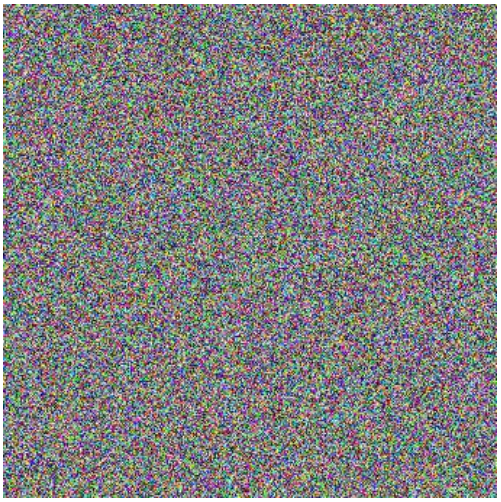
7.3.2 Histograms of Original and Encrypted Image

The histogram provides a compact summarization of the distribution of data in an image, it is possible to use the process of histogram matching and mismatching between two images for analyzing the security level of a cryptographic technique.
In this context, we would compute the histograms of two images plaintext and ciphertext to consider the security level based on the proposed algorithm. Image histogram describes how the image pixels are distributed by plotting the number of pixels (along y-axis) at each intensity level (along x-axis) [64].

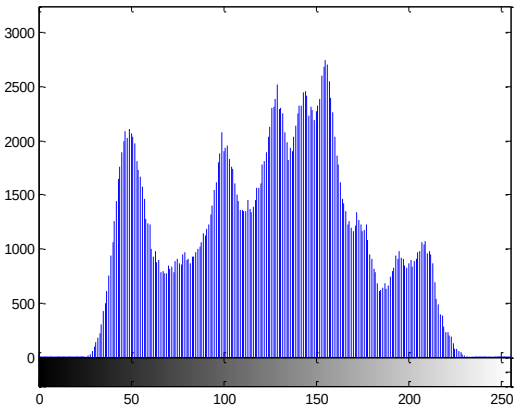
As noticed in Figs. 7.6 and 7.7 (c), the gray level histogram of a plain images contains large spikes. These spikes correspond to color values that appear more often in the plain image. The histogram of the cipher images as shown in Fig. 7.6 and 7.7 (d), is more regular, significantly different from that of the original image, and bears no statistical resemblance to the plain image. Furthermore, the colored histogram of original images in Figs. 7.6 and 7.7 (e) shows exactly the color distribution is high for particular colors, while the colored histogram of encrypted images as demonstrated in Figs. 7.6 and 7.7 (f), is more uniform therefore encrypted image bear no statistical resemblance to the plane image and hence do not provide any clue to employ any statistical attack on the proposed technique. In Fig. 7.7 (c) some values in the ranges [0-80] and [180-250] are missing, but they exist in the encrypted image histogram Figure 7.7(d).

It is clear that the histogram (gray level and colored) of the encrypted images is fairly uniform and significantly different from the respective histograms of the original images and hence does not provide any clue to employ any statistical attack on the proposed image encryption process.

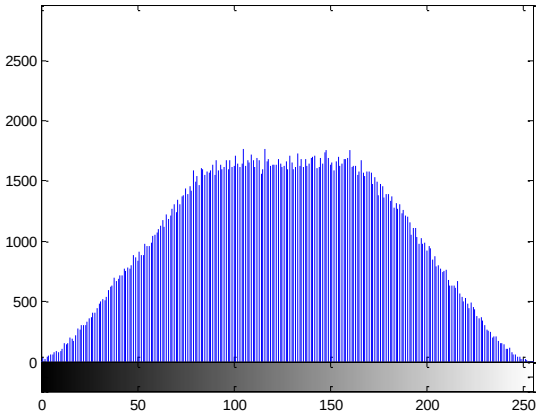
a) original image



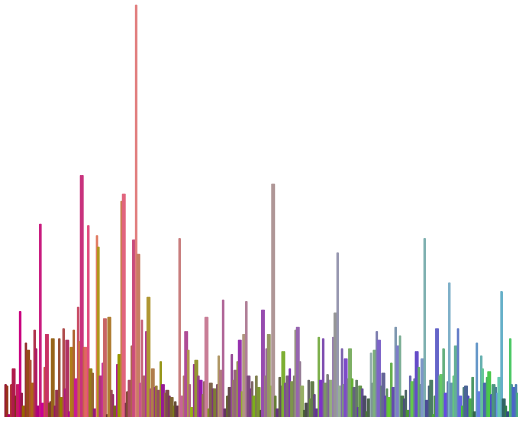
b) Encrypted image



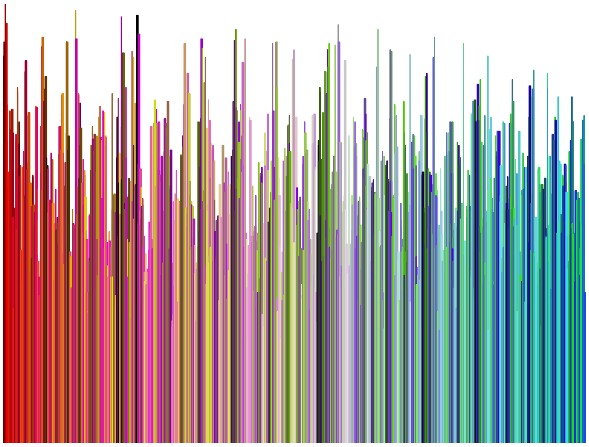
c) Gray level histogram of original image



d) Gray level histogram of encrypted image

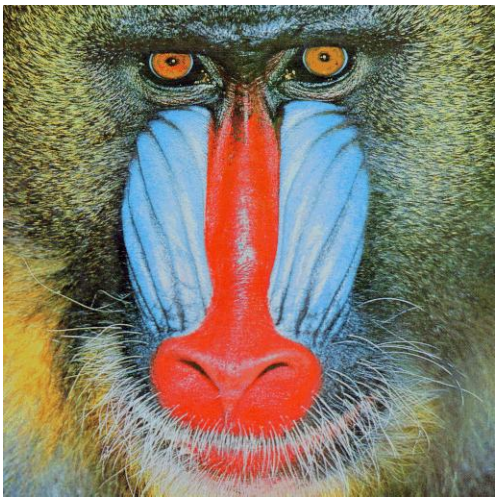


e) Colored histogram of original image

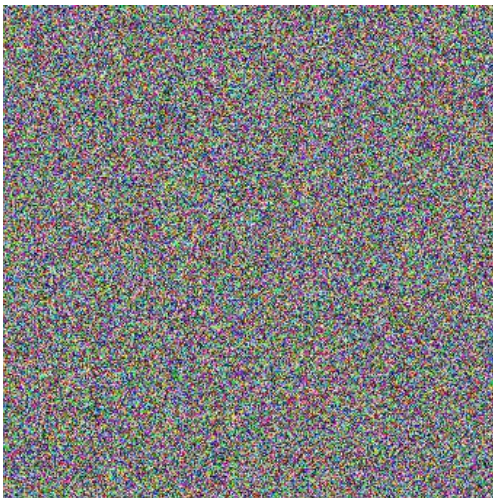


f) Colored histogram of encrypted image

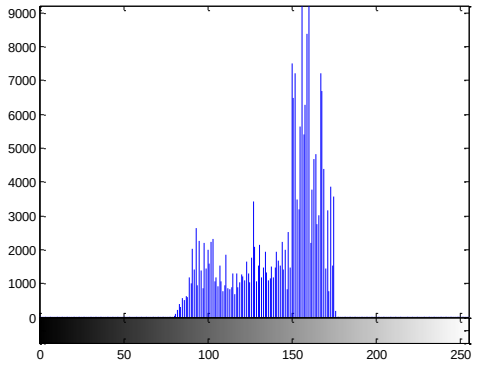
Fig. 7.6 Lena original & encrypted images, the gray level histograms of images and the colored histogram of images



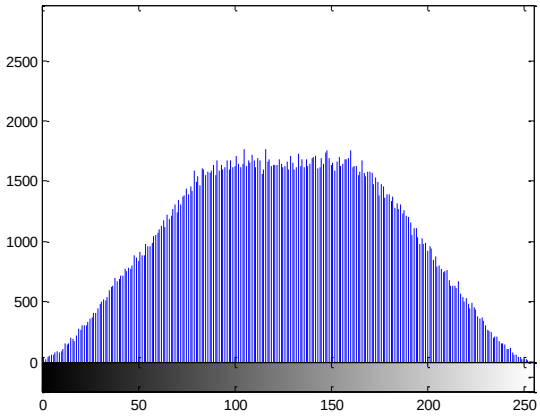
a) original image



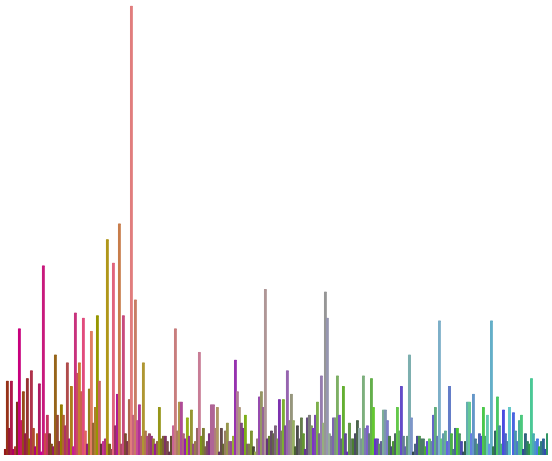
b) Encrypted image



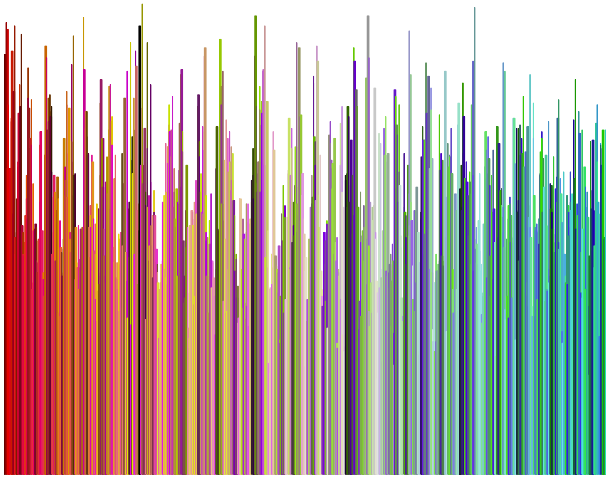
c) Gray level histogram of original image



d) Gray level histogram of encrypted image



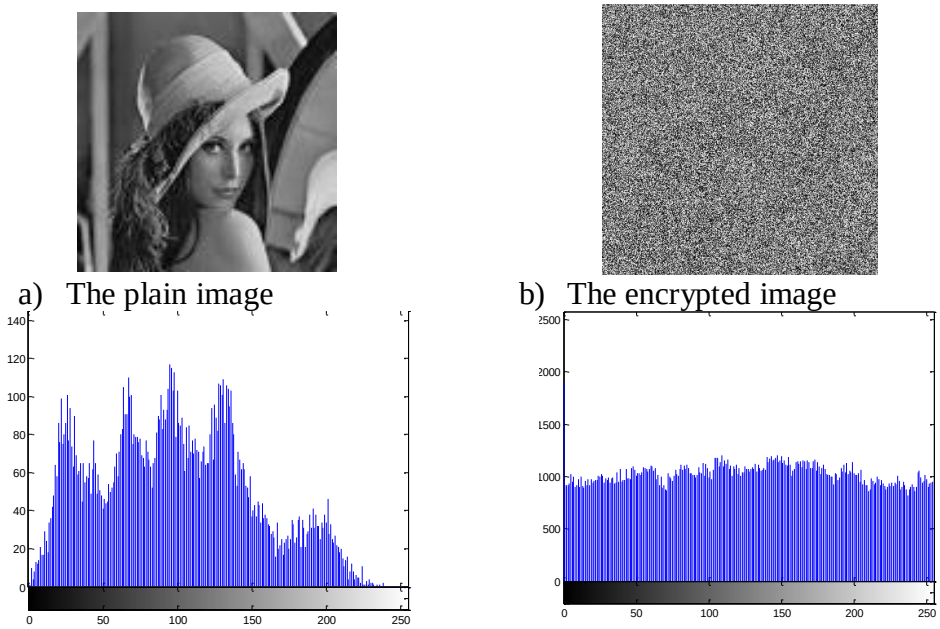
e) Colored histogram of original image



f) Colored histogram of encrypted image

Fig. 7.7 Mandrill original & encrypted images, the gray level histograms of images and the colored histogram of images

For grayscale encryption model, the following figures show the histogram of plain image and encrypted image, the plain image is Lena grayscale image 124 X 124 pixels, The histogram of the cipher image as shown in Fig. 7.8(d), is more uniform, significantly different from that of the original image that the proposed algorithm has a good ability of diffusion and confusion as well as resisting the statistical analysis attack.



a) The plain image b) The encrypted image

c) Histogram of plain image d) Histogram of encrypted image

Fig. 7.8 Lena original & encrypted images and the gray level histograms of images.

7.3.3 The Correlation Coefficient of the Adjacent Pixels. (*Distribution of Two Adjacent Pixels*)

In this section, some simulations are carried out to test the correlation distribution between two horizontally, vertically and diagonally adjacent pixels, in the original and encrypted images. To quantify the dependence between two images, Pearson's correlation coefficient is commonly used. Given by (7.5), this coefficient is obtained by dividing the covariance between the two images (7.4) by the product of their standard deviations as in (7.2) and (7.3). E in (7.2) is the expected value operator. $P_1\ (i, j)$ and $C_1\ (i, j)$ are pixels gray values of the original and the encrypted images, respectively [65].

$$E(x) = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} P_1\ (i, j) \qquad (7.2)$$

$$D(P_1) = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} [P_1(i,j) - E(P_1(i,j))]^2 \quad (7.3)$$

$$cov(P_1, C_1) = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} [P_1(i,j) - E(P_1(i,j))]\,[C_1(i,j) - E(C_1(i,j))] \quad (7.4)$$

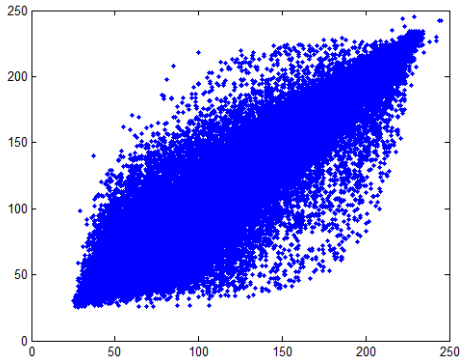
$$r_{P_1 C_1} = \frac{cov(P_1, C_1)}{\sqrt{D(P_1)}\sqrt{D(C_1)}} \quad (7.5)$$

Table 7.2: The Correlation coefficients of adjacent pixels for Lena image.

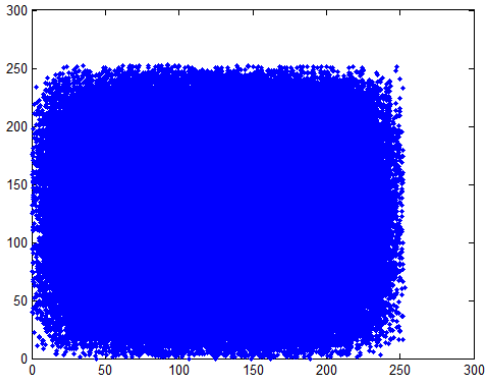
Direction of adjacent pixels	Plain image	Ciphered image
Horizontal (H)	0.9719	0.0032
Vertical (V)	0.9850	0.0012
Diagonal (D)	0.9593	0.0005
$\sqrt{\frac{1}{3}(H^2 + V^2 + D^2)}$	0.9721	0.0019
Average (H, V, D)	0.9720	0.0016

Table 7.3: Comparison of correlation coefficient for the ciphered Lena image of the proposed algorithm with the other techniques.

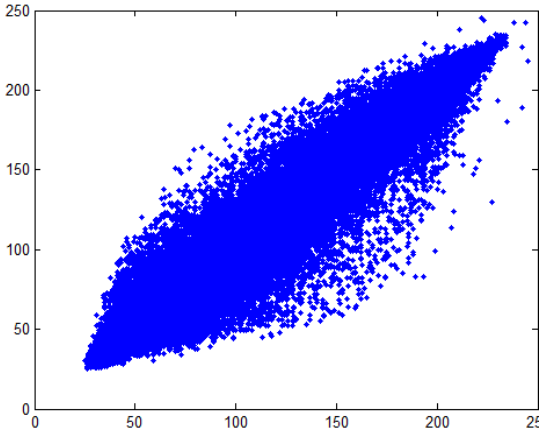
Direction of adjacent pixels	Fei t.al [66]	Zhang et.al.[67]	Gao et. al.[68]	Kumar et. al. [69]	Mandal et. al. [64]	Proposed schema
Horizontal (*H*)	0.0361	0.0820	0.0160	0.0356	0.0166	**0.0032**
Vertical (*V*)	0.0022	0.0400	0.0650	0.0335	0.0411	**0.0012**
Diagonal (*D*)	0.0324	0.0050	0.0320	0.0335	0.0021	**0.0005**
$\sqrt{\frac{1}{3}(H^2 + V^2 + D^2)}$	0.0278	0.0527	0.4283	0.1850	0.0256	**0.0019**
Average (*H, V, D*)	0.0235	0.0423	0.0376	0.0342	0.0199	**0.0016**



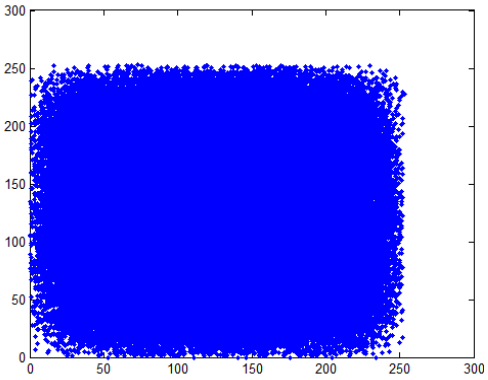
a) horizontally adjacent pixels Correlation in plain image



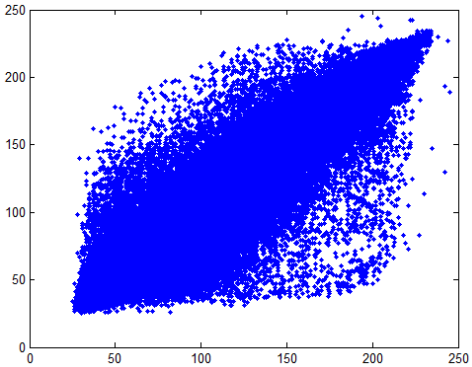
b) horizontally adjacent pixels Correlation in cipher image



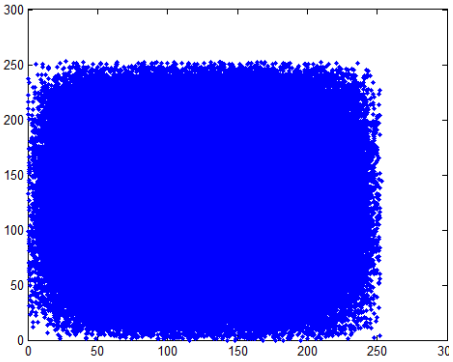
c) Vertical adjacent pixels Correlation in plain image



d) Vertical adjacent pixels Correlation in cipher image



e) Diagonal adjacent pixels Correlation in Plain image



f) Diagonal adjacent pixels Correlation in cipher image

Fig. 7.9 Scatter diagrams of the (a) plain image and the (b) ciphered image in horizontal direction of adjacent pixels, scatter diagrams of the (c) plain image and the (d) ciphered image in vertical direction of adjacent pixels and scatter diagrams of the (e) plain image and the (f) ciphered image in diagonal direction of adjacent pixels

The calculated values of the correlation coefficients between the adjacent pixels in the horizontal, vertical and diagonal directions are shown in Table 7.2 for the case of plain image and ciphered image. A comparison of correlation coefficient of the ciphered image based on the proposed algorithm and the other techniques available in the literature are summarized in Table 7.3. According to the Table 7.3 all values of the correlation coefficient of the proposed algorithm are less than the all other schemes.

Fig. 7.9 shows the correlation distribution of two horizontally adjacent pixels in the plain image and that in the cipher image are far apart. Similar results for diagonal and vertical directions were obtained, so the strong correlation between the adjacent pixels in the original image is reduced in the encrypted image.

Table 7.4: The Correlation coefficients of adjacent pixels for Lena grayscale image.

Direction of adjacent pixels	Plain image	Ciphered image
Horizontal (*H*)	0.8531	-0.0008
Vertical (*V*)	0.9298	0.0080
Diagonal (*D*)	0.8171	-0.0036
Average (*H*, *V*, *D*)	0.8667	0.0012

The previous table demonstrates how the correlation between the adjacent pixels reduces dramatically the correlation between the adjacent pixels.

7.3.4 Intra-Components Correlation Coefficients of Plain and Encrypted Images

Table 7.5 Mean values of correlation coefficients of intra-component of original and encrypted images.

	Lena image		Mandrill image	
	original	encrypted	original	encrypted
Red (R) component correlation	0.0642	0.0019	0.1683	0.0019
Green (G) component correlation	0.0426	0.0018	0.0742	0.0021
Blue (B) component Correlation	0.0360	0.0018	0.0830	0.0019
Mean	0.0476	0.0018	0.1085	0.002

7.3.5 Inter-Components Correlation Coefficients of Plain and Encrypted Images

Table 7.6 Mean values of correlation coefficients of inter-component of original and encrypted images.

	Lena image		Mandrill image	
	original	encrypted	original	encrypted
Correlation between R and G	0.8786	-0.0023	0.3481	0.0028
Correlation between G and B	0.9106	0.0021	0.8273	-0.0003
Correlation between B and R	0.6764	-0.0022	0.1258	-0.0030
Mean	0.8219	-0.0008	0.4337	-0.0002

Tables 7.5 and 7.6 give the correlation coefficients of intra-components and inter-components of original and encrypted images. It can be seen that the use of chaotic maps reduces significantly the intra-component correlation coefficients which are already low.

7.3.6 Information Entropy Analysis

Entropy is a statistical measure of randomness that can be used to characterize the texture of an image. It is well known that the entropy *H(m)* of an image *m* can be calculated as :

$$H(m) = \sum_{i=0}^{2^N-1} p(m_i) log_2 \frac{1}{p(m_i)} \quad (7.6)$$

Where $p(m_i)$ represents the probability of message m_i. When an image is encrypted, its entropy should ideally be 8. If it is less than this value, there exists a certain degree of predictability which threatens its security [65].

Table 7.7 The entropy values of original and encrypted images.

	Original image	encrypted
Lena image	7.45	7.99
Mandrill image	7.12	7.98
Lena grayscale image	7.59	**7.99**

The obtained results are very close to the theoretical value. This means that information leakage in the encryption process is negligible, and the encryption system is secure against the entropy attack.

7.3.7 Analysis of Anti-Differential Attack

In general, the opponent may make a slight change (e.g., modify only one pixel) of the encrypted image, and then observes the change of the result. In this way, he may be able to find out a meaningful relationship between the plain image and the cipher image. If one minor change in the plain image can cause a significant change in the cipher image, with respect to diffusion and confusion, then this differential attack would become very inefficient and practically useless. To test the influence of one-pixel change on the whole image encrypted by the proposed algorithm, a common measure is used: number of pixels change rate (NPCR) to test the difference between the original image P_1 and the permuted one C_1.

NPCR stands for the number of pixel change rate Eq. (7.8). Then, if *D* is a matrix with the same size as images P_1 and C_1, *D (i, j)* is determined as follows:

$$D(i,j)=\begin{cases}1 \ if \ P_1(i,j) \neq C_1(i,j)\\ 0 \ else\end{cases} \quad (7.7)$$

NPCR is defined by the following formula:

$$NPCR = \frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} D(i,j)}{M \times N} \times 100\% \quad (7.8)$$

where *M* and *N* are the width and height of P_1 and C_1.

NPCR Between original & encrypted images = 99.9302 %, while NPCR between encrypted image1 & encrypted image2 (where plain image 2 is slightly different from plain image 1) = 99.4743 % which showing thereby that the encryption scheme is very sensitive with respect to small changes in the plain image. Furthermore the NPCR Between original & encrypted images using grayscale encryption model = 99.6510 %,

7.4 Sensitivity analysis

An ideal image encryption procedure should be sensitive with respect to both the secret key and plain image. The change of a single bit in either the secret key or plain image should produce a completely different encrypted image. To prove the robustness

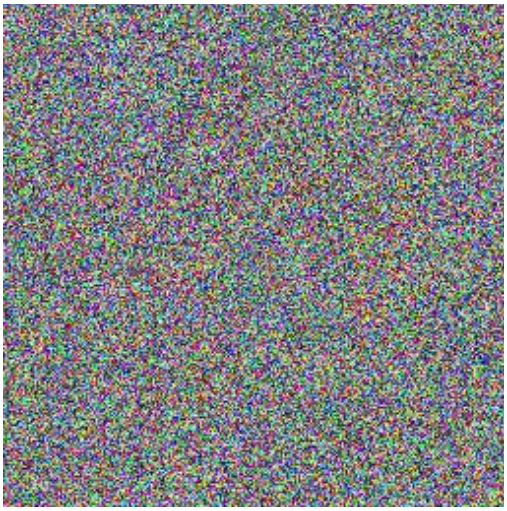
of the proposed cryptosystem, we will perform sensitivity analysis with respect to both key and plain image.

7.4.1 Key Sensitivity Analysis

High key sensitivity is required by secure image cryptosystems, which means that the cipher image cannot be decrypted correctly although there is only a slight difference between encryption or decryption keys. This guarantees the security of the proposed cryptosystem against brute force attack.

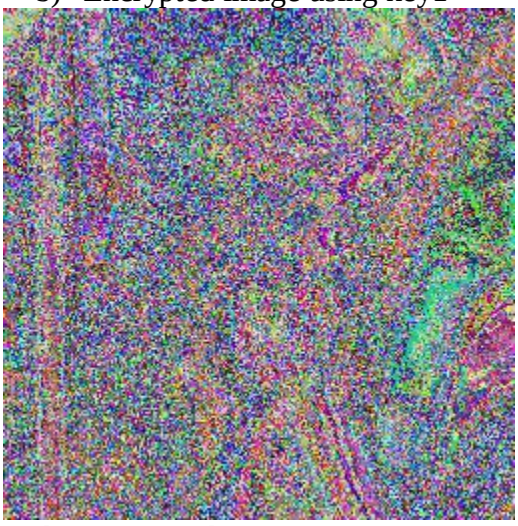
In the following figures we test the key sensitivity with four different images, whereas we try to decrypt the cipher image using slightly different key.

a) Plain Lena image



b) Encrypted image using key1

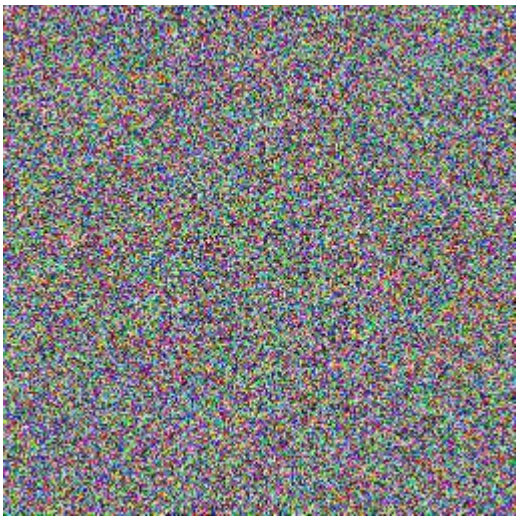
c) Correctly decrypted image using key1



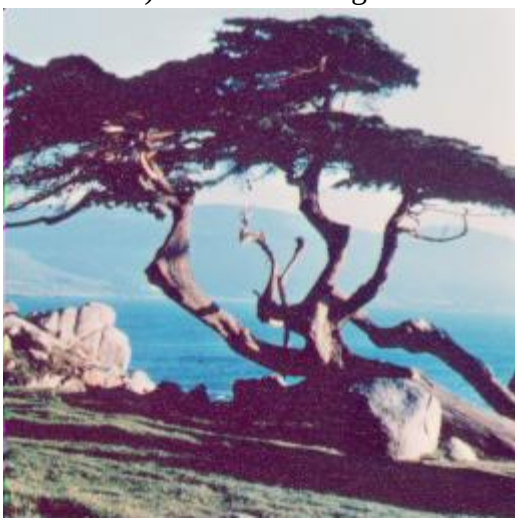
d) Decrypted image using wrong key2

Fig. 7.10 Key sensitive test result with Lena image

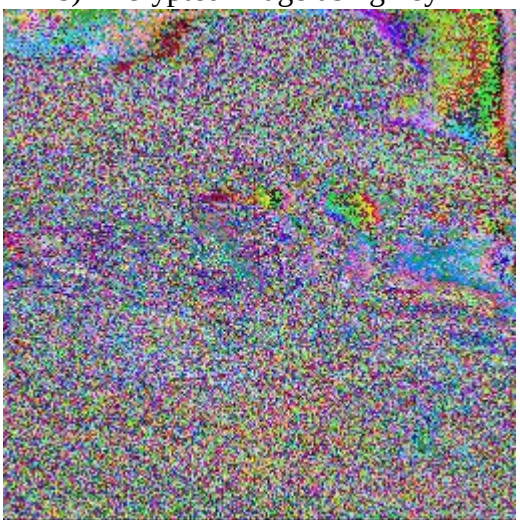
a) Plain Tree image



b) Encrypted image using key1



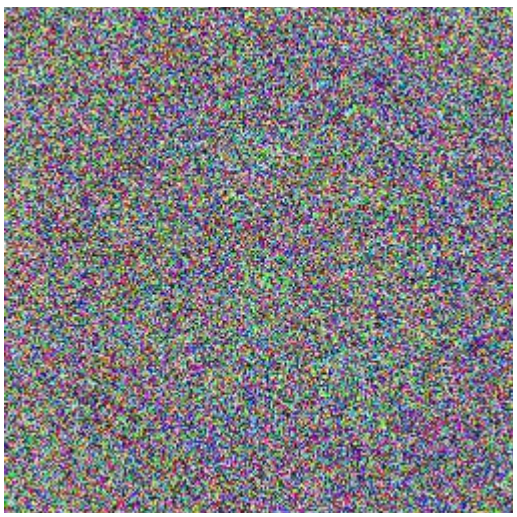
c) Correctly decrypted image using key1



d) Decrypted image using wrong key2

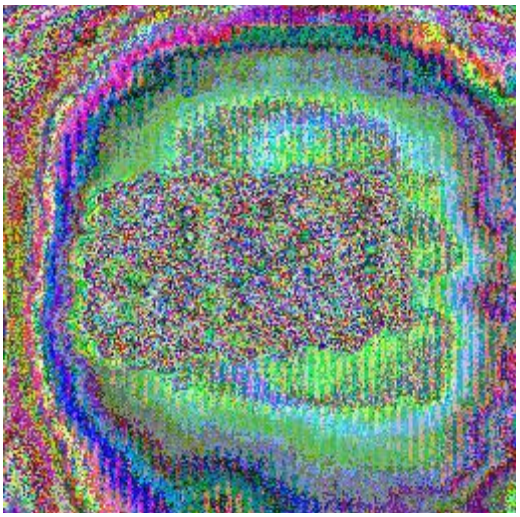
Fig. 7.11 Key sensitive test result with Tree image

a) Plain Jelly beans image



b) Encrypted image using key1

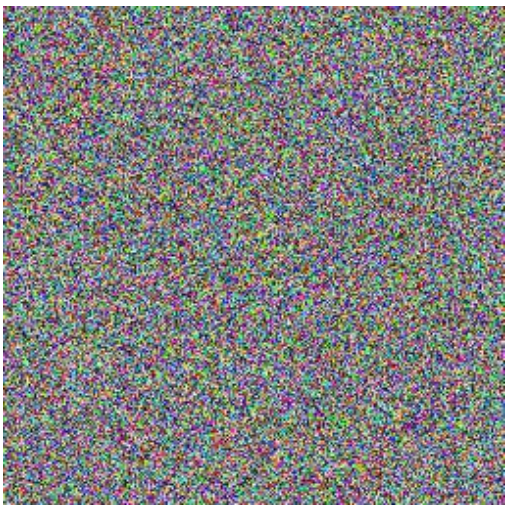
c) Correctly decrypted image using key1



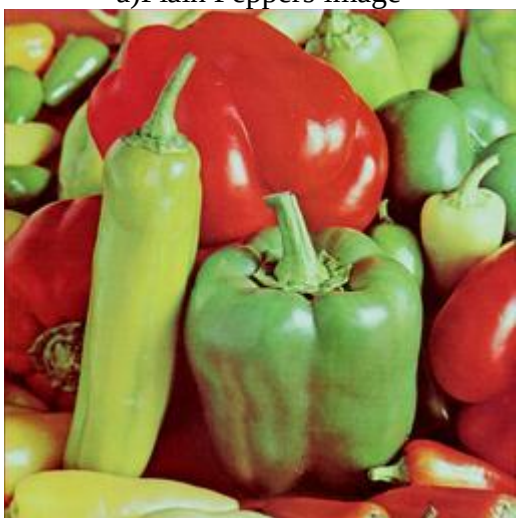
d) Decrypted image using wrong key2

Fig. 7.12 Key sensitive test result with Jelly beans image

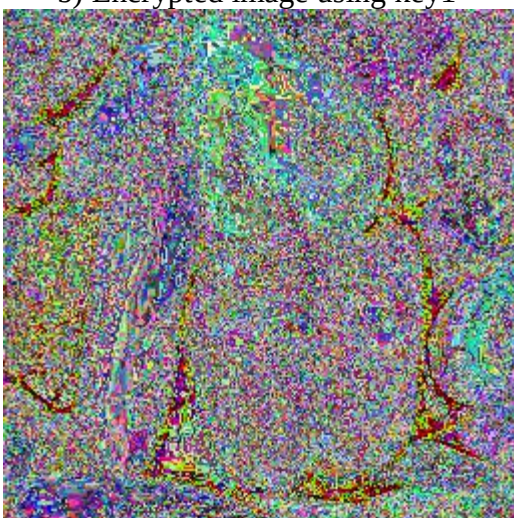
a)Plain Peppers image



b) Encrypted image using key1



c) Correctly decrypted image using key1



d) Decrypted image using wrong key2

Fig. 7.13 Key sensitive test result with Peppers image

It is clear from the Figs. 7.10 – 7.13 that the decryption with a slightly different key fails completely and hence the proposed image encryption procedure is highly key sensitive.

Table 7.8 shows the difference between the two ciphered images of Lena. Similar results are obtained for Mandrill image. As we can see here, our algorithm is quite sensitive to the key. The two obtained encrypted images are very different and resemble random data. Because it is very sensitive to the key, it is secure against the brute force attack.

Table 7.8 Correlation and NPCR between two ciphered images encrypted with slightly different keys.

	Correlation	NPCR
Encrypted key1 & Encrypted key2	**0.0030**	**99.5007**

7.4.2 Plain image sensitivity analysis

For the test of sensitivity on small plain image changes, we used two plain images of Lena different by only one bit. The obtained encrypted images are identical only in 0.05%. This result demonstrates that the cipher is sensitive to small changes in the original image. Then, we can conclude that the algorithm resists the plaintext attack and differential attack.

Table 7.9 Correlation and NPCR between two ciphered slightly different images.

	Correlation	NPCR
Encrypted key1 & Encrypted key2	**0.0015**	**99.4743**

It is shown that we have obtained a low correlation between the original and the encrypted images. The NPCR is high enough to confirm that the two images are very different.

7.5 Key Space Analysis

For a secure image cryptosystem, the key space should be large enough to make the brute force attack infeasible. Key spaces imply the total number of different keys which can be used for the purpose of encryption and decryption. With respect to the

speed of today's computers, the key space should be more than $2^{100} = 10^{30}$ in order to avoid brute force attacks [70]. The proposed cryptosystem uses a 256 bits key, so that the key space is $2^{256} - 1$. As a result, the cryptosystem is secure against brute force attack.

The proposed cryptosystem has 2^{256} different combinations of the secret key. An image cipher with such a long key space is sufficient for reliable practical use. In the proposed cryptosystem, a chaotic map is employed which is sensitive on the initial condition. The initial condition for logistic map is calculated from the secret key.

7.6 Visual Testing

A number of colored and grayscale images are encrypted by the proposed schemas, and visual test is performed. Numerous examples of encryption are shown in Figs. 7.5, 7.8, 7.10 – 7.13, with different images size (256x256 and 512x512) pixels. By comparing the original and the encrypted images we notice that there is no visual information observed in the encrypted image, and the encrypted images are visual indistinguishable. Consequently, this proposed cryptosystem succeeded in encrypting images of different sizes.

7.7 Summary

In this chapter, we introduce a cryptosystem based on the combination of chaotic cryptography and quasi group cryptography. The original image was detached into its RGB color components, which were rearranged using a quasi group transposition, and then we reconstructed the image to convert it to YCrCb color space, in order to detach it into its components, again a quasi group transposition was used to shuffle the YCrCb color space components, after that the image was restructured and perform the chaotic map transposition on it, to destroy the image characteristics.

This new scheme employs the quasi group transposition to shuffle the positions of image pixels, this transposition is done in both RGB color space components and YCrCb color space components, i.e. the red, green, blue, luminance component (Y), and chrominance information is stored as two color-difference components (Cb and Cr) component, each of them is handled individually by mixed up its cells using quasi group transposition. Consequently the colors distributed and new colors created and existed, moreover the colors distracted and redistributed in the encrypted image.

The chaotic map was used to confuse the relationship between the cipher image and the plain image, thereby significantly increasing the resistance to statistical and differential attacks

The results showed that the correlation between image elements was significantly decreased by using the proposed technique. The results also exhibit that high key sensitivity which guarantees the security of the proposed cryptosystem against brute-force attack. Moreover the results demonstrate that the cipher is sensitive to small changes in the original image, which conclude that the cryptosystem resists the plaintext attack and differential attack.

Since the entropy is found to be close to the theoretical value, we can observe that the information leakage is negligible and hence the scheme is highly secure against the entropy attack.

The histogram of the ciphered image is significantly different from that of the original image, hence does not provide any clue to employ any statistical attack on the proposed cryptosystem.

The common measures like NPCR, intra and inter-components correlation coefficients, and distribution of two adjacent pixels applied to images, prove the robustness and the high security level of the proposed cryptosystem.

The high difference between the original and the encrypted images and the randomness of the encrypted images prove that the cryptosystem is secure against the ciphertext only attack.

The almost uniform histograms of the encrypted image, non-uniform histograms of the plain image, scatter diagrams and the correlation coefficients (Table 7.2 and Table 7.3) established that the proposed cryptosystem has a good ability of diffusion and confusion as well as resisting the statistical analysis attack.

The proposed scheme uses a key of 256 bits and therefore, the key space is approximately $2^{256} \approx 10^{77}$, which is large enough to protect the cryptosystem against brute force attack.

It's clear from the measurements results and visual inspection, that the encrypted image contains new colors (which not existed in the original image), besides the new distribution of colors which appears in the encrypted image.

Chapter 8

The Hybrid Cryptosystem

Modern cryptosystems are usually designed to take advantages of both private key and public key cryptosystems. Public key cryptosystems are used to distribute symmetric keys at the start of a session. Once a secret key is known to all parties of the session, faster private key cryptosystems using that key can be used to encrypt the remainder of the session.

For private key cryptosystem, a new schema has been proposed; it is composed of two cryptographic schemas, namely quasi group cryptography and chaotic cryptography. To achieve the proposed system, the work is divided into phases as shown below.

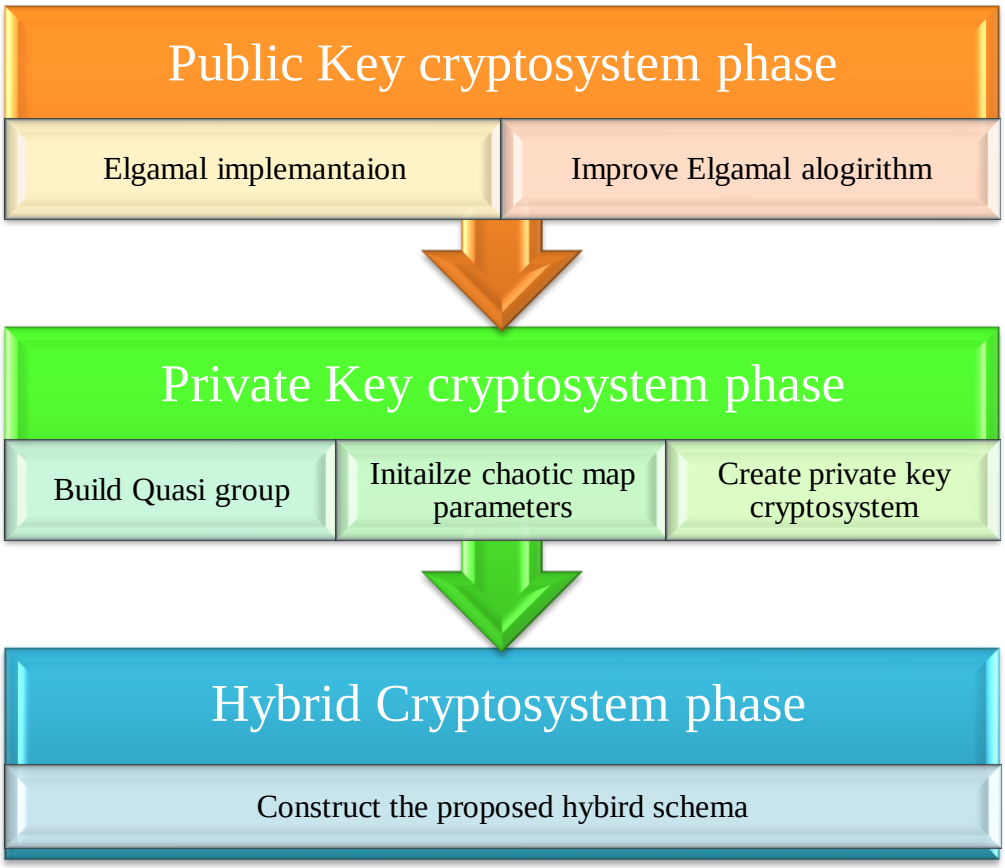


Fig. 8.1: A general block diagram of the proposed hybrid cryptosystem.

In the thesis, a hybrid cryptosystem is proposed. The new cryptosystem utilizes both private and public key cryptosystems, takes advantage of the virtues of both.

8.1 Design

The first phase which is a public key cryptosystem phase. In this phase I implement Elgamal cryptosystem, and improve it to eliminate its private shortcoming.

While the second phase is a private key cryptosystem phase, I design and implement a new cryptosystem that involve a mathematical theories (chaotic systems and quasi group) in the encryption process to get the highest possible protection of encrypted data, and get a high degree of randomization, due to their properties. The private key cryptosystem is divided into three models based on data type; the first mode deals with row data (especially text data), the second model is customized for voice waves and the last is specified for images encryption.

The last phase is the hybrid cryptosystem phase; here I will make a combination between the resulted systems of previous phases to obtain the proposed schema that utilizes both private and public key cryptosystems.

The following figure illustrates the hybrid cryptosystem, showing the three main phases in details.

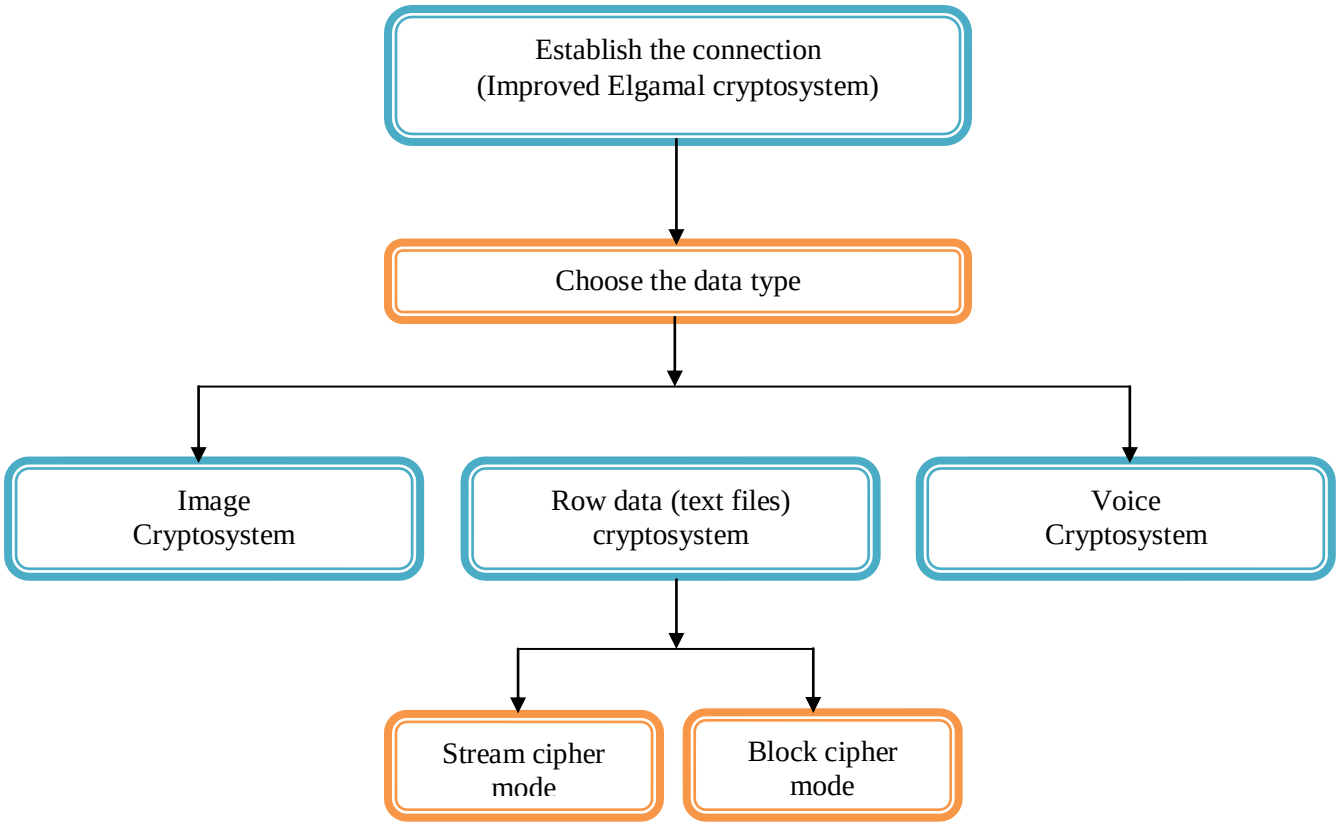


Fig. 8.2: The proposed hybrid cryptosystem.

The improved Elgamal cryptosystem is described in chapter 4, while the private key cryptosystem of row data is shown in chapter 5, whereas the voice cryptosystem is explored in chapter 6 and the image cryptosystem is demonstrated in chapter 7.

8.2 Implementation and Results

As the previous the work, the proposed hybrid cryptosystem has been implemented using Matlab and the simulation results were observed on Core2 Duo, 2.40GHz with 2 GB RAM.

The user interface in my program gives you many choices as shown in Fig. 8.3; firstly, you need to connect the receiver as illustrated in Fig. 8.4, the connection starts the improved Elgamal cryptosystem, which implicitly means the secret key is transfer through insecure channel.

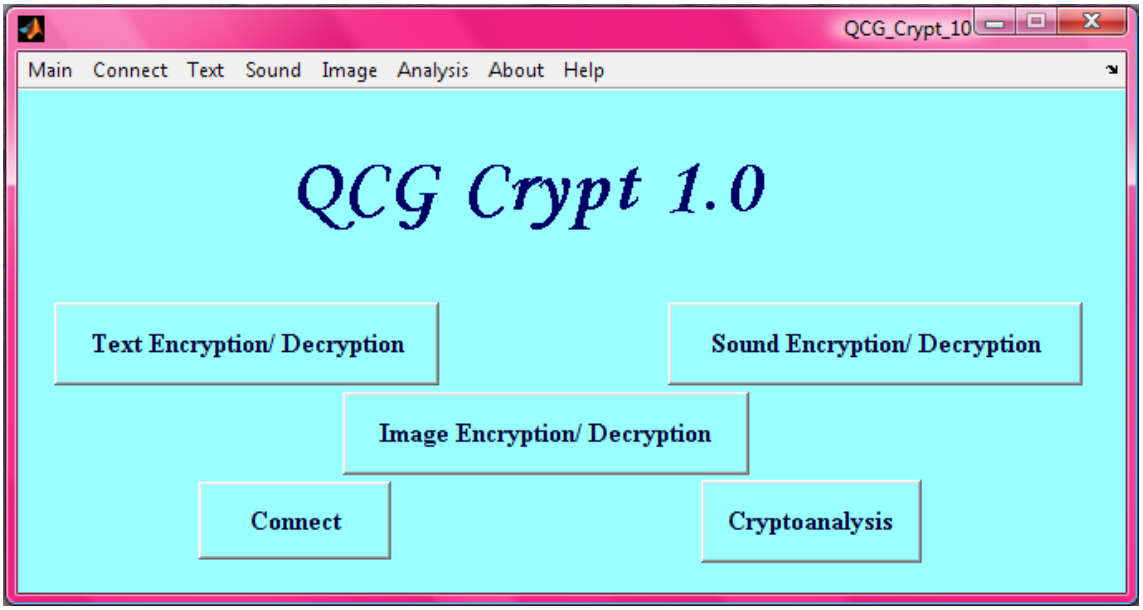


Fig. 8.3: The main user interface

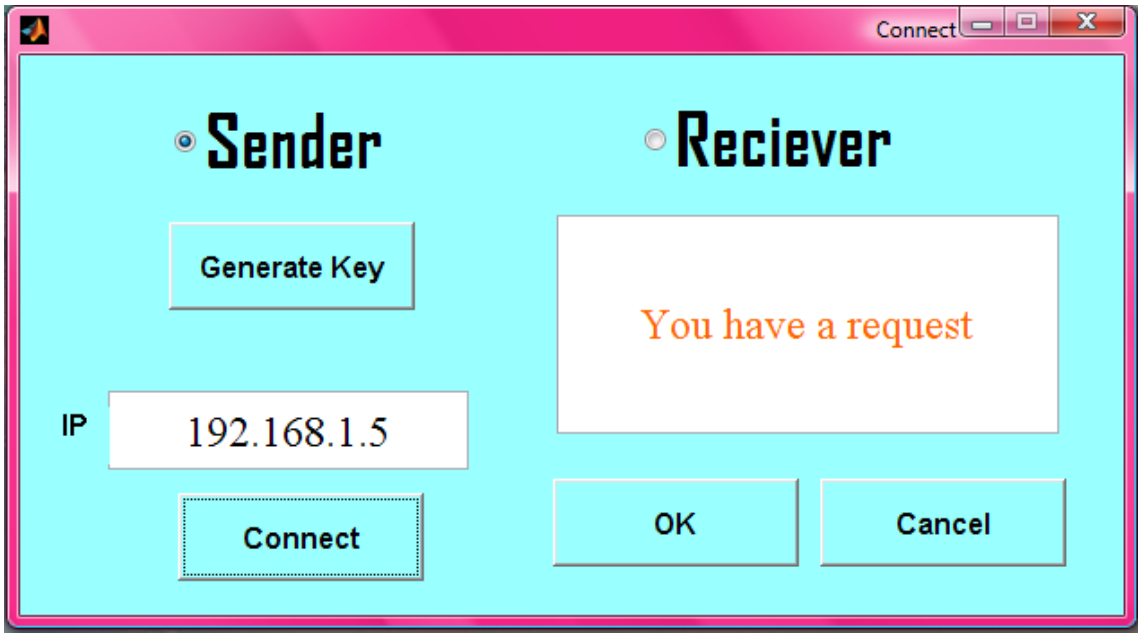


Fig. 8.4: Connect window.

Now you can select the data type and encrypt it in order to send it, the following figures show the user interface for each data type. Fig. 8.5, shows the user interface which used to encrypt/ decrypt text.

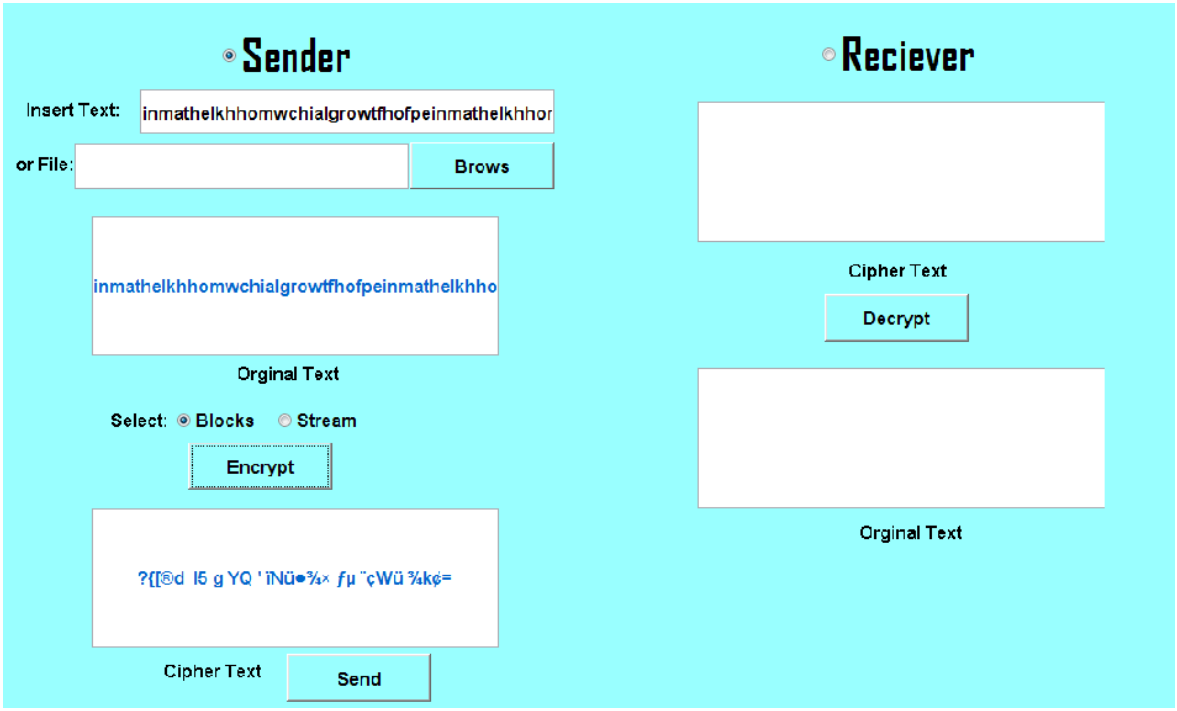


Fig. 8.5: Text encryption / decryption window.

The following figure contains the interface that used to encrypt / decrypt sound signals.

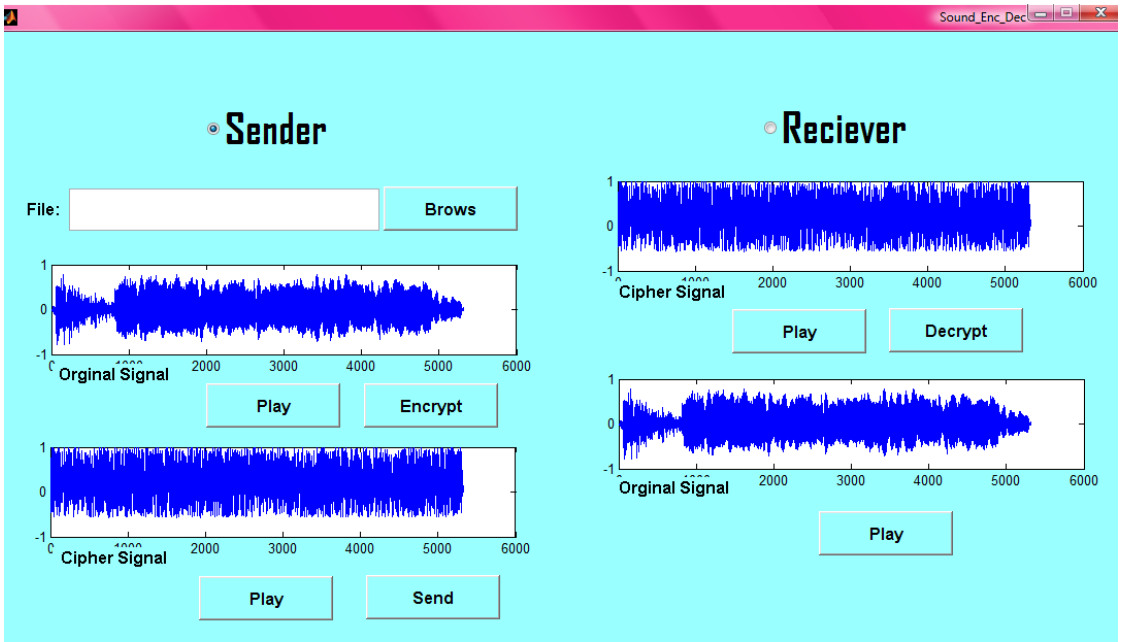


Fig. 8.6: Voice waves encryption / decryption window.

To encrypt or decrypt image files, you need to use the interface shown in Fig. 8.7.

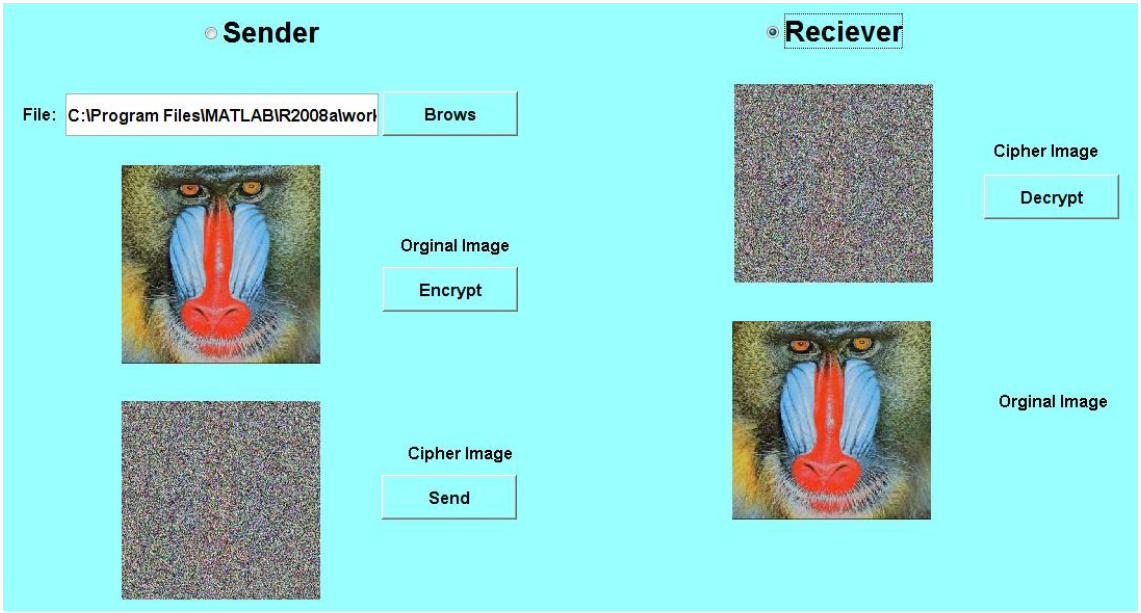


Fig. 8.7: Image encryption / decryption window.

The last user interface demonstrates the security analysis of image cryptosystem as shown in Fig. 8.8.

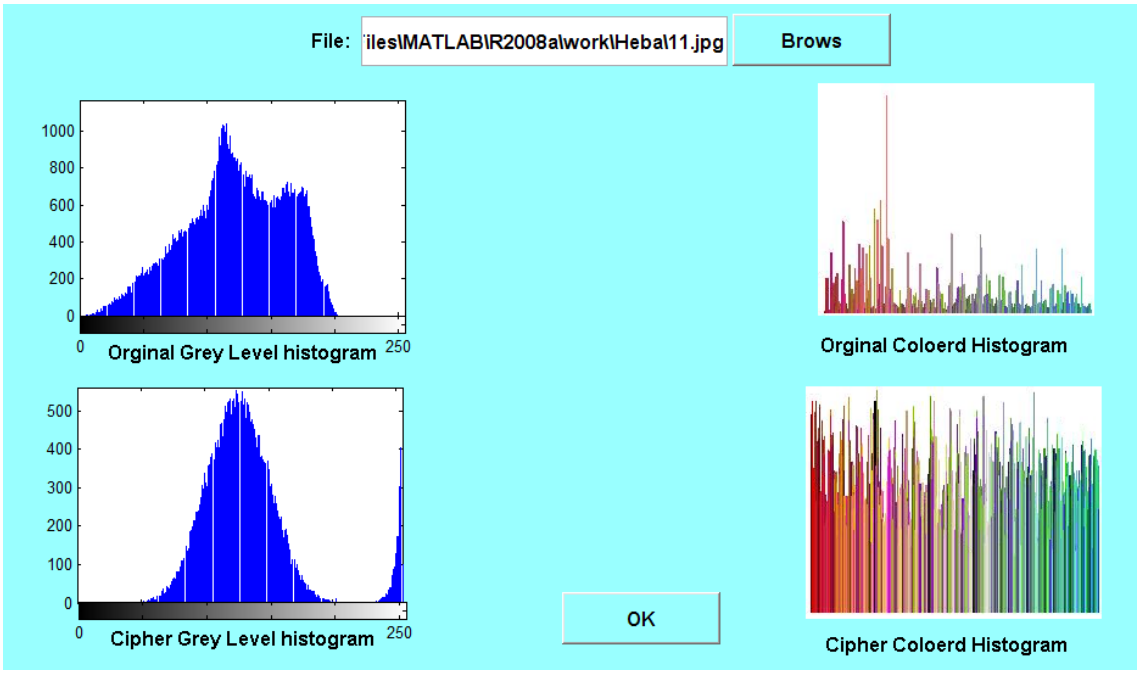


Fig. 8.8: Image cryptanalysis window.

8.3 Overall Security Analysis

In the hybrid cryptosystem, we gather the secure key distribution as a advantage of improved Elgamal cryptosystem, in addition to a secure and fast cryptosystem that deal with multiple data types.

The hybrid schema has the security level of its contents, i.e. as the private key cryptosystem is secure against the statistics analysis; the hybrid cryptosystem is also secure against the statistics analysis, and so on.

So, our hybrid cryptosystem is secure against each of the chosen ciphertext attacks, statistics analysis, plaintext attacks , differential attacks and brute force attacks, it destroys the plaintext characteristics. Furthermore, the sensitivity analysis proves the robustness and the high security level of the proposed cryptosystem. Also the scheme is highly secure against the entropy attacks.

8.4 Summery

In this chapter, the design of the proposed hybrid schema has been described, where the hybrid cryptosystem utilizes both private and public key cryptosystems. The improved Elgamal is used to exchange the keys, while the proposed cryptosystem based on quasi groups and chaotic maps, is used as a private key cryptosystem.

The proposed hybrid cryptosystem takes the advantages of private key cryptosystem in addition to the advantage of presence a key distribution schema. The hybrid cryptosystem is implemented and the security is analyzed and described.

Chapter 9

Conclusion and Future work

The section 9.1 provides the conclusions from this thesis, whilst the future works are proposed in section 9.2.

9.1 Conclusion

In this thesis, a hybrid cryptosystem as proposed, it consists of public and private key cryptosystems, it combine the advantages of these cryptosystem types. Elgamal public key cryptosystem is improved; its disadvantages have been eliminated. In the private key cryptosystem, we investigate the usage of chaotic maps in cryptography, their properties, such as sensitive dependency on initial conditions and system parameters, and random-like outputs, are similar to confusion and diffusion cryptography properties. Furthermore, quasi groups provide a powerful technique for generating a larger set of permutation transformations, so, we adopt the chaotic maps and quasi groups in our proposed cryptosystem.

Experimental results are explored in this thesis to demonstrate the efficiency of the proposed cryptosystem. We illustrate the security of the proposed cryptosystem using different data types, we confirm that the proposed cryptosystem is secure against each of the chosen ciphertext attacks, statistics analysis, plaintext attacks, differential attacks and brute force attacks, it destroys the plaintext characteristics. Furthermore, the sensitivity analysis proves the robustness and the high security level of the proposed cryptosystem. Also the scheme is highly secured against the entropy attacks. It can be observed that the output is random even when the input is highly predictable.

We construct the private key cryptosystem to deal with a sound signal, it performs good results, and it gives high entropy and destroys the characteristics of the original signal.

We develop a novel image cryptosystem by using two color spaces, in order to work in image space components level, which means we change the values of red, green, blue, luminance and color-difference components, for each pixel. This method gives outstanding results; it maximizes the entropy to be very close to the theoretical value, also the results of other measures are very good.

It's clear from the measurements results and visual inspection, that the encrypted image contains new colors (which not existed in the original image), besides the new distribution of colors which appears in the encrypted image.

9.2 Future Work

The work of the thesis may be extended using one of the following developments:

- Use a Lattice-based cryptography, for example NTRU cryptosystem as a public key cryptosystem.
- The proposed cryptosystem can be implemented in parallel structure which would further enhance the productivity of the system.
- In Image cryptosystem, investigate usage other image spaces.
- In the private key cryptosystem, adopt the 2-dimensional and 3-dimensional chaotic maps.

References

[1] A. Menezes, P. van Oorschotand, S. Vanstone, “Handbook of Applied Cryptography,” CRC Press, pp. 4- 15, 516, 1996

[2] H. Delfs and H. Knebl , "Introduction to Cryptography Principles and Applications" , Springer, 2007

[3] W. Stallings, "Cryptography and Network Security Principles and Practices", 4th Ed, Prentice-Hall, 2003.

[4] National Bureau of Standards, "Data Encryption Standard", Federal Information Processing Standard (FIPS), Publication 46, National Bureau of Standards, U.S. Department of Commere , 1997

[5] M. Robshaw, "Stream Cipher", RSA Laboratories Technical Report TR- 701, Version 2.0, July 1995

[6] J. Dj. Golic, "Cryptanalysis of alleged A5 stream cipher," in Advances in Cryptology-EUROCRYPT'97. New York: Springer-Verlag, vol. LNCS 1233, pp 239-255, 1997.

[7] R. L. Rivest, "The RC5 encryption algorithm," in Proceedings of the 1994 Leuven Workshop on Fast Software Encryption, Springer, pp. 86-96, 1995

[8] Niels Ferguson, Doug Whiting, Bruce Schneier, John Kelsey, Stefan Lucks, and Tadayoshi Kohno, " Helix: Fast encryption and authentication in a single cryptographic primitive" In Proceedings of the International Workshop on Fast Software Encryption (FSE 2003), 2003

[9] J. Robert and Jr. ISAAC, "Fast Software Encryption", vol. LNCS 1039 , pp.41, 1996

[10] Department of the Army Publication, "Basic Cryptanalysis", Field Manual 34-40-2 ,Mar 1997

[11] ElGamal, T. "A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms." IEEE Transactions on Information Theory, July 1985.

[12] D. Boneh, A. Joux, P. Nguyen, "Why Textbook ElGamal and RSA Encryption Are Insecure", In Proceedings of the 6th International Conference on the Theory and Application of Cryptology and Information Security, London, UK, pages 30-43, 2000.

[13] J. Buchmann, "An Introduction to Cryptography", 2nd edition , Springer ,2000

[14] R. Rivest, "The MD5 Message-Digest Algorithm", RFC 1321, MIT and RSA Data Security, Inc, 1992.

[15] Secure Hash Standard. Federal Information Processing Standard Publication 180-2. U.S. Department of Commerce, National Institute of Standards and Technology (NIST), 2002

[16] Y .Zheng, , J. Pieprzyk, & J. Seberry, "HAVAL a one-way hashing algorithm with variable length of output" in Proceedings of the Workshop on the Theory and

Application of Cryptographic Techniques: Advances in Cryptology, Springer-Verlag, Berlin, New York, Tokyo, Vol. 718 , pp. 83-104, 1993

[17] Xiaoyun Wang, Yiqun Lisa Yin, Hongbo Yu , " Finding Collisions in the Full SHA-1" , Advances in Cryptology, proceedings of CRYPTO 2005, Lecture Notes in Computer Science, , Vol. 3621, pp. 17-36, 2005

[18] Vinod Patidar, N. K. Pareek, G. Purohit and K.K.Sud, " Modified substitution-diffusion image cipher using chaotic standard and logistic maps," Communications in Nonlinear Science and Numerical Simulation, 2009

[19] JiWon Yoon, ,Hyoungshick Kim , " An image encryption scheme with a pseudorandom permutation based on chaotic maps", Communications in Nonlinear Science and Numerical Simulation, 2010

[20] Zbigniew Kotulski, Janusz Szczepański," Discrete chaotic cryptography", Ann. Physik, vol. 6, pp. 381-394, 1997

[21] R. Schmitz and J. Franklin, "Use of Chaotic Dynamical Systems in Cryptography" , vol. 338, pp. 429-441, 2001.

[22] Fangjun Huang, Zhi-Hong Guan, "Cryptosystem using Chaotic Keys", Chaos Soliton Fractals, Vol. 23, No. 3, pp. 851-855, 2005

[23]L. Kocarev, "Chaos-Based Cryptography: A Brief Overview", IEEE Circuits and System Magazine, Vol. 1 , No. 3, pp. 6-21, 2001.

[24] F. Anstett, G. Millerioux, and G. Bloch, "Chaotic cryptosystems: Cryptanalysis and identifiability," in IEEE Tran. Circuits and Systems I , vol. 53, no. 12, pp. 2673-2680,2006

[25] T. Ritter, "Orthogonal latin squares, nonlinear balanced block mixers," Ritter Software Engineering Report, 1998.

[26] Jean-Charles Faugère, Rune Steinsmo Ødegård, Ludovic Perret, Danilo Gligoroski, " Analysis of the MQQ Public Key Cryptosystem.", CANS 2010, pp. 169-183

[27] Q. Mahesar and V. Sorge, "Classification of Quasigroup - structures with respect to their Cryptographic Properties ". Cryptographic Properties, Proc. of the ARW 2009 Bringing the GAP between Theory and Practice, Liverpool, UK, pp. 23–25, 2009

[28] V. Dimitrova and J. Markovski, "On quasigroup sequence random generator", Proceedings of the 1st Balkan Conference in Informatics, pages 393–401, 2004.

[20] B. McKay and E. Rogoyski , "Latin Squares of Order 10: , Electronic Journal of Combinatorics, , Vol. 2, No. 3, pp. 1-4, 1995

[30] E. Ochodková ans V. Snášel, "Using Quasigroups for Secure Encoding of File System" , Proceedings of the International Scientific NATO PfP/PWP Conference "Security and Information Protection 2001", Brno, Czech Republic, pp.175–181,May ,2001

[31] N. Koblitz, "Elliptic curve cryptosystems," Math. Computer, vol. 48, no.177, pp.203–209, 1987.

[32] V. S. Miller, “Use of elliptic curves in cryptography,” in Advances in Cryptology - CRYPTO’85, H. C. Williams, Ed. Heidelberg, Germany: Springer-Verlag, vol. 218, Lecture Notes in Computer Science, pp. 417–426, 1986,

[33] K. Jarvinen and J. Skytta , " On Parallelization of High-Speed Processors for Elliptic Curve Cryptography " In IEEE Trans. Very Large Scale Integration (VLSI) Systems, vol. 16, no. 9, pp. 1162 – 1175, 2008

[34] R. C. C. Cheung, N. J. Telle, W. Luk, and P. Y. K. Cheung, "Customizable elliptic curve cryptosystems," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 13, no. 9, pp. 1048-1059, 2005.

[35] G. Meurice de Dormale and J.-J. Quisquater, “High-speed hardware implementations of elliptic curve cryptography: A survey,” J. Syst. Ar- chitect., vol 53, pp.72–84, 2007

[36] Yong-Je Choi, Moo-Seop Kim, Hang-Rok Lee, and Ho-Wan Kim, “Implementation and analysis of Elliptic Curve Cryptosystems over Polynomial basis and ONB”, PWAEST, Vol. 10, December, pp. 130-134, 2005.

[37] M.W.Paryasto, Kuspriyanto, S. Sutikno, A. Sasongko, "Issues in Elliptic Curve Cryptography Implementation," Internetworking Indonesia Journal, vol. 1, no.1, pp.29-33, 2009.

[38] Mehrdad S. Sharbaf , " Quantum Cryptography: A New Generation of Information Technology Security System " in Proc. The 6th International Conf. of information Technology: New Generations, Las Vegas, Nevada, pp. 1644-1648, 2009

[39] N. Papanikolaou, “An introduction to quantum cryptography”, ACM Crossroads Magazine, vol. 11, no. 3, pp. 1-16, 2005

[40] Billy Bob Brumley,and Kimmo U. Järvinen , " Conversion Algorithms and Implementations for Koblitz Curve Cryptography ECC " in IEEE Tran. Computers, vol. 59, no. 1, pp. 81-92, 2010

[41] T. Lange, “Koblitz Curve Cryptosystems,” Finite Fields and their Applications, vol. 11, no. 2, pp. 200-229, 2005.

[42] N.K. Pareek , Vinod Patidar , K.K. Sud," Discrete chaotic cryptography using external key", Phys Lett A , vol. 309, pp. 75–82, 2003.

[43] G. Álvarez , F. Montoya, M. Romera , G. Pastor," Cryptanalysis of a discrete chaotic cryptosystem using external key". Phys Lett A, vol. 319, pp. 334–339, 2003

[44] N.K. Pareek a,b, Vinod Patidar A. and K.K. Sud, " Cryptography using multiple one-dimensional chaotic maps", Communications in Nonlinear Science and Numerical Simulation, vol. 10, pp. 715–723, 2005

[45] Jun Wei , Xiaofeng Liao , Kwok-wo Wong , and Tsing Zhou, " Cryptanalysis of a cryptosystem using multiple one-dimensional chaotic maps", Communications in Nonlinear Science and Numerical Simulation, vol. 12, pp. 814–22, 2007

[46] Tao Xiang , Kwok-wo Wong , and Xiaofeng Liao," An improved chaotic cryptosystem with external key", Communications in Nonlinear Science and Numerical Simulation, vol. 14, pp. 574–581, 2008

[47] C. Kościenly, "Generating quasi groups for cryptographic applications," Int. J. Appl. Math. Computer. Sci., vol. 12, no. 4, pp. 559–569, 2002

[48] D. Gligoroski, S. Markovski, and S.J. Knapskog, "Multivariate Quadratic Trap-door Functions Based on Multivariate Quadratic Quasi groups," In Proceedings of the American Conference on Applied Mathematics, (MATH08), Cambridge, Massachusetts, USA, 2008.

[49] M. Satti and S. Kak, "Multilevel indexed quasigroup encryption for data and speech," in IEEE Trans. on Broadcasting, vol. 55, pp. 270-281,2009

[50] Abdelfatah A. Yahya and Ayman M. Abdalla, "A Shuffle Image-Encryption Algorithm" Journal of Computer Science, vol. 4, no. 12, pp. 999-1002, 2008

[51] F. Liu, C.K. Wu and X.J. Lin, " Colour visual cryptography schemes " in Information Security, IET, vol. 2, no. 4, pp. 151 - 165, 2008

[52] X. Tong, M. Cui, "Image encryption with compound chaotic sequence cipher shifting dynamically", Image and Vision Computing , vol. 26, no. 6, pp. 843–850, 2008

[53] Chengqing Li, Shujun Li, Guanrong Chen, Wolfgang A. Halang ," Cryptanalysis of an image encryption scheme based on a compound chaotic sequence ", Image and Vision Computing , vol. 27 , pp. 1035–1039 , 2009

[54] Guosheng Gu ,Guoqiang Han "An Enhanced Chaos Based Image Encryption Algorithm", IEEE Proceedings of the First International Conference on Innovative Computing, Information and Control (ICICIC'06) in 2006.

[55] D. Dolev, C. Dwork, and M. Naor. "Nonmalleable cryptography". SIAM J. Comput., Vol. 30, no. 2, pp. :391–437, 2000..

[56] Chosen-ciphertext attack. http://en.wikipedia.org/wiki/Chosen-ciphertext_attack , Wikipedia. [Online] Wikimedia Foundation Inc., 5 20, 2011. [Cited: 5 29, 2011].

[57] Entropy (information theory). http://en.wikipedia.org/wiki/Entropy_(information_theory). Wikipedia. [Online] Wikimedia Foundation Inc., 5 25, 2011. [Cited: 5 29, 2011].

[58] Analysis, N-Gram. http://www.cryptool-online.org/index.php?option=com_content&view=article&id=94&Itemid=112&lang=en . cryptool-online. [Online][Cited: 5 29, 2011].

[59] N-gram. http://en.wikipedia.org/wiki/N-gram. Wikipedia. [Online] Wikimedia Foundation Inc., 5 22, 2011. [Cited: 5 29, 2011].

[60] Autocorrelation. http://en.wikipedia.org/wiki/Autocorrelation, Wikipedia. [Online] Wikimedia Foundation Inc., 5 27, 2011. [Cited: 5 29, 2011].

[61] K. Nahrstedt and R. Steinmetz, " Multimedia Fundamentals: Media Coding and Content Processing", Prentice Hall, 2002

[62] RGB color space. http://en.wikipedia.org/wiki/RGB_color_space, Wikipedia. [Online] Wikimedia Foundation Inc., 5 May 2011. [Cited: 5 29, 2011].

[63]YCbCr color space. http://en.wikipedia.org/wiki/YCbCr, Wikipedia. [Online] Wikimedia Foundation Inc., 31 May 2011. [Cited: 5 31, 2011].

[64] D. Chattopadhyay, M. K. Mandal and D. Nandi, " Robust Chaotic Image Encryption based on Perturbation Technique " , ICGST-GVIP Journal, Vol. 11, Issue 2, pp. 41- 50 , April 2011

[65] A. Awad and D. Awad, " Efficient Image Chaotic Encryption Algorithm with No Propagation Error " , ETRI Journal, Vol. 32, No. 5, pp. 774- 783, October 2010

[66] P. Fei, S. S. Qiu and L. Min, "An Image Encryption Algorithm Based on Mixed chaotic Dynamic Systems and External Keys", IEEE, pp. 1135-1139, 2005

[67] L. Zhang, X. Liao and X. Wang, "An Image Encryption Approach Based on Chaotic Maps", Chaos, Solitons & Fractals, vol. 24, No. 3, pp. 759-765 , 2005

[68] H. Gao, Y. Zhang, S. Liang, and D. Li, "A New Chaotic Algorithm for Image Encryption ", Chaos, Solitons & Fractals, vol. 29, No. 2, pp. 393-399 , 2006

[69] G. M. Kumar and V. Chandrasekaran, "A Generic Framework for Robust Image Encryption Using Multiple Chaotic Flows",International Journal of Computational Cognition, vol. 8, No. 3, Sept 2010.

[70] P. Fei, S. S. Qiu and L. Min. An Image Encryption Algorithm Based on Mixed chaotic Dynamic Systems and External Keys, IEEE, 2005, pp. 1135-1139.

[71] Latin square. http://en.wikipedia.org/wiki/Latin_square, Wikipedia. [Online] Wikimedia Foundation Inc., 5 20, 2011. [Cited: 5 29, 2011].

[72] E. Ochodkova, J. Dvorský, V. Snásel, A. Abraham, "Large Quasigroups in Cryptography and their Properties Testing", In Proc. of Nature & Biologically Inspired Computing (NaBIC), Coimbatore, India, pp. 965- 971, 2009.

[73] MD2 (cryptography). http://en.wikipedia.org/wiki/ MD2_(cryptography), Wikipedia. [Online] Wikimedia Foundation Inc., 4 29, 2011. [Cited: 5 31, 2011].

[74] SHA-2. http://en.wikipedia.org/wiki/SHA-2 . Wikipedia. [Online] Wikimedia Foundation Inc., 24 May 2011[Cited: 5 31, 2011].

[75] W. Diffie and M. Hellman . "New Directions in Cryptography". IEEE Transactions on Information Theory , Vol. IT-22, No.6, pp. 644–654, November 1976

[76] Diffie-Hellman , http://library.thinkquest.org/C0126342/dh.htm, Oracle thinkquest, [Online]. [Cited: 5 31, 2011].

[77] Diffie-Hellman key exchange. http://en.wikipedia.org/wiki/Diffie-Hellman_key_exchange, Wikipedia. [Online] Wikimedia Foundation Inc., 3 14, 2011. [Cited: 5 31, 2011].

[78] M.S. Hwang, C.C. Chang, and K.F. Hwang, "An ElGamal-like cryptosystem for enciphering large messages," *IEEE Trans. Knowledge and Data Engineering*, vol. 14, no. 2, pp. 445{446, 2002**.**

Appendices

Appendix A

Mathematical Background

Here, we clarify some related mathematical concepts to help the reader to understand this thesis.

A.1: Function

A *set* consists of distinct objects which are called *elements* of the set. For example, a set X might consist of the elements a, b, c, and this is denoted $X = \{a;\ b;\ c\}$.

***Definition A.1*:** A *function* is defined by two sets $X\ and\ Y$ and a *rule* f which assigns to each element in X precisely one element in Y . The set X is called the *domain* of the function and Y the *co-domain*. If x is an element of X (usually written $x \in X$) the *image* of x is the element in Y which the rule f associates with x; the image y of x is denoted by $y = f(x)$.

Standard notation for a function f from set $X\ to\ set\ Y\ is\ f : X \to Y\ .If\ y \in Y$, then a *preimage* of y is an element $x \in X$ for which $f(x) = y$. The set of all elements in Y which have at least one preimage is called the *image* of f, denoted $Im(f)$.

Example: Take $X = \{1, 2, 3, \dots, 10\}$ and let f be the rule that for each x $\in$ X, $f(x) = r_x$, where r_x is the remainder when x^2 is divided by 11. Explicitly then

$f(1) = 1, f(2) = 4, f(3) = 9, f(4) = 5, f(5) = 3, f(6) = 3,\ f(7) = 5,\ f(8) = 9, f(9) = 4, f(10) = 1$.The image of $f\ is\ the\ set\ Y = \{1;\ 3;\ 4;\ 5;\ 9\}$. [1]

A.2: Latin Square

A Latin square of order n is a $n \times n$ matrix in which n^2 symbols, taken from a set A, are arranged so that each symbol occurs only once in each row and exactly once in each column. For any positive integer n there exists a Latin square of order n. A Latin square of order n for which

$$A = \{0, 1, \ldots, n-1\} \quad \text{(A.1)}$$

is said to be normalized or reduced if the elements of both its first row and its first column are in a natural order. [47, 71, 72]. For example, the two Latin squares of order two are given by

$$\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}, \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

A.3: Modular Exponentiation of $\mathbb{Z}_p^*$

Modular exponentiation can be performed efficiently with the repeated square-and multiply algorithm, which is crucial for many cryptographic protocols.

This algorithm is based on the following observation. Let the binary representation of k be $\sum_{i=o}^{t} k_i\, 2^i$, where each $k_i \in \{0, 1\}$. Then

$$a^k = \prod_{i=0}^{t} a^{k_i 2^i} = (a^{2^0})^{k_0}\ (a^{2^1})^{k_1}\ \ldots\ (a^{2^t})^{k_t}$$

Algorithm Repeated square-and-multiply algorithm for exponentiation in $\mathbb{Z}_n$

Inputs:$a \in \mathbb{Z}_n$, and integer $0 \leq k < n$ whose binary representation is $k = \sum_{i=o}^{t} k_i\, 2^i$.

Outputs: $a^k \bmod n$.

Procedure:

Step 1: Set $b \leftarrow 1$. If $k = 0$ then return (b).

Step 2: Set$A \leftarrow a$.

Step 3: If $k_0 = 1$ then set $b \leftarrow a$.

Step 4: For i from $1\ to\ t$ do the following:

4.1: Set $A \leftarrow A^2 \bmod n$.

4.2: If $k_i = 1$ then set $b \leftarrow A.b \bmod n$.

Step 5: Return (b) [1].

A.4: Selecting a Prime p and Generator of $\mathbb{Z}_p^*$

Algorithm Selecting a k-bit prime p and a generator α of $\mathbb{Z}_p^*$

Inputs: the required bit length k of the prime and a security parameter t.

Outputs: a k-bit prime p such that $p - 1$ has a prime factor $\geq t$, and a generator α of $\mathbb{Z}_{\mathbf{p}}^*$

Procedure:

Step 1: Repeat the following:

1.1 Select a random k-bit prime p

1.2 Factor $p - 1$.

Until $p - 1$ has a prime factor $\geq t$.

Step 2: Use the following algorithm with $G = \mathbb{Z}_{\boldsymbol{p}}^*$ and $n = p - 1$ to find a generator α of $\mathbb{Z}_{\mathbf{p}}^*$.

Step 3: Return(p, α)

Algorithm Finding a generator of a cyclic group

Inputs: a cyclic group G of order n, and the prime factorization $n = p_1^{e_1} p_2^{e_2} \ldots p_k^{e_k}$

Outputs: a generator α of G.

Procedure:

Step 1: Choose a random element $\alpha \; in \; G$.

Step 2: For i from 1 to k do the following:

2.1: Compute $b \leftarrow \alpha^{n/p_i}$.

2.2: If $b = 1$ then go to step 1.

Step 3: Return (α) [1].

Appendix B

Colors Spaces

A color space is a mathematical representation of a set of colors. The most popular color models are RGB (used in computer graphics); YIQ, YUV or YCbCr (used in video systems; and CMYK (used in color printing). All of the color spaces can be derived from RGB information**.**

B.1: RGB Color Space

Red, green and blue are three primary additive (individual components are added together to form a desired color) and are represented in Fig. B.1 by a three-dimensional [62].

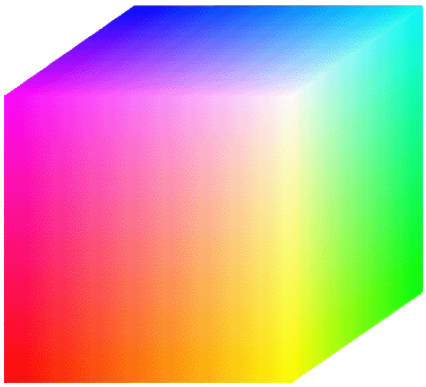


Fig. B.1: RGB color cube.

The indicated diagonal of cube, with equal amounts of each primary component represents various gray levels. Table B.1 contains the RGB values for 100% amplitude, 100% saturated color bars.

Table B.1: 100% RGB color bar.

	Nominal range	White	Yellow	Cyan	Green	Magenta	Red	Blue	Black
R	0 to 255	255	255	0	0	255	255	0	0
G	0 to 255	255	255	255	255	0	0	0	0
B	0 to 255	255	0	255	0	255	0	255	0

The RGB color space is the most prevalent choice for computer graphics because color displays use red, green and blue to create the desired color. Therefore, the choice of the RGB color space simplifies the system design. Also, a system that is designed using RGB color space can take advantage of a large number of existing software routines.

B.2: YCbCr Color Space

Y is the luminance which meaning that light intensity is non-linearly encoded using gamma correction, component and Cb and Cr are the blue-difference and red-difference chroma components .

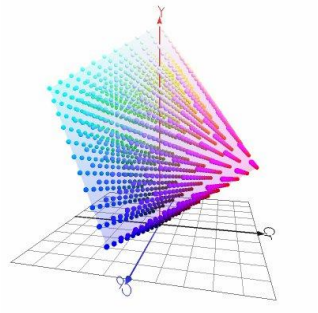


Fig. B.2: YCbCr color space.

YCbCr is not an absolute color space; it is a way of encoding RGB information. The actual color displayed depends on the actual RGB primaries used to display the signal. Therefore a value expressed as YCbCr is predictable only if standard RGB primary chromaticities are used [63].

Y is defined to have a nominal range 16-235; Cb and Cr are defined to have a nominal range 16-240.

$$Y = (77/256)R + (150/256)G + (29/256)B$$

$$Cb = -(44/256)R - (87/256)G + (131/256)B + 128$$

$$Cr = (131/256)R - (110/256)G - (21/256)B + 128$$

Table B.2: 75% YCbCr color bar

		Nominal range	White	Yellow	Cyan	Green	Magenta	Red	Blue	Black
SDTV	Y	16-235	180	162	131	112	84	65	35	16
	Cb	16-240	128	44	156	72	184	100	212	128
	Cr	16-240	128	142	44	58	198	212	114	128
HDTV	Y	16-235	180	168	145	133	63	51	28	16
	Cb	16-240	128	44	147	63	193	109	212	128
	Cr	16-240	128	136	44	52	204	212	120	128

Appendix C

Some Hash Functions

C.1: MD2

The MD2 Message-Digest Algorithm is a cryptographic hash function developed by Ronald Rivest in 1989. The algorithm is optimized for 8-bit computers.

The 128-bit hash value of any message is formed by padding it to a multiple of the block length on the computer (128 bits or 16 bytes) and adding a 16-byte checksum to it. For the actual calculation, a 48-byte auxiliary block and a 256-byte S-table generated indirectly from the digits of the fractional part of pi are used. The algorithm runs through a loop where it permutes each byte in the auxiliary block 18 times for every 16 input bytes processed. Once all of the blocks of the (lengthened) message have been processed, the first partial block of the auxiliary block becomes the hash value of the message [73].

C.2: SHA-256, SHA-384 and SHA-512

SHA stands for Secure Hash Algorithm. SHA-1: A 160-bit hash function which resembles the earlier MD5 algorithm. This was designed by the National Security Agency (NSA) to be part of the Digital Signature Algorithm.

SHA-2: A family of two similar hash functions, with different block sizes, known as SHA-256 and SHA-512. They differ in the word size; SHA-256 uses 32-byte (256 bits) words where SHA-512 uses 64-byte(512 bits) words. There are also truncated versions of each standardized, known as SHA-224 and SHA-384. These were also designed by the NSA.

SHA-2 is a set of cryptographic hash functions (SHA-224, SHA-256, SHA-384, SHA-512) designed by the (NSA) and published in 2001 by the NIST as a U.S. Federal Information Processing Standard. SHA-2 includes a significant number of changes from its predecessor, SHA-1. SHA-2 consists of a set of four hash functions with digests that are 224, 256, 384 or 512 bits.

In 2005, security flaws were identified in SHA-1, namely that a mathematical weakness might exist, indicating that a stronger hash function would be desirable. Although SHA-2 bears some similarity to the SHA-1 algorithm, these attacks have not been successfully extended to SHA-2 [74].

Appendix D

Diffie-Hellman

The Diffie-Hellman key agreement protocol (also called exponential key agreement) was developed by Diffie and Hellman in 1976 and published in the ground-breaking paper "New Directions in Cryptography". The protocol allows two users to exchange a secret key over an insecure medium without any prior secrets [75].

The protocol has two system parameters p and g. they are both public and may be used by all the users in a system. Parameter p is a prime number and parameter g (usually called a generator) is an integer less than p, with the following property: for every number n between 1 and $p-1$ inclusive, there is a power k of g such that $n = g^k \ mod \ p$.

Using the Diffie-Hellman key agreement protocol

In this case, Alice and Bob want to agree on a shared secret key using the Diffie-Hellman key agreement protocol [76, 77].

1. Alice generates a random private value a and Bob generates a random private value b. Both a and b are drawn from the set of $integers \ \{1, ¼, p-2\}$.
2. They derive their public values using parameters p and g and their private values. Alice's public value is $g^a \ mod \ p$ and Bob's public value is $g^b \ mod \ p$.
3. They exchange their public values.
4. Alice computes $g^{ab} = (g^b)^a \ mod \ p$, and Bob $computes \ g^{ba} = (g^a)^b \ mod \ p$.
5. Since $g^{ab} = g^{ba} = k$, Alice and Bob now have a shared secret key k.

The protocol depends on the discrete logarithm problem for its security. It assumes that it is computationally infeasible to calculate the shared secret key $k = g^{ab} mod \ p$ given the two public values $g^a \ mod \ p$ and $g^b \ mod \ p$ when the prime p is sufficiently large. Maurer has shown that breaking the Diffie-Hellman protocol is equivalent to computing discrete logarithms under certain assumptions.