



Sudan University of Science and Technology
College of Graduate Studies



***Improve Data Encryption Standard
using 256Bit key Length***

**تحسين خوارزمية تشفير البيانات القياسية باستخدام المفتاح
الثنائي بطول 256**

A supplementary research is submitted to meet the
requirements for obtaining a master degree in information
technology

Preparation:-

Alfatih Mohamed Abkr Mohamed

Supervision:-

Dr. Faisal Mohamed Abdulla Ali

January 2019

Table of Contents

آية.....	i
Dedication.....	ii
Thanks and appreciation	iv
Abstract.....	v
المستخلص	vi
List Of Figures.....	vii
List of Tables	viii
List Of Abbreviations	ix
CHAPTER I.....	1
INTRODUCTION	1
1.1. Preface.....	1
1.2. Research Problem	3
1.3. Research Objectives	3
1.4. Importance of Research.....	3
1.5. Research Methodology.....	3
1.6. Research Contents	3
CHAPTER II	4
Literature review.....	4
2.1. Introduction.....	4
2.2. Cryptography	4
2.3. Cryptanalysis.....	5
2.4. Security Services of Cryptography	5
2.5. Cryptography Primitives	6
2.6. Cryptosystem	7
CHAPTER III.....	40
Methodology.....	40
3.1. Introduction.....	40
3.2. Proposed Work.....	41
3.3. How The Algorithm Works	41
3.3.1. Steps of an algorithm to perform encryption:	41
3.3.2. Steps of an algorithm to perform decryption:	41
3.4. Algorithm structure	42
CHAPTER IV.....	45
Implementation and results.....	45

4.1.	INTRODUCTION	45
4.2.	IMPLEMENTATION	45
4.3.	RESULT	46
4.4.	Test And Discussion	47
4.4.1.	The Frequency Test on E-DES	47
4.4.2.	The Block Test on E-DES	48
4.4.3.	The Runs Test on E-DES	49
4.4.4.	The Frequency Test on Data Encryption Standard [DES]	50
4.4.5.	The Block Test on Data Encryption Standard [DES]	51
4.4.6.	The Runs Test on Data Encryption Standard [DES].....	52
CHAPTER V	54
CONCLUSIONS	54
5.1.	CONCLUSIONS.....	54
<i>References</i>	55

آية

قال تعالى:-

(وَعَلَّمَ آدَمَ الْأَسْمَاءَ كُلَّهَا ثُمَّ عَرَضَهُمْ عَلَى الْمَلَائِكَةِ فَقَالَ أَنْبِئُونِي بِأَسْمَاءِ هَؤُلَاءِ إِنْ كُنْتُمْ صَادِقِينَ (31) قَالُوا سُبْحَانَكَ لَا عِلْمَ لَنَا إِلَّا مَا عَلَّمْتَنَا ۗ إِنَّكَ أَدَبُ الْعَلِيمِ الْعَظِيمِ (32) قَالَ يَا آدَمُ أَنْبِئْهُمْ بِأَسْمَائِهِمْ ۗ فَلَمَّا أَنْبَأَهُمْ بِأَسْمَائِهِمْ قَالَ أَلَمْ أَقُلْ لَكُمْ إِنِّي أَنزَلْتُ مِنَ السَّمَاءِ مَاءً وَالْأَرْضِ وَأَنْزَلْتُ مَا تُنْبِئُونَ وَمَا كُنْتُمْ تَكْتُمُونَ (33))

صدق الله العظيم

سورة البقرة

Dedication

*I dedicate my dissertation work to my supporter
and my strength and my refuge after God*

*To the pot of the cup is empty to hold me a drop of love
To those who ate his hope to give us a moment of happiness
To the one who took the thorns from Derby to guide me
through the flag to the great heart*

.....My dear father

*To those who gave me love and tenderness
To the symbol of love and balm healing to the heart pure white
My beloved mother*

*To those who have shown themselves to those who have shown
me what is more beautiful than life*

My brothers

To the Spirit that inhabited my soul and supported me

...Baby

He did not let me down

...to Dr. Faisal Mohamed Abdallah

My beloved teacher taught me

To whom you have had the most beautiful moments

'I will miss them ... and I hope they miss me

To whom God made my brothers in God My friends

To those who combine my happiness and my sorrow

To those who did not know them they will not know me

To whom I wish to remind them if they remind me

Thank you all

Thanks and appreciation

Praise is to Allah, Lord of the Worlds, and prayers and peace is upon the teacher of mankind and the guidance of humanity and on his family and companions and those who follow them in charity to the Day of Judgment.

Thanks to God alone and the Almighty who is in his hand facilitate things

I would like to express my deep thanks to all those who contributed to the implementation of this research, to all who have been the cause of my education, guidance and assistance.

Then thanks to the University of Sudan

And then thanks to the family of the College of Computer Science and Technology

Abstract

Network security is becoming more crucial as the volume of data being exchanged on the internet. A more practical way to protect information is to alter it so that only an authorized receiver can understand it. Security of data and telecommunication can be done by a technique called cryptography. In this thesis Data Encryption Standard algorithm has been improved. Improved algorithm uses input data 512bit and key of 256bit length to encrypt the data. It uses S-boxes similar to those of Data Encryption Standard [DES] in a new structure that simultaneously allows a more rapid avalanche. The aim is to develop stronger encryption system that results in minimum execution time and maximum security. security analysis is applied on proposed algorithm using National Institute of Standards and Technology [NIST] statistical test tool with cryptool ,the results show that reasonable strength.

المستخلص

أصبح أمن الشبكات أكثر أهمية مع زيادة حجم البيانات المتبادلة على الإنترنت. وهناك طريقة أكثر عملية لحماية المعلومات وهي تغييرها بحيث لا يتمكن سوى المستلم المرخص له من فهمها. يمكن إجراء أمن للبيانات والاتصالات عن طريق تقنيات التشفير. في هذا البحث تم تطوير خوارزميه تشفير البيانات القياسيه. تستخدم هذه الخوارزمية بيانات الإدخال bit 512 ومفتاح طولها bit 256 لتشفير البيانات. ويستخدم-S boxesمشابهة لتلك الخاصة بـ خوارزميه تشفير البيانات القياسيه في بنية جديدة. الهدف هو تطوير نظام تشفير قوى يأخذ وقت اقل في عمليه التشفير وفك التشفير ويوفر حد أقصى من الأمان. تم اجراء اختبار علي الخوارزمية المقترحه باستخدام ادوات المعهد الوطني للمعايير والتكنولوجيا وCRYPTOOL وظهرت نتائج افضل.

List Of Figures

Figure 2.1 simple model of a cryptosystem that provides confidentiality	7
Figure 2.2 symmetric cryptosystems	10
Figure 2.3 symmetric cryptosystems	11
Figure 2.4 a passive attack.....	14
Figure 2.5 active attack.....	15
Figure 2.6 Block & Stream Cipher	20
Figure 2.7 Block Cipher	20
Figure 2.8 Feistel structure	22
Figure 2.9 Key schedule	25
Figure 2.10 Round Function.....	25
Figure 2. 11 Expansion Permutation Box	26
Figure 2.12 Permutation logic	26
Figure 2.13 permutation logic described as table	27
Figure 2.14 Array of S-Boxes.....	27
Figure 2.15 S-box rule.....	28
Figure 2.16 S-box table	28
Figure 2.17 Key Generation	29
Figure2.18 Electronic Codebook mode	31
Figure 2.19 Cipher Block Chaining.....	32
Figure 2.20 Cipher Feedback.....	34
Figure 2.21 Output Feedback	35
Figure 2.22 Counter.....	36
Figure 3.1 General Structure.....	42
Figure 3.2 Single Round.....	42
Figure 3.3 Generate Sub Key.....	42
Figure 3.4 Function proceed XOR to right block with subkey	42
Figure 3.5 Cipher And Reverse Cipher For The First Approach.....	42
Figure 4.1 Encryption Result.....	45
Figure 4.2. Decryption Result.....	46
Figure 4.3. Frequency Test on E-DES	48
Figure 4.4 Data Encryption Standard [DES] Test Result with Cryptool	48
Figure 4.5. Block Test on E-DES	49
Figure 4.6. Runs Test on E-DES	50
Figure 4.7. Data Encryption Standard [DES] Test Result With Nist.....	51
Figure 4.8. E-DES Test Result with Cryptool	51
Figure 4.9. Block Test on Data Encryption Standard [DES] With Nist.....	52
Figure 4.10. Runs Test on Data Encryption Standard With Nist	53

List of Tables

Table 2.1 shows the primitives that can achieve a particular security service on their own.	7
Table 2.2 basic key properties of two types of cryptosystems.....	12
Table 3.1 Comparison of DES, Triple DES, AES and Blow Fish algorithm.....	40
Table 4.1 Compare Between DES, E-DES, AES In Encryption And Decryption Speed	46
Table 4.2 Compare Between DES, E-DES In Randomness Test	47

List Of Abbreviations

Abbreviation	Meaning
AES	Advanced Encryption Standard
ARX	Add-Rotate-Xor
BFA	Brute Force Attack
CBC	Cipher Block Chaining
CFB	Cipher Feedback
CIA	Confidentiality, Integrity And Availability
CTR	Counter
DES	Data Encryption Standard
ECB	Electronic Code Book
IDEA	International Data Encryption
IV	Initialization Vector
LEA	Link Encryption Algorithm
MAC	Message Authentication codes
MD5	Message-Digest Algorithm
MIM	Man in Middle Attack
NIST	National Institute of Standards and Technology
OFB	Output Feedback
RSA	Rivest-Shamir-Adleman
SCA	Side Channel Attack

CHAPTER I

INTRODUCTION

1.1. Preface

In Internet era, whenever any confidential and sensitive information transmitted over a public channel, there is possibility that an eavesdropper could intercept this message and steal this sensitive information. To protect the principles of cryptography are used. Cryptography is the study of transmitting secure messages and the art of secret writing by which the sensitive information may be prevented from any adversary. The general concept behind the cryptography is that the sender selects a message would like to transmit, applies some sort of encryption process, and transmits this encrypted message across the channel. The receiver obtains the encrypted message, also referred to as a cipher text, uses a known decryption process to recover the sender's original message. If an adversary intercepts an encrypted message, will be unable to recover the original message without knowledge of the secret decryption process. There are various types of cryptographic techniques available for securing the sensitive information based on symmetric key cryptography and public key cryptography. In symmetric key cryptography, sender and receiver use a shared key for encryption and decryption, known as secret key. Data Encryption Standard [DES], Three key triple-Data Encryption Standard [3-DES], Advanced Encryption Standard [AES], International Data Encryption [IDEA] are some of the most famous symmetric key algorithms. In public key cryptography, sender uses public key of receiver, known to everyone, to encrypt the message and receiver uses his private key, known only to him, to

decrypt the message. Rivest–Shamir–Adleman [RSA] is the one of the most famous public key algorithm which is based on Diffie-Hellman Key Exchange [1].

Nowadays, a lot of applications based on internet are emerged such as on-line shopping, stock trading, internet banking and electronic bill payment etc. Such transactions takes place through wire or wireless public networks demand end-to-end secure connections, should be confidential, to ensure data authentication, accountability and confidentiality, integrity and availability, also known as Confidentiality, Integrity And Availability [CIA] triad [2]. Security is the mechanism by which information and services are protected from unintended or unauthorized access, change or destruction. Security in networking is based on Cryptography (a word with Greek origins, means “secret writing”), the science and art of transforming messages to make them secure and immune to attack [3]. Encryption is one of the principal means to guarantee security of sensitive information. Encryption algorithm performs various substitutions and transformations on the plaintext (original message before encryption) and transforms it into cipher text (scrambled message after encryption).

For many applications, the Data Encryption Standard algorithm is nearing the end of its useful life. Its 56-bit key is too small, as shown by a recent distributed key search exercise [3]. Although triple-Data Encryption Standard [3-DES] can solve the key length problem, the Data Encryption Standard [DES] algorithm was also designed primarily for hardware encryption, yet the great majority of applications that use it today implement it in software, where it is relatively inefficient.

1.2. Research Problem

Data Encryption Standard [DES] Algorithm suffer from many attacks[3] many of these attacks due to short length of the key used, also un sufficient block size in now days applications

1.3. Research Objectives

The objective of this project is to design new encryption algorithm that is capable to be strong against brut force attack

1.4. Importance of Research

The significance of this work is to increase the security and privacy of the sensitive data that is send through an open system such as internet by design a new encryption algorithm

1.5. Research Methodology

This research was followed by a revision of the DES algorithm and identified its weaknesses and then an algorithm was designed to address these points.

1.6. Research Contents

This research contains of five chapters.

Chapter one contains introduction about information security and research problem and objective and the importance of this research and the way to solve the problem

Chapter two contains related work.

Chapter three contains proposed work.

Chapter four contains implementation and test.

Chapter five contains importance result.

CHAPTER II

Literature review

2.1. Introduction

Cryptography referred almost exclusively to encryption, which is the process of converting ordinary information (called plaintext) into unintelligible form (called ciphertext).

Decryption is the reverse, in other words, moving from the unintelligible ciphertext back to plaintext. A cipher (or cypher) is a pair of algorithms that create the encryption and the reversing decryption. The detailed operation of a cipher is controlled both by the algorithm and in each instance by a "key". The key is a secret (ideally known only to the communicants), usually a short string of characters, which is needed to decrypt the ciphertext. Formally, a "cryptosystem" is the ordered list of elements of finite possible plaintexts, finite possible cyphertexts, finite possible keys, and the encryption and decryption algorithms which correspond to each key. Keys are important both formally and in actual practice, as ciphers without variable keys can be trivially broken with only the knowledge of the cipher used and are therefore useless (or even counter-productive) for most purposes.

2.2. Cryptography

Cryptography is the art and science of making a cryptosystem that is capable of providing information security.

Cryptography deals with the actual securing of digital data. It refers to the design of mechanisms based on mathematical algorithms that provide fundamental

information security services. You can think of cryptography as the establishment of a large toolkit containing different techniques in security applications.

2.3. Cryptanalysis

The art and science of breaking the cipher text is known as cryptanalysis.

Cryptanalysis is the sister branch of cryptography and they both co-exist. The cryptographic process results in the cipher text for transmission or storage. It involves the study of cryptographic mechanism with the intention to break them. Cryptanalysis is also used during the design of the new cryptographic techniques to test their security strengths.

Cryptography concerns with the design of cryptosystems, while cryptanalysis studies the breaking of cryptosystems.

2.4. Security Services of Cryptography

The primary objective of using cryptography is to provide the following four fundamental information security services. Let us now see the possible goals intended to be fulfilled by cryptography.

2.4.1. Confidentiality

Confidentiality is the fundamental security service provided by cryptography. It is a security service that keeps the information from an unauthorized person. It is sometimes referred to as privacy or secrecy.

Confidentiality can be achieved through numerous means starting from physical securing to the use of mathematical algorithms for data encryption.

2.4.2. Data Integrity

It is security service that deals with identifying any alteration to the data. The data may get modified by an unauthorized entity intentionally or accidentally. Integrity service confirms that whether data is intact or not since it was last created, transmitted, or stored by an authorized user.

Data integrity cannot prevent the alteration of data, but provides a means for detecting whether data has been manipulated in an unauthorized manner.

2.4.3. Authentication

Authentication provides the identification of the originator. It confirms to the receiver that the data received has been sent only by an identified and verified sender.

Authentication service has two variants:

Message authentication identifies the originator of the message without any regard router or system that has sent the message.

Entity authentication is assurance that data has been received from a specific entity, say a particular website.

Apart from the originator, authentication may also provide assurance about other parameters related to data such as the date and time of creation/transmission.

2.4.4. Non-repudiation

It is a security service that ensures that an entity cannot refuse the ownership of a previous commitment or an action. It is an assurance that the original creator of the data cannot deny the creation or transmission of the said data to a recipient or third party.

Non-repudiation is a property that is most desirable in situations where there are chances of a dispute over the exchange of data. For example, once an order is placed electronically, a purchaser cannot deny the purchase order, if non-repudiation service was enabled in this transaction.

2.5. Cryptography Primitives

Cryptography primitives are nothing but the tools and techniques in Cryptography that can be selectively used to provide a set of desired security services:

Encryption

Table 2.1 shows the primitives that can achieve a particular security service on their own.

Primitives Service	Encryption
Confidentiality	Yes
Integrity	No
Authentication	No
Non Reputation	No

Cryptographic primitives are intricately related and they are often combined to achieve a set of desired security services from a cryptosystem.

2.6. Cryptosystem

A cryptosystem is an implementation of cryptographic techniques and their accompanying infrastructure to provide information security services. A cryptosystem is also referred to as

Cipher system.

A simple model of a cryptosystem that provides confidentiality to the information being transmitted. This basic model is depicted in the illustration below:

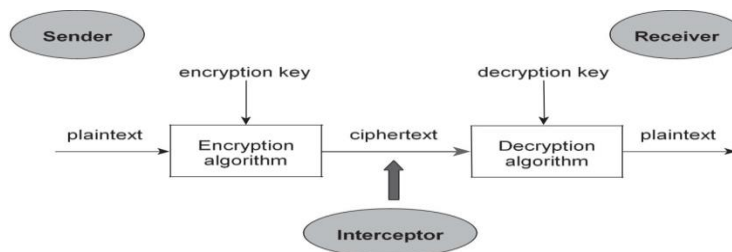


Figure 0.1 simple model of a cryptosystem that provides confidentiality

The illustration shows a sender who wants to transfer some sensitive data to a receiver in such a way that any party intercepting or eavesdropping on the communication channel cannot extract the data.

The objective of this simple cryptosystem is that at the end of the process, only the sender and the receiver will know the plaintext.

2.6.1. Components of a Cryptosystem

The various components of a basic cryptosystem are as follows:

Plaintext. It is the data to be protected during transmission.

Encryption Algorithm. It is a mathematical process that produces a ciphertext for any given plaintext and encryption key. It is a cryptographic algorithm that takes plaintext and an encryption key as input and produces a ciphertext.

Ciphertext. It is the scrambled version of the plaintext produced by the encryption algorithm using a specific the encryption key. The ciphertext is not guarded. It flows on public channel. It can be intercepted or compromised by anyone who has access to the communication channel.

Decryption Algorithm, It is a mathematical process, that produces a unique plaintext for any given ciphertext and decryption key. It is a cryptographic algorithm that takes a ciphertext and a decryption key as input, and outputs a plaintext. The decryption algorithm essentially reverses the encryption algorithm and is thus closely related to it.

Encryption Key. It is a value that is known to the sender. The sender inputs the encryption key into the encryption algorithm along with the plaintext in order to compute the ciphertext.

Decryption Key. It is a value that is known to the receiver. The decryption key is related to the encryption key, but is not always identical to it. The receiver inputs

the decryption key into the decryption algorithm along with the ciphertext in order to compute the plaintext.

For a given cryptosystem, a collection of all possible decryption keys is called a key space.

An interceptor (an attacker) is an unauthorized entity who attempts to determine the plaintext. He can see the ciphertext and may know the decryption algorithm. He, however, must never know the decryption key.

2.6.2. Types of Cryptosystems

Fundamentally, there are two types of cryptosystems based on the manner in which encryption-decryption is carried out in the system:

Symmetric Key Encryption

Asymmetric Key Encryption

The main difference between these cryptosystems is the relationship between the encryption and the decryption key. Logically, in any cryptosystem, both the keys are closely associated. It is practically impossible to decrypt the ciphertext with the key that is unrelated to the encryption key.

2.6.2.1. Symmetric Key Encryption

The encryption process where same keys are used for encrypting and decrypting the information is known as Symmetric Key Encryption.

The study of symmetric cryptosystems is referred to as symmetric cryptography.

Symmetric cryptosystems are also sometimes referred to as secret key cryptosystems.

A few well-known examples of symmetric key encryption methods are:

Data Encryption Standard (DES), Triple- Data Encryption Standard [Triple-DES], International Data Encryption [IDEA].

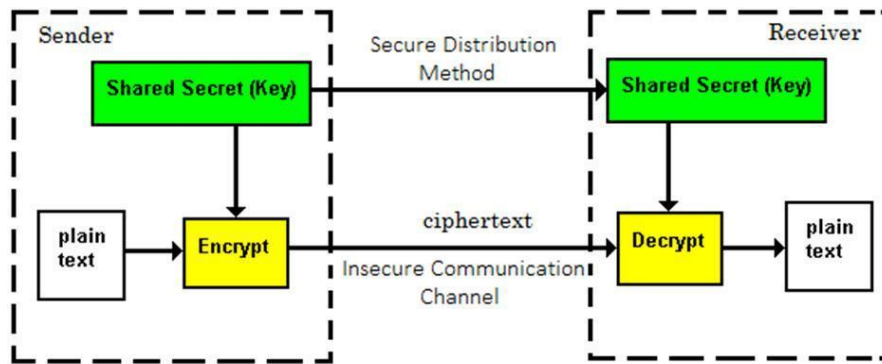


Figure 2.2 symmetric cryptosystems

Prior to 1970, all cryptosystems employed symmetric key encryption. Even today, its relevance is very high and it is being used extensively in many cryptosystems. It is very unlikely that this encryption will fade away, as it has certain advantages over asymmetric key encryption.

The salient features of cryptosystem based on symmetric key encryption are:

Persons using symmetric key encryption must share a common key prior to exchange of information.

Keys are recommended to be changed regularly to prevent any attack on the system.

A robust mechanism needs to exist to exchange the key between the communicating parties. As keys are required to be changed regularly, this mechanism becomes expensive and cumbersome.

In a group of n people, to enable two-party communication between any two persons, the number of keys required for group is $n \times (n - 1)/2$.

Length of Key (number of bits) in this encryption is smaller and hence, process of encryption-decryption is faster than asymmetric key encryption.

Processing power of computer system required to run symmetric algorithm is less.

2.6.2.1.1. Challenge of Symmetric Key Cryptosystem

There are two restrictive challenges of employing symmetric key cryptography.

Key establishment – Before any communication, both the sender and the receiver need to agree on a secret symmetric key. It requires a secure key establishment mechanism in place.

Trust Issue – Since the sender and the receiver use the same symmetric key, there is an implicit requirement that the sender and the receiver ‘trust’ each other. For example, it may happen that the receiver has lost the key to an attacker and the sender is not informed.

These two challenges are highly restraining for modern day communication. Today, people need to exchange information with non-familiar and non-trusted parties. For example, a communication between online seller and customer. These limitations of symmetric key encryption gave rise to asymmetric key encryption schemes.

2.6.2.2. Asymmetric Key Encryption

The encryption process where different keys are used for encrypting and decrypting the information is known as Asymmetric Key Encryption. Though the keys are different, they are mathematically related and hence, retrieving the plaintext by decrypting ciphertext is feasible. The process is depicted in the following illustration

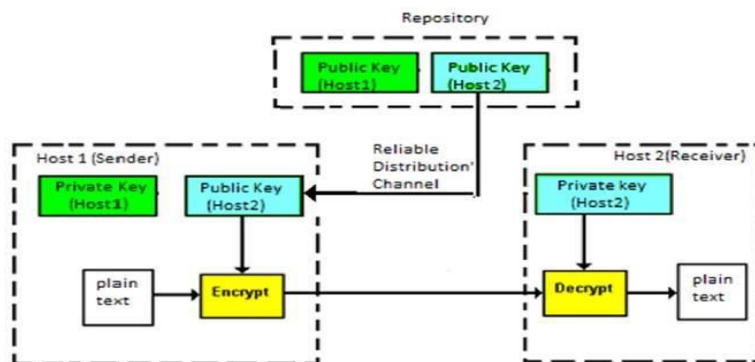


Figure 0.3 asymmetric cryptosystems

2.6.2.2.1 Challenge of Public Key Cryptosystem

Public-key cryptosystems have one significant challenge – the user needs to trust that the public key that he is using in communications with a person really is the public key of that person and has not been spoofed by a malicious third party.

This is usually accomplished through a Public Key Infrastructure (PKI) consisting a trusted third party. The third party securely manages and attests to the authenticity of public keys. When the third party is requested to provide the public key for any communicating person X, they are trusted to provide the correct public key.

The third party satisfies itself about user identity by the process of attestation, notarization, or some other process – that X is the one and only, or globally unique, X. The most common method of making the verified public keys available is to embed them in a certificate which is digitally signed by the trusted third party.

2.6.3. Relation between Encryption Schemes

Table 2.2 basic key properties of two types of cryptosystems

	Symmetric Cryptosystem	Public Key Cryptosyatem
Relation between Keys	Same	Different, but mathematically related
Encryption Key	Symmetric	Public
Decryption Key	Symmetric	Private

Due to the advantages and disadvantage of both the systems, symmetric key and public-key cryptosystems are often used together in the practical information security systems.

2.6.4. Kerckhoff's Principle for Cryptosystem

In the 19th century, a Dutch cryptographer A. Kerckhoff furnished the requirements of a good cryptosystem. Kerckhoff stated that a cryptographic system should be secure even if everything about the system, except the key, is public knowledge.

The second rule is currently known as Kerckhoff principle. It is applied in virtually all the contemporary encryption algorithms such as Data Encryption Standard [DES], Advanced Encryption Standard [AES]. These public algorithms are considered to be thoroughly secure. The security of the encrypted message depends solely on the security of the secret encryption key.

Keeping the algorithms secret may act as a significant barrier to cryptanalysis. However, keeping the algorithms secret is possible only when they are used in a strictly limited circle.

In modern era, cryptography needs to cater to users who are connected to the Internet. In such cases, using a secret algorithm is not feasible, hence Kerckhoff principles became essential guidelines for designing algorithms in modern cryptography.

2.6.5. Attacks On Cryptosystems

In the present era, not only business but almost all the aspects of human life are driven by information. Hence, it has become imperative to protect useful information from malicious activities such as attacks. Let us consider the types of attacks to which information is typically subjected to.

Attacks are typically categorized based on the action performed by the attacker. An attack, thus, can be passive or active.

2.6.5.1. Passive Attacks

The main goal of a passive attack is to obtain unauthorized access to the information. For example, actions such as intercepting and eavesdropping on the communication channel can be regarded as passive attack.

These actions are passive in nature, as they neither affect information nor disrupt the communication channel. A passive attack is often seen as stealing information. The only difference in stealing physical goods and stealing information is that theft of data still leaves the owner in possession of that data. Passive information attack is thus more dangerous than stealing of goods, as information theft may go unnoticed by the owner.

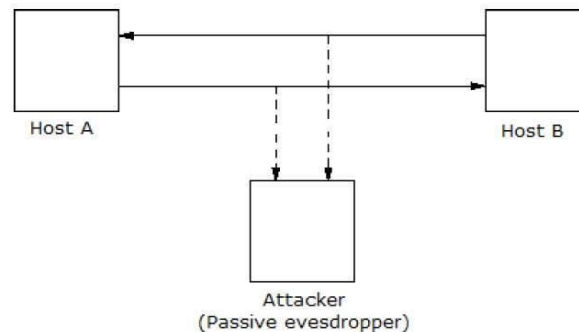


Figure 0.4 a passive attack

2.6.5.2. Active Attacks

An active attack involves changing the information in some way by conducting some process on the information. For example,

Modifying the information in an unauthorized manner.

Initiating unintended or unauthorized transmission of information.

Alteration of authentication data such as originator name or timestamp associated with information

Unauthorized deletion of data.

Denial of access to information for legitimate users (denial of service).

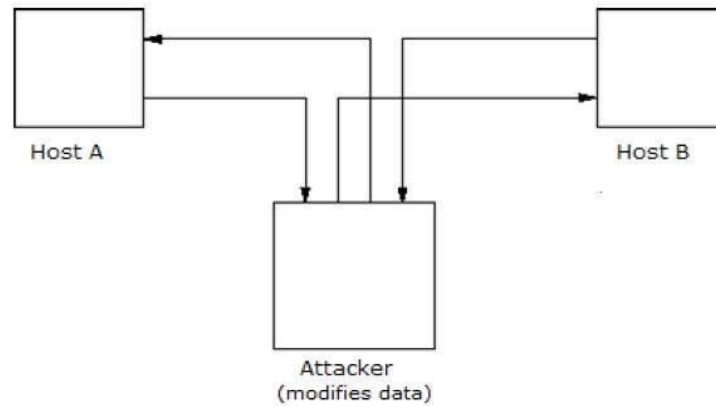


Figure 2.5 active attack

Cryptography provides many tools and techniques for implementing cryptosystems capable of preventing most of the attacks described above.

Environment around Cryptosystem

While considering possible attacks on the cryptosystem, it is necessary to know the cryptosystems environment. The attacker's assumptions and knowledge about the environment decides his capabilities.

In cryptography, the following three assumptions are made about the security environment and attacker's capabilities.

Details of the Encryption Scheme

The design of a cryptosystem is based on the following two cryptography algorithms –

Public Algorithms – With this option, all the details of the algorithm are in the public domain, known to everyone.

Proprietary algorithms – The details of the algorithm are only known by the system designers and users.

In case of proprietary algorithms, security is ensured through obscurity. Private algorithms may not be the strongest algorithms as they are developed in-house and may not be extensively investigated for weakness.

Secondly, they allow communication among closed group only. Hence they are not suitable for modern communication where people communicate with large number of known or unknown entities. Also, according to Kerckhoff's principle, the algorithm is preferred to be public with strength of encryption lying in the key.

Thus, the first assumption about security environment is that the encryption algorithm is known to the attacker.

Availability of Ciphertext

Once the plaintext is encrypted into ciphertext, it is put on unsecure public channel (say email) for transmission. Thus, the attacker can obviously assume that it has access to the ciphertext generated by the cryptosystem.

Availability of Plaintext and Ciphertext

This assumption is not as obvious as other. However, there may be situations where an attacker can have access to plaintext and corresponding ciphertext. Some such possible circumstances are

The attacker influences the sender to convert plaintext of his choice and obtains the ciphertext.

The receiver may divulge the plaintext to the attacker inadvertently. The attacker has access to corresponding ciphertext gathered from open channel.

In a public-key cryptosystem, the encryption key is in open domain and is known to any potential attacker. Using this key, he can generate pairs of corresponding plaintexts and ciphertexts.

2.6.6. Cryptographic Attacks

The basic intention of an attacker is to break a cryptosystem and to find the plaintext from the ciphertext. To obtain the plaintext, the attacker only needs to find out the secret decryption key, as the algorithm is already in public domain.

Hence, he applies maximum effort towards finding out the secret key used in the cryptosystem. Once the attacker is able to determine the key, the attacked system is considered as broken or compromised.

Based on the methodology used, attacks on cryptosystems are categorized as follows –

Ciphertext Only Attacks (COA) – In this method, the attacker has access to a set of ciphertext(s). He does not have access to corresponding plaintext. Ciphertext Only Attacks [COA] is said to be successful when the corresponding plaintext can be determined from a given set of ciphertext. Occasionally, the encryption key can be determined from this attack. Modern cryptosystems are guarded against ciphertext-only attacks.

Known Plaintext Attack (KPA) – In this method, the attacker knows the plaintext for some parts of the ciphertext. The task is to decrypt the rest of the ciphertext using this information. This may be done by determining the key or via some other method. The best example of this attack is linear cryptanalysis against block ciphers.

Chosen Plaintext Attack (CPA) – In this method, the attacker has the text of his choice encrypted. So he has the ciphertext-plaintext pair of his choice. This simplifies his task of determining the encryption key. An example of this attack is differential cryptanalysis applied against block ciphers as well as hash functions. A popular public key cryptosystem, RSA is also vulnerable to chosen-plaintext attacks.

Dictionary Attack – This attack has many variants, all of which involve compiling a ‘dictionary’. In simplest method of this attack, attacker builds a dictionary of ciphertexts and corresponding plaintexts that he has learnt over a period of time. In future, when an attacker gets the ciphertext, he refers the dictionary to find the corresponding plaintext.

Brute Force Attack (BFA) – In this method, the attacker tries to determine the key by attempting all possible keys. If the key is 8 bits long, then the number of possible keys is $2^8 = 256$. The attacker knows the ciphertext and the algorithm, now he attempts all the 256 keys one by one for decryption. The time to complete the attack would be very high if the key is long.

Birthday Attack – This attack is a variant of brute-force technique. It is used against the cryptographic hash function. When students in a class are asked about their birthdays, the answer is one of the possible 365 dates. Let us assume the first student's birthdate is 3rd Aug. Then to find the next student whose birthdate is 3rd Aug, we need to enquire $1.25 * \sqrt{365} \approx 25$ students.

Similarly, if the hash function produces 64 bit hash values, the possible hash values are 1.8×10^{19} . By repeatedly evaluating the function for different inputs, the same output is expected to be obtained after about 5.1×10^9 random inputs.

If the attacker is able to find two different inputs that give the same hash value, it is a collision and that hash function is said to be broken.

Man in Middle Attack (MIM) – The targets of this attack are mostly public key cryptosystems where key exchange is involved before communication takes place.

Host A wants to communicate to host B, hence requests public key of B.

An attacker intercepts this request and sends his public key instead.

Thus, whatever host A sends to host B, the attacker is able to read.

In order to maintain communication, the attacker re-encrypts the data after reading with his public key and sends to B.

The attacker sends his public key as A's public key so that B takes it as if it is taking it from A.

Side Channel Attack (SCA) – This type of attack is not against any particular type of cryptosystem or algorithm. Instead, it is launched to exploit the weakness in physical implementation of the cryptosystem.

Timing Attacks – They exploit the fact that different computations take different times to compute on processor. By measuring such timings, it is possible to know about a particular computation the processor is carrying out. For example, if the encryption takes a longer time, it indicates that the secret key is long.

Power Analysis Attacks – These attacks are similar to timing attacks except that the amount of power consumption is used to obtain information about the nature of the underlying computations.

Fault analysis Attacks – In these attacks, errors are induced in the cryptosystem and the attacker studies the resulting output for useful information.

2.6.7. Practicality of Attacks

The attacks on cryptosystems described here are highly academic, as majority of them come from the academic community. In fact, many academic attacks involve quite unrealistic assumptions about environment as well as the capabilities of the attacker. For example, in chosen-ciphertext attack, the attacker requires an impractical number of deliberately chosen plaintext-ciphertext pairs. It may not be practical altogether.

Nonetheless, the fact that any attack exists should be a cause of concern, particularly if the attack technique has the potential for improvement.

2.6.8. Modern Symmetric Key Encryption

Digital data is represented in strings of binary digits (bits) unlike alphabets. Modern cryptosystems need to process this binary strings to convert in to another binary string. Based on how these binary strings are processed, a symmetric encryption schemes can be classified in to –

2.6.8.1. Block Ciphers

In this scheme, the plain binary text is processed in blocks (groups) of bits at a time; i.e. a block of plaintext bits is selected, a series of operations is performed on this block to generate a block of ciphertext bits. The number of bits in a block is

fixed. For example, the schemes DES and AES have block sizes of 64 and 128, respectively.

2.6.8.2. Stream Ciphers

In this scheme, the plaintext is processed one bit at a time i.e. one bit of plaintext is taken, and a series of operations is performed on it to generate one bit of ciphertext. Technically, stream ciphers are block ciphers with a block size of one bit.

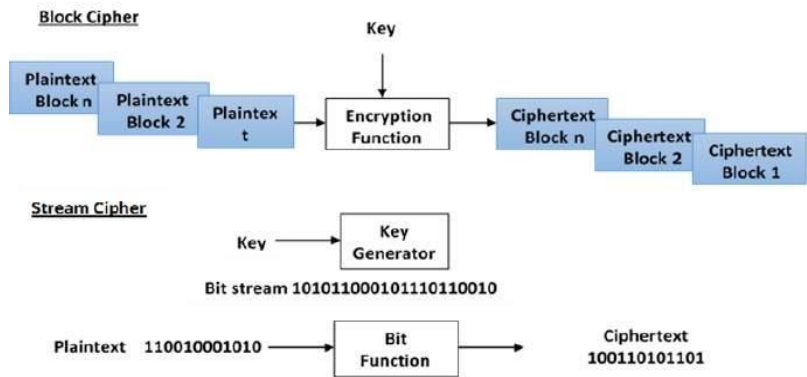


Figure 0.6 Block & Stream Cipher

2.6.9.1.1. Block Cipher

The basic scheme of a block cipher is depicted as follows –

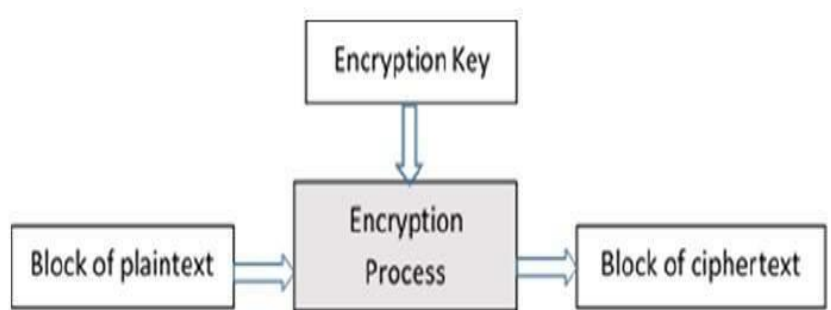


Figure 0.7 Block Cipher

A block cipher takes a block of plaintext bits and generates a block of ciphertext bits, generally of same size. The size of block is fixed in the given scheme. The

choice of block size does not directly affect to the strength of encryption scheme. The strength of cipher depends up on the key length.

2.6.9.1.2. Block Size

Though any size of block is acceptable, following aspects are borne in mind while selecting a size of a block.

Avoid very small block size – Say a block size is m bits. Then the possible plaintext bits combinations are then 2^m . If the attacker discovers the plain text blocks corresponding to some previously sent ciphertext blocks, then the attacker can launch a type of ‘dictionary attack’ by building up a dictionary of plaintext/ciphertext pairs sent using that encryption key. A larger block size makes attack harder as the dictionary needs to be larger.

Do not have very large block size – With very large block size, the cipher becomes inefficient to operate. Such plaintexts will need to be padded before being encrypted.

Multiples of 8 bit – A preferred block size is a multiple of 8 as it is easy for implementation as most computer processor handle data in multiple of 8 bits.

2.6.9.1.3. Padding in Block Cipher

Block ciphers process blocks of fixed sizes (say 64 bits). The length of plaintexts is mostly not a multiple of the block size. For example, a 150-bit plaintext provides two blocks of 64 bits each with third block of balance 22 bits. The last block of bits needs to be padded up with redundant information so that the length of the final block equal to block size of the scheme. In our example, the remaining 22 bits need to have additional 42 redundant bits added to provide a complete block. The process of adding bits to the last block is referred to as padding.

Too much padding makes the system inefficient. Also, padding may render the system insecure at times, if the padding is done with same bits always.

2.6.9.1.4. Block Cipher Schemes

There is a vast number of block ciphers schemes that are in use. Many of them are publically known. Most popular and prominent block ciphers are listed below.

Data Encryption Standard [DES] – The popular block cipher of the 1990s. It is now considered as a ‘broken’ block cipher, due primarily to its small key size.

Triple-Data Encryption Standard [3-DES] – It is a variant scheme based on repeated Data Encryption Standard [DES] applications. It is still a respected block ciphers but inefficient compared to the new faster block ciphers available.

2.6.9.1.5. Feistel Block Cipher

Feistel Cipher is not a specific scheme of block cipher. It is a design model from which many different block ciphers are derived. Data Encryption Standard [DES] is just one example of a Feistel Cipher. A cryptographic system based on Feistel cipher structure uses the same algorithm for both encryption and decryption.

2.6.9.1.6. Encryption Process

The encryption process uses the Feistel structure consisting multiple rounds of processing of the plaintext, each round consisting of a “substitution” step followed by a permutation step.

Feistel Structure is shown in the following illustration –

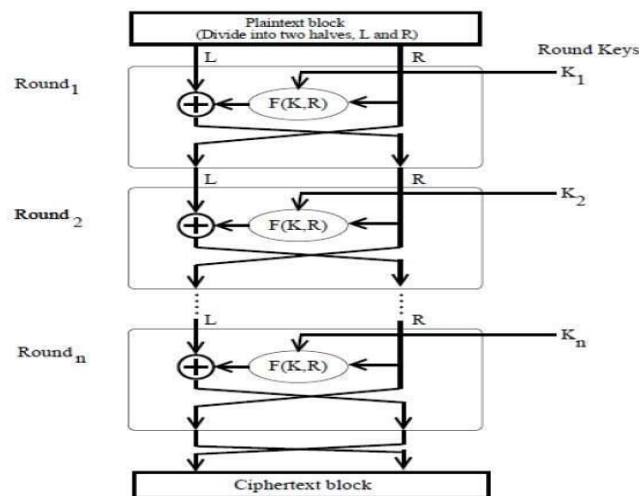


Figure 0.8 Feistel structure

The input block to each round is divided into two halves that can be denoted as L and R for the left half and the right half.

In each round, the right half of the block, R, goes through unchanged. But the left half, L, goes through an operation that depends on R and the encryption key. First, we apply an encrypting function 'f' that takes two input – the key K and R. The function produces the output $f(R,K)$. Then, we XOR the output of the mathematical function with L.

In real implementation of the Feistel Cipher, such as Data Encryption Standard [DES], instead of using the whole encryption key during each round, a round-dependent key (a subkey) is derived from the encryption key. This means that each round uses a different key, although all these subkeys are related to the original key.

The permutation step at the end of each round swaps the modified L and unmodified R. Therefore, the L for the next round would be R of the current round. And R for the next round be the output L of the current round.

Above substitution and permutation steps form a 'round'. The number of rounds are specified by the algorithm design.

Once the last round is completed then the two sub blocks, 'R' and 'L' are concatenated in this order to form the ciphertext block.

The difficult part of designing a Feistel Cipher is selection of round function 'f'. In order to be unbreakable scheme, this function needs to have several important properties that are beyond the scope of our discussion.

2.6.9.1.7. Decryption Process

The process of decryption in Feistel cipher is almost similar. Instead of starting with a block of plaintext, the ciphertext block is fed into the start of the Feistel structure and then the process thereafter is exactly the same as described in the given illustration.

The process is said to be almost similar and not exactly same. In the case of decryption, the only difference is that the subkeys used in encryption are used in the reverse order.

The final swapping of 'L' and 'R' in last step of the Feistel Cipher is essential. If these are not swapped then the resulting ciphertext could not be decrypted using the same algorithm.

2.6.9.1.8. Number of Rounds

The number of rounds used in a Feistel Cipher depends on desired security from the system. More number of rounds provide more secure system. But at the same time, more rounds mean the inefficient slow encryption and decryption processes. Numbers of rounds in the systems thus depend upon efficiency–security tradeoff.

2.6.9.1.9. Data Encryption Standard

The Data Encryption Standard (DES) is a symmetric-key block cipher published by the National Institute of Standards and Technology (NIST).

DES is an implementation of a Feistel Cipher. It uses 16 round Feistel structure. The block size is 64-bit. Though, key length is 64-bit, DES has an effective key length of 56 bits, since 8 of the 64 bits of the key are not used by the encryption algorithm (function as check bits only).

2.6.9.1.9.1. General Structure of DES .

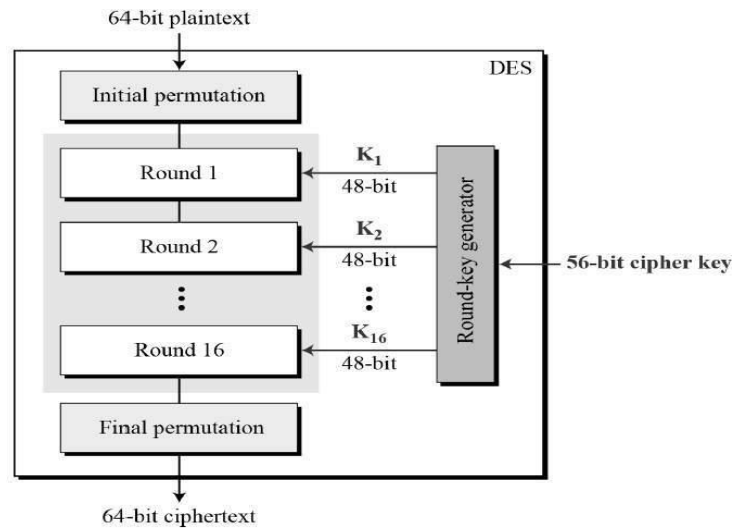


Figure 0.9 Key schedule

Since DES is based on the Feistel Cipher, all that is required to specify DES is –

Round function

Key schedule

Any additional processing – Initial and final permutation

Initial and Final Permutation

The initial and final permutations are straight Permutation boxes (P-boxes) that are inverses of each other. They have no cryptography significance in Data Encryption Standard [DES]. The initial and final permutations are shown as follows

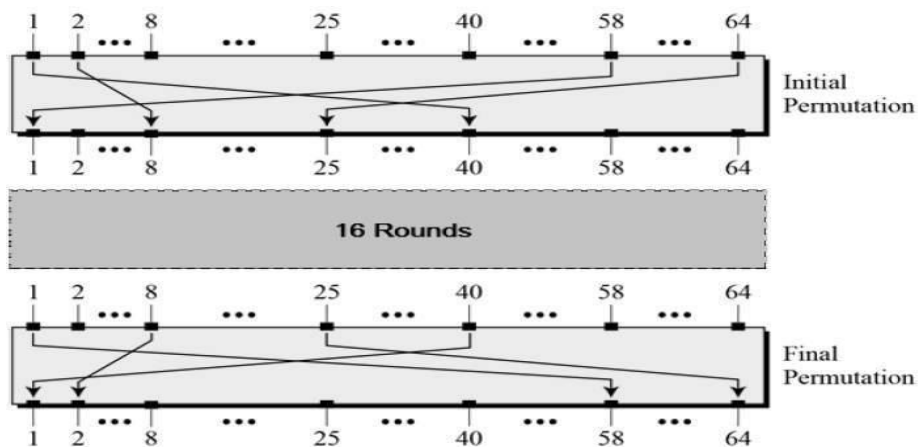


Figure 0.10 Round Function

Round Function

The heart of this cipher is the Data Encryption Standard [DES] function, f . The Data Encryption Standard function applies a 48-bit key to the rightmost 32 bits to produce a 32-bit output.

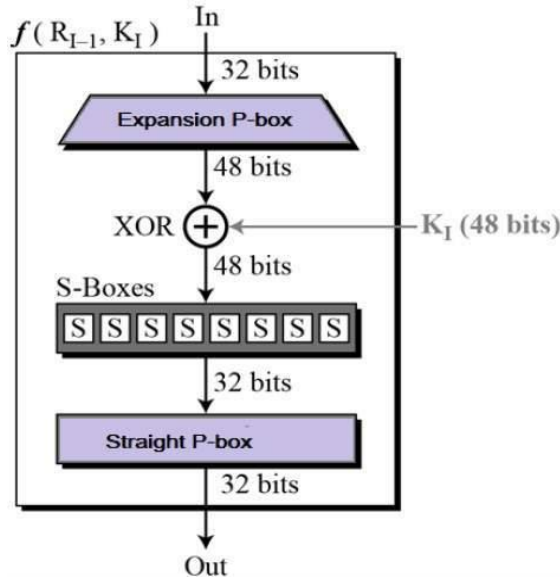


Figure 2. 11 Expansion Permutation Box

Expansion Permutation Box – Since right input is 32-bit and round key is a 48-bit, we first need to expand right input to 48 bits. Permutation logic is graphically depicted in the following illustration

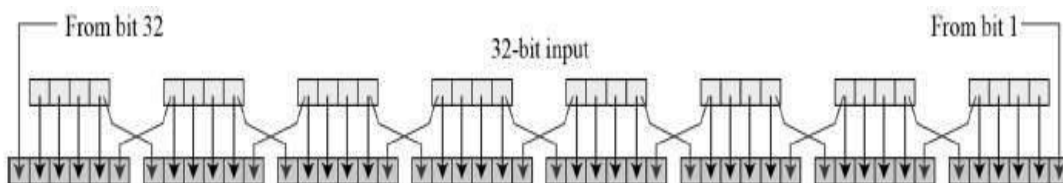


Figure 0.12 Permutation logic

The graphically depicted permutation logic is generally described as table in Data Encryption Standard specification illustrated as shown

32	01	02	03	04	05
04	05	06	07	08	09
08	09	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	31	31	32	01

Figure 0.13 permutation logic described as table

XOR (Whitener). – After the expansion permutation, Data Encryption Standard [DES] does XOR operation on the expanded right section and the round key. The round key is used only in this operation.

Substitution Boxes. – The S-boxes carry out the real mixing (confusion). Encryption Standard uses 8 S-boxes, each with a 6-bit input and a 4-bit output. Refer the following illustration –

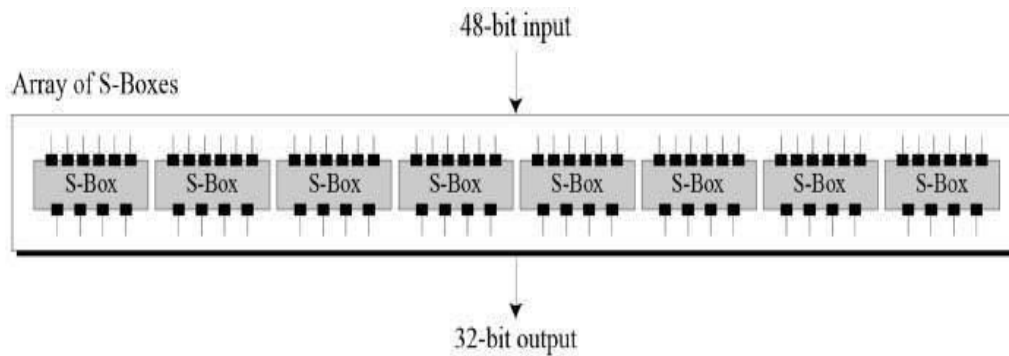


Figure 0.14 Array of S-Boxes

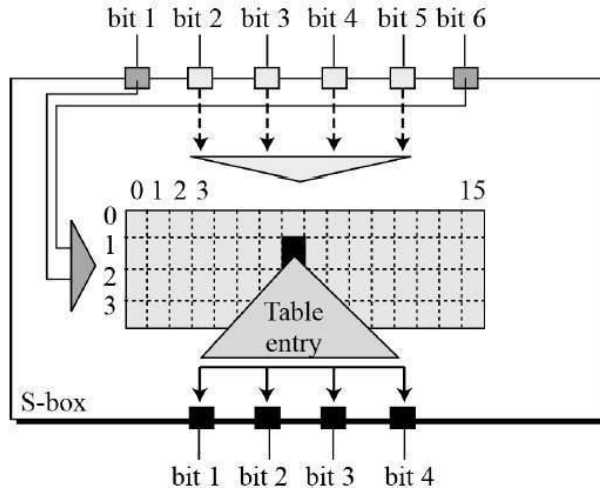


Figure 0.15 S-box rule

There are a total of eight S-box tables. The output of all eight s-boxes is then combined in to 32 bit section.

Straight Permutation – The 32 bit output of S-boxes is then subjected to the straight permutation with rule shown in the following illustration:

16	07	20	21	29	12	28	17
01	15	23	26	05	18	31	10
02	08	24	14	32	27	03	09
19	13	30	06	22	11	04	25

Figure 0.16 S-box table

Key Generation

The round-key generator creates sixteen 48-bit keys out of a 56-bit cipher key. The process of key generation is depicted in the following illustration

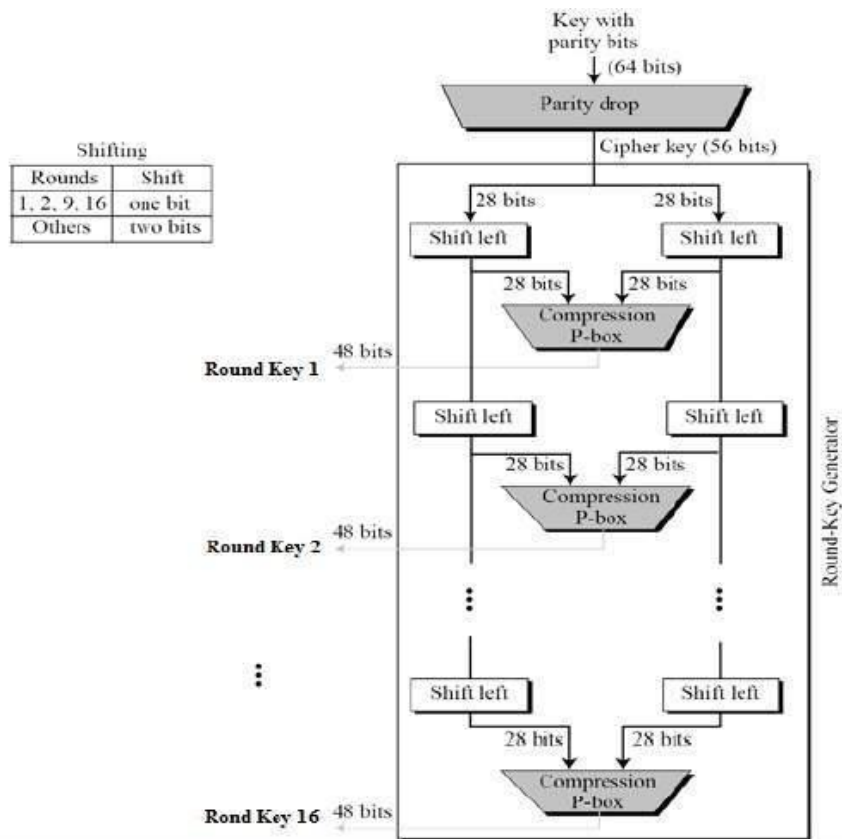


Figure 0.17 Key Generation

The logic for Parity drop, shifting, and Compression P-box is given in the Data Encryption Standard description.

2.6.9.1.9.2.DES Analysis

The Data Encryption Standard [DES] satisfies both the desired properties of block cipher. These two properties make cipher very strong.

Avalanche effect – A small change in plaintext results in the very great change in the ciphertext.

Completeness – each bit of ciphertext depends on many bits of plaintext.

During the last few years, cryptanalysis has found some weaknesses in Data Encryption Standard when key selected are weak keys. These keys shall be avoided.

Data Encryption Standard has proved to be a very well designed block cipher. There have been no significant cryptanalytic attacks on Data Encryption Standard other than exhaustive key search.

2.6.9. Block Cipher Modes of Operation

In this chapter, we will discuss the different modes of operation of a block cipher. These are procedural rules for a generic block cipher. Interestingly, the different modes result in different properties being achieved which add to the security of the underlying block cipher.

A block cipher processes the data blocks of fixed size. Usually, the size of a message is larger than the block size. Hence, the long message is divided into a series of sequential message blocks, and the cipher operates on these blocks one at a time.

2.6.9.1. Electronic Code Book (ECB) Mode

This mode is a most straightforward way of processing a series of sequentially listed message blocks.

Operation

The user takes the first block of plaintext and encrypts it with the key to produce the first block of ciphertext. then takes the second block of plaintext and follows the same process with same key and so on so forth.

The Electronic Codebook [ECB] mode is deterministic, that is, if plaintext block P_1, P_2, \dots, P_m are encrypted twice under the same key, the output ciphertext blocks will be the same.

In fact, for a given key technically we can create a codebook of ciphertexts for all possible plaintext blocks. Encryption would then entail only looking up for required plaintext and select the corresponding ciphertext. Thus, the operation is analogous to the assignment of code words in a codebook, and hence gets an

official name – Electronic Codebook mode of operation (ECB). It is illustrated as follows –

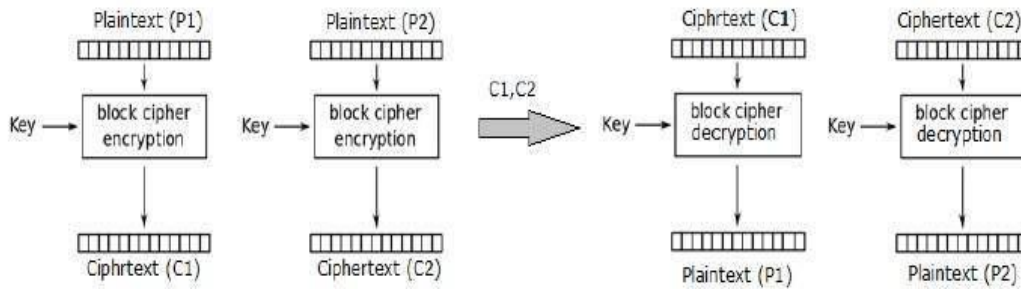


Figure2.18 Electronic Codebook mode

Analysis of ECB Mode

In reality, any application data usually have partial information which can be guessed. For example, the range of salary can be guessed. A ciphertext from Electronic Codebook can allow an attacker to guess the plaintext by trial-and-error if the plaintext message is within predictable.

For example, if a ciphertext from the Electronic Codebook [ECB] mode is known to encrypt a salary figure, then a small number of trials will allow an attacker to recover the figure. In general, we do not wish to use a deterministic cipher, and hence the Electronic Codebook mode should not be used in most applications.

2.6.9.2. Cipher Block Chaining (CBC) Mode

Cipher Block Chaining [CBC] mode of operation provides message dependence for generating ciphertext and makes the system non-deterministic.

Operation

The operation of Cipher Block Chaining mode is depicted in the following illustration. The steps are as follows –

Load the n-bit Initialization Vector (IV) in the top register.

XOR the n-bit plaintext block with data value in top register.

Encrypt the result of XOR operation with underlying block cipher with key K.

Feed ciphertext block into top register and continue the operation till all plaintext blocks are processed.

For decryption, Initialization Vector data is XORed with first ciphertext block decrypted. The first ciphertext block is also fed into to register replacing Initialization Vector for decrypting next ciphertext block.

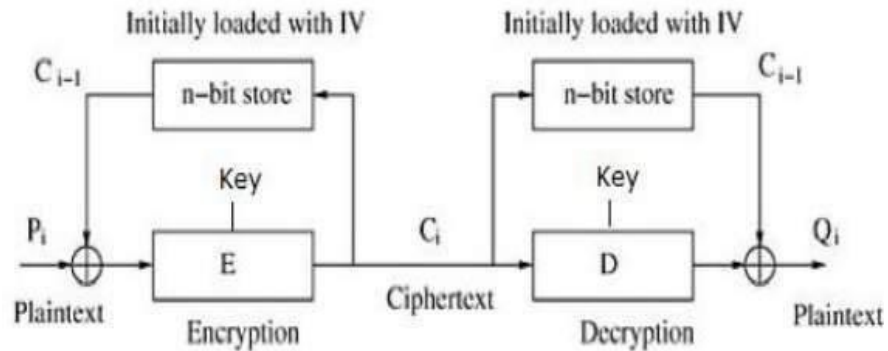


Figure 0.19 Cipher Block Chaining

Analysis of CBC Mode

In Cipher Block Chaining [CBC] mode, the current plaintext block is added to the previous ciphertext block, and then the result is encrypted with the key. Decryption is thus the reverse process, which involves decrypting the current ciphertext and then adding the previous ciphertext block to the result.

Advantage of Cipher Block Chaining over Electronic Codebook is that changing Initialization Vector [IV] results in different ciphertext for identical message. On the drawback side, the error in transmission gets propagated to few further block during decryption due to chaining effect.

It is worth mentioning that Cipher Block Chaining mode forms the basis for a well-known data origin authentication mechanism. Thus, it has an advantage for those applications that require both symmetric encryption and data origin authentication.

2.6.9.3. Cipher Feedback (CFB) Mode

In this mode, each ciphertext block gets 'fed back' into the encryption process in order to encrypt the next plaintext block.

Operation

The operation of Cipher Feedback [CFB] mode is depicted in the following illustration. For example, in the present system, a message block has a size 's' bits where $1 < s < n$. The Cipher Feedback mode requires an initialization vector (IV) as the initial random n-bit input block. The initialization vector need not be secret.

Steps of operation are –

Load the IV in the top register.

Encrypt the data value in top register with underlying block cipher with key K.

Take only 's' number of most significant bits (left bits) of output of encryption process and XOR them with 's' bit plaintext message block to generate ciphertext block.

Feed ciphertext block into top register by shifting already present data to the left and continue the operation till all plaintext blocks are processed.

Essentially, the previous ciphertext block is encrypted with the key, and then the result is XORed to the current plaintext block.

Similar steps are followed for decryption. Pre-decided initialization vector is initially loaded at the start of decryption.

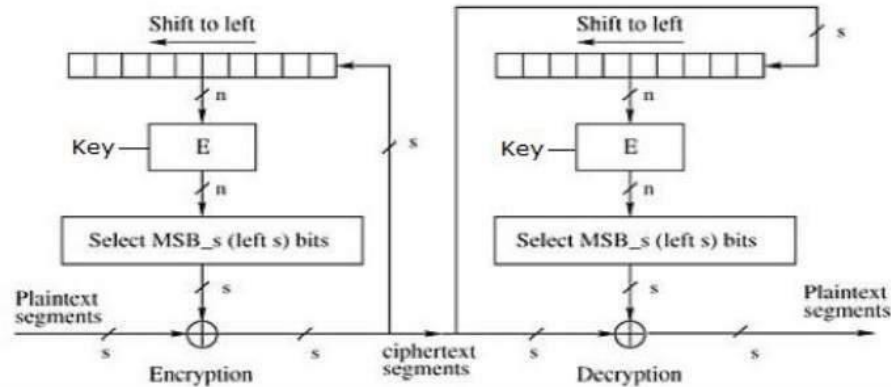


Figure 0.20 Cipher Feedback

Analysis of CFB Mode

Cipher Feedback [CFB] mode differs significantly from Electronic Codebook [ECB] mode, the ciphertext corresponding to a given plaintext block depends not just on that plaintext block and the key, but also on the previous ciphertext block. In other words, the ciphertext block is dependent of message.

Cipher Feedback has a very strange feature. In this mode, user decrypts the ciphertext using only the encryption process of the block cipher. The decryption algorithm of the underlying block cipher is never used.

Apparently, Cipher Feedback mode is converting a block cipher into a type of stream cipher. The encryption algorithm is used as a key-stream generator to produce key-stream that is placed in the bottom register. This key stream is then XORed with the plaintext as in case of stream cipher.

By converting a block cipher into a stream cipher, Cipher Feedback mode provides some of the advantageous properties of a stream cipher while retaining the advantageous properties of a block cipher.

On the flip side, the error of transmission gets propagated due to changing of blocks.

2.6.9.4. Output Feedback (OFB) Mode

It involves feeding the successive output blocks from the underlying block cipher back to it. These feedback blocks provide string of bits to feed the encryption algorithm which act as the key-stream generator as in case of Cipher Feedback mode.

The key stream generated is XOR-ed with the plaintext blocks. The Output Feedback [OFB] mode requires an initialization vector [IV] as the initial random n-bit input block. The initialization vector need not be secret.

The operation is depicted in the following illustration

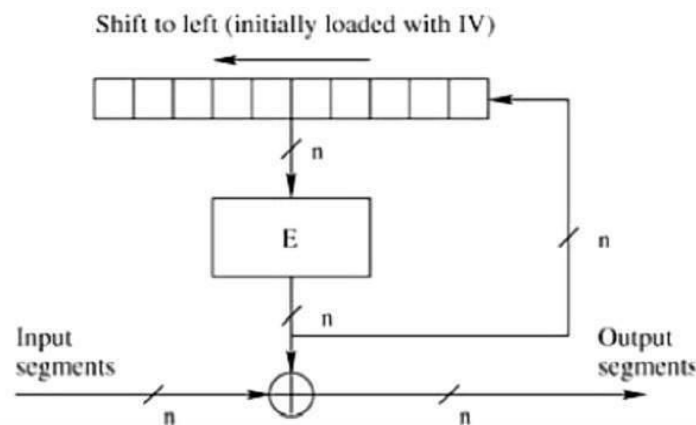


Figure 0.21 Output Feedback

2.6.9.5. Counter (CTR) Mode

It can be considered as a counter-based version of Cipher Feedback [CFB] mode without the feedback. In this mode, both the sender and receiver need to access to a reliable counter, which computes a new shared value each time a ciphertext block is exchanged. This shared counter is not necessarily a secret value, but challenge is that both sides must keep the counter synchronized.

Operation

Both encryption and decryption in Counter [CTR] mode are depicted in the following illustration. Steps in operation are –

Load the initial counter value in the top register is the same for both the sender and the receiver. It plays the same role as the IV in Cipher Feedback and Cipher Block Chaining [CBC] mode.

Encrypt the contents of the counter with the key and place the result in the bottom register.

Take the first plaintext block P1 and XOR this to the contents of the bottom register. The result of this is C1. Send C1 to the receiver and update the counter.

The counter update replaces the ciphertext feedback in Cipher Feedback mode.

Continue in this manner until the last plaintext block has been encrypted.

The decryption is the reverse process. The ciphertext block is XORed with the output of encrypted contents of counter value. After decryption of each ciphertext block counter is updated as in case of encryption.

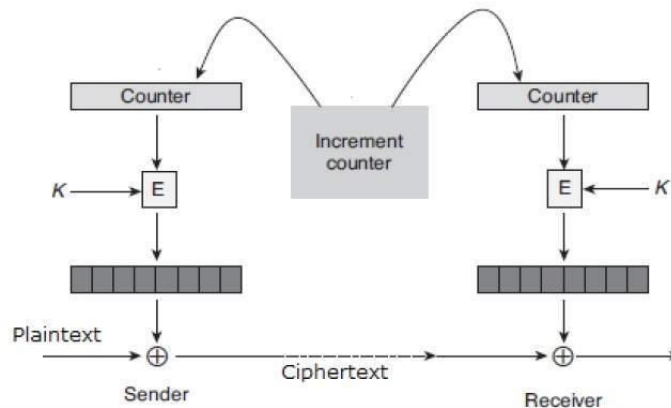


Figure 0.22 Counter

Analysis of Counter Mode

It does not have message dependency and hence a ciphertext block does not depend on the previous plaintext blocks.

Like Cipher Feedback [CFB] mode, Counter [CTR] mode does not involve the decryption process of the block cipher. This is because the Counter mode is really using the block cipher to generate a key-stream, which is encrypted using the XOR

function. In other words, Counter mode also converts a block cipher to a stream cipher.

The serious disadvantage of Counter mode is that it requires a synchronous counter at sender and receiver. Loss of synchronization leads to incorrect recovery of plaintext.

However, Counter mode has almost all advantages of Cipher Feedback mode. In addition, it does not propagate error of transmission at all.

2.6.10. Related Works

Lai and Massey [4] introduced a proposal for a new block encryption standard named International Data Encryption [IDEA] which is based on add–rotate–xor [ARX] design with three different group operations modular addition, bitwise XOR and modular multiplication to achieve strong confusion and diffusion.

International Data Encryption is vulnerable due the problem of weak keys. Biryukov et al. [5] introduced new weak key classes of International Data Encryption. In 2007, Biham et al. [6] presented a new attack against 6-round (reduce) ID International Data Encryption EA. Khovratovich et al.[7] introduced narrow-bicliques cryptanalysis of full International Data Encryption which is a type of meet in the middle attack.

Gonzalo et al. [8] introduced a new block cipher algorithm named Akelarre which is also based on add–rotate–xor [ARX] design and its overall structure is same as of International Data Encryption instead of 16 bit sub-block, fixed length key and fixed number of rounds as in International Data Encryption, it uses 32 bit sub-block, variable key length and variable number of rounds. It used data dependent rotations instead of modular multiplication.

In 1997, Ferguson and Schneier [9] presented the cryptanalysis of Akelarre in which they conclude that Akelarre is disappointingly weak. They have shown that

the round function of Akelarre preserve the parity of input and insecure against chosen plain text attack. They also conclude that the 31- bits information about master key can be recover using trivial methods from improved key schedule of a new version of Akelarre [10].

Hong et al. [11] introduced a 128-bit block cipher for fast encryption on common processor named LEA which is based on add–rotate–xor [ARX] design technique. Its key schedule uses several constants with modular addition and rotations to generate round keys. Link Encryption Algorithm [LEA] is iterative in nature have different number of encryption rounds for different block size. It is faster than Data Encryption Standard [DES], Advance Encryption Standard [AES] and various existing algorithms.

Serpent is an SP-network with input and output block sizes of 128 bits. These are processed through 32 rounds, in each of which we first add 128 bits of key material, then pass the text through 32 S-boxes of 4bits width, then perform a linear transformation that takes each output of one round to the inputs of a number of S-boxes in the next round. Rather than each input bit in one round coming from a single output bit in the last, it is the exclusive-or of between two and seven of them. This means that a change in an input bit propagates rapidly through the cipher—a so-called avalanche effect that makes both linear and differential attacks harder. After the final round, a further 128 bits of key material are added to give the ciphertext. The 33 times 128 bits of key material required are computed from a usersupplied key of up to 256 bits. This is a real cipher using the structure of Figure 5.10, but modified to be “wide” enough and to have enough rounds. The S-boxes are chosen to make linear and differential analysis hard; they have fairly tight bounds on the maximum linear correlation between input and output bits, and on the maximum effect of toggling patterns of input bits. Each of the 32 S-boxes in

a given round is the same; this means that bit-slicing techniques can be used to give a very efficient software implementation on 32-bit processors. Its simple structure makes Serpent easy to analyze, and it can be shown that it withstands all the currently known attacks. (A full specification of Serpent is given in [12] and can be downloaded, together with implementations in a number of languages, from [13].)

CHAPTER III

Methodology

3.1. Introduction

Of known algorithms, the Data Encryption Standard is one of the most widely used algorithms in the world. It uses a cryptographic key consisting of 56 binary numbers, and because it is only broken, it has been developed into Triple- Data Encryption Standard [DES], the International Data Encryption Algorithm (IDEA) is one of the fastest and most current data encryption algorithms in existence. It uses a 128-bit binary key. The most common algorithm used is the Message-Digest Algorithm [MD5] algorithm, which is used in data encryption, password, operating systems and data bases. SP Invented the credibility of any credible data that we know the result of data encryption Is reached true either wrong.

The Rivest cipher 6 [RC6] is the latest algorithm designed by Ronald Rivest. The main goal of this algorithm was to achieve full compatibility with Advanced Encryption Standard [AES] requirements.

As with the Rivest cipher 5 [RC5] algorithm, Rivest cipher 6 [RC6] depends on the variables, but it should be noted that the maximum number of encryption key length is 2040 bits.

Table 3.1 Comparison of DES, Triple DES, AES and Blow Fish algorithm

	Symmetric Encryption Algorithms			
	DES	TDES	AES	BLOWFISH
Block Size	64 bit	64 bit	128 bit	64 bit
Key Size	56 bit	168 bit	128,192,256 bit	32-448 bit

Algorithm Structure	Fiestel Network	Fiestel Network	Substitution Permutation Network	Fiestel Network
Rounds	16	48	9,11,13	16
Attacks	Brute Force Attack	Theoretically Possible	Side Channel Attacks	Not Yet

3.2. Proposed Work

In this section a new symmetric key encryption algorithm has been designed. The algorithm use input data 512bit and key of 256bit length to encrypt the data. It use the S-boxes from Data Encryption Standard, which have been studied intensely for many years and whose properties are thus well understood, in a new structure optimized for efficient implementation on modern processors while simultaneously allowing us to apply the extensive analysis already done on Data Encryption Standard. And festial network as design model.

3.3. How The Algorithm Works

3.3.1. Steps of an algorithm to perform encryption:

Input the original text or plain text and store it.

Input the master key and store it.

Send the master key to s-box and get encryption sub keys for each round.

Split plain text to left and right.

For $L_{i+1} = R_i$, $R_{i+1} = L_i + F(R_i, K_i)$.

For the number of round minimum is 8 to each block.

We get (R_{n+1}, L_{n+1}) and that is cipher text .

3.3.2. Steps of an algorithm to perform decryption:

Input the cipher text and store it.

Input the master key and store it.

Send the master key to s-box and get decryption sub keys for each round.
 Split cipher text to left and right.

$$\text{For } R_i = L_{i+1}, L_i = R_{i+1} + F(L_{i+1}, K_i).$$

We get (L0, R0) and that is plain text.

3.4. Algorithm structure

A diagram showing the general structure of the algorithm

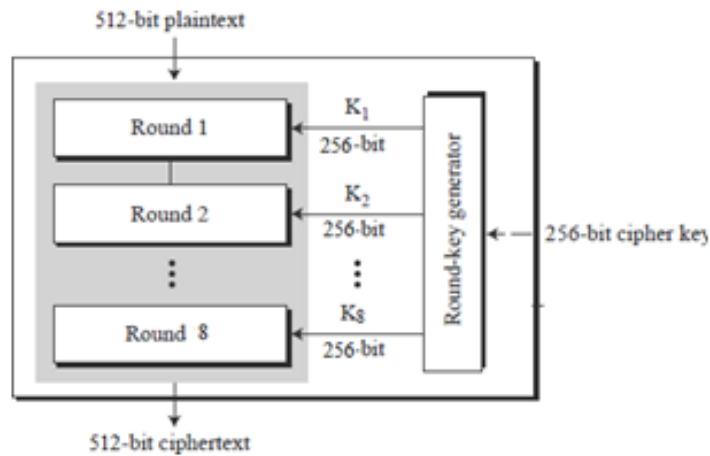


Figure 0.1 General Structure

A diagram showing the encryption process in one round

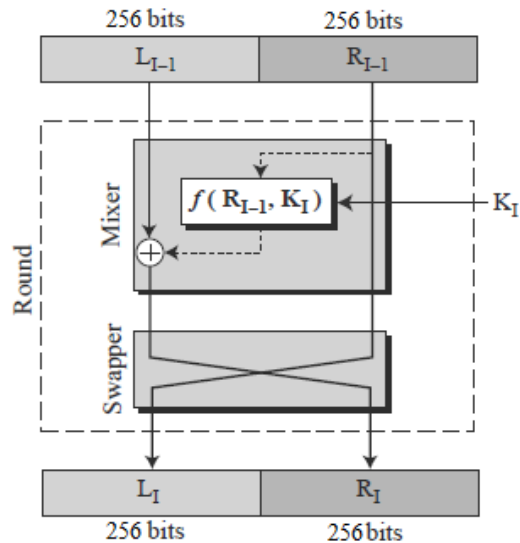


Figure 0.2 Single Round

A diagram showing the process of generating sub keys

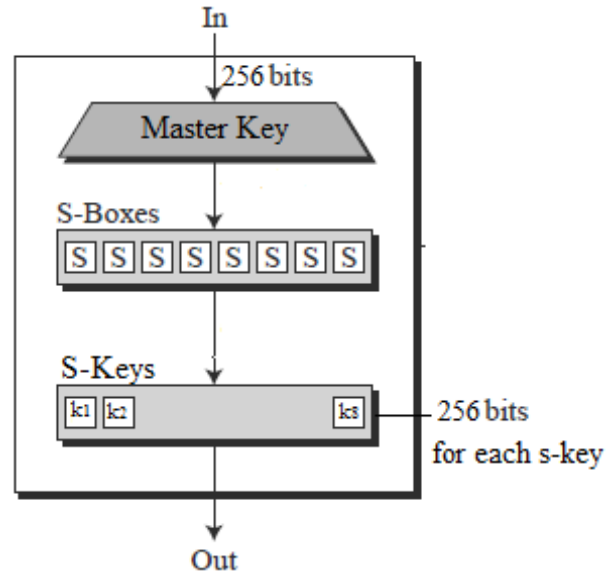


Figure 3.3 Generate Sub Key

A diagram showing how the function works

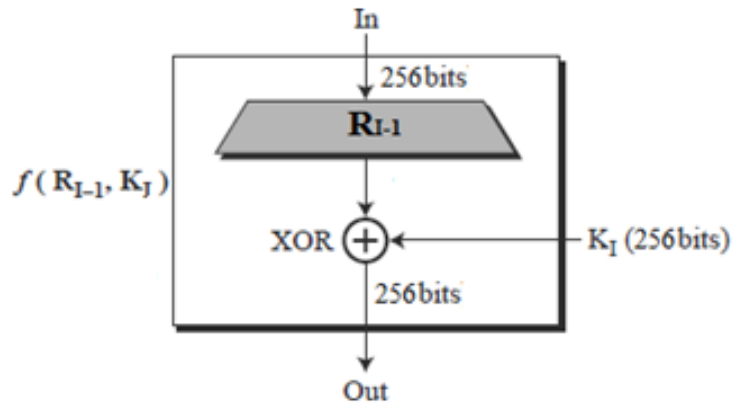


Figure 0.4 Function proceed XOR to right block with subkey

A diagram showing the encryption and decryption process

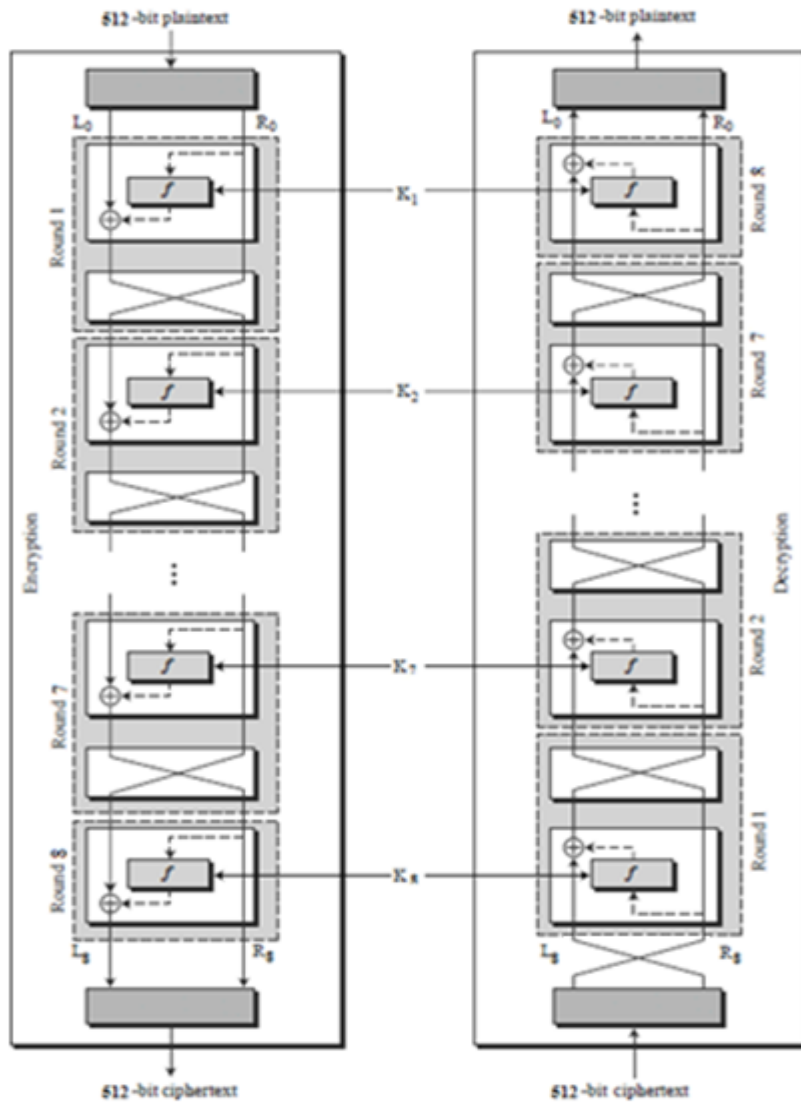


Figure 0.5 Cipher And Reverse Cipher For The First Approach

CHAPTER IV

Implementation and results

4.1. INTRODUCTION

This chapter contains the result of the implementation of the algorithm in addition to the tests and compares the results with the Data Encryption Standard [DES] algorithm, The algorithm was written using Java and through the NetBeans integrated development environment [IDE].

4.2. IMPLEMENTATION

The image below shows the output of the encryption process

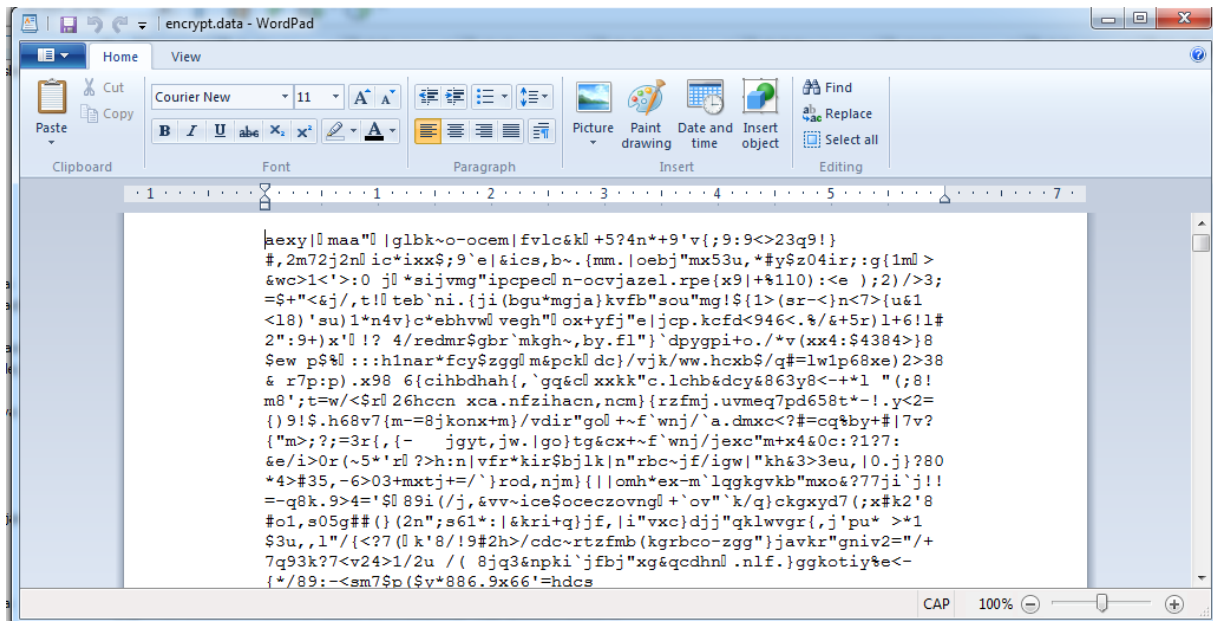


Figure 0.1 Encryption Result

The image below shows the output of the decryption process

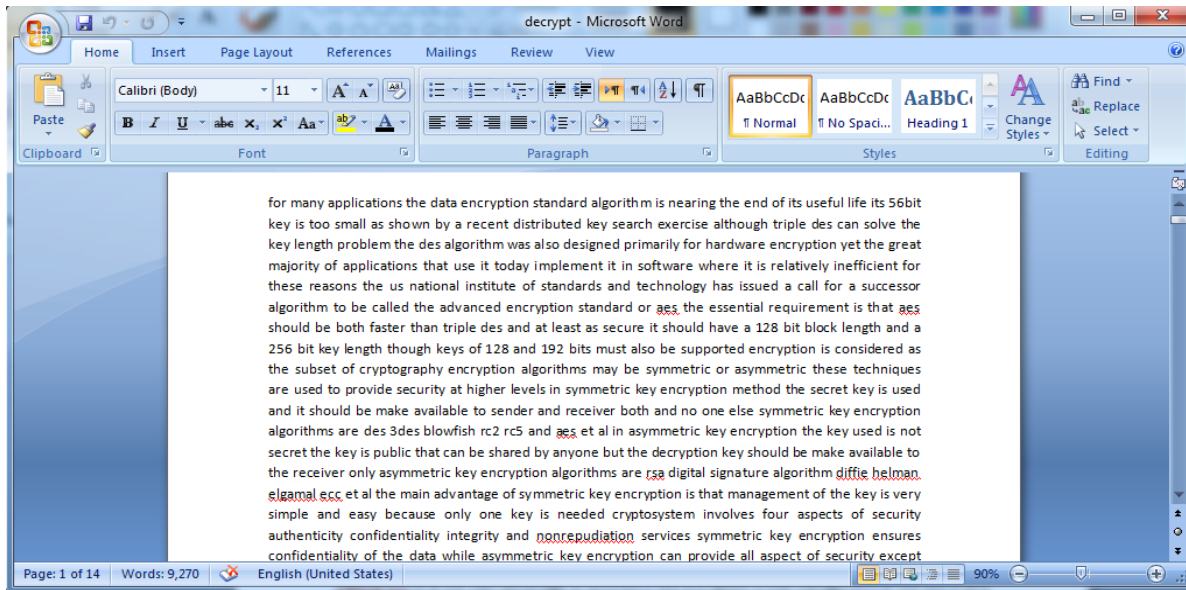


Figure 0.2. Decryption Result

4.3. RESULT

Compared between (Data Encryption Standard - E-DES) to encrypt and decrypt the same file compared to the time of encryption and decryption and obtained the following results

Table 4-1 Compare Between Data Encryption Standard, E-DES, AES In Encryption And Decryption Speed

Algorithm	File type	File size	Encryption speed	Decryption speed
DES	docx	13kb	6s	5s
E-DES	docx	13kb	5s	2s

The comparison between (Data Encryption Standard - E-DES) by using the tool National Institute of Standards and Technology [NIST] and (CRYPTOOL) was obtained and the following results were obtained

Table 0-2 Compare Between Data Encryption Standard, E-DES In Randomness Test

Algorithm	Frequency (mono-bit) test [cryptool]	Black test [National Institute of Standards and Technology]	Run test [National Institute of Standards and Technology]
DES	FAILED	PASS	FAILED
E-DES	PASS	PASS	PASS

4.4. Test And Discussion

The tests below were applied to Data Encryption Standard [DES] and (E-DES) explaining each test

Test on E-DES algorithm

4.4.1. The Frequency Test on E-DES

Consider the pattern 0100 0000 0000 0010. The pattern doesn't appear to be random because there are too many 0s. The most fundamental test for randomness is a frequency test. If a pattern is generated randomly, you'd expect the number of 0s and 1s to be roughly the same. Too many 0s or too many 1s suggest non-randomness.

Frequency Test failed on (E-DES) using NIST tools

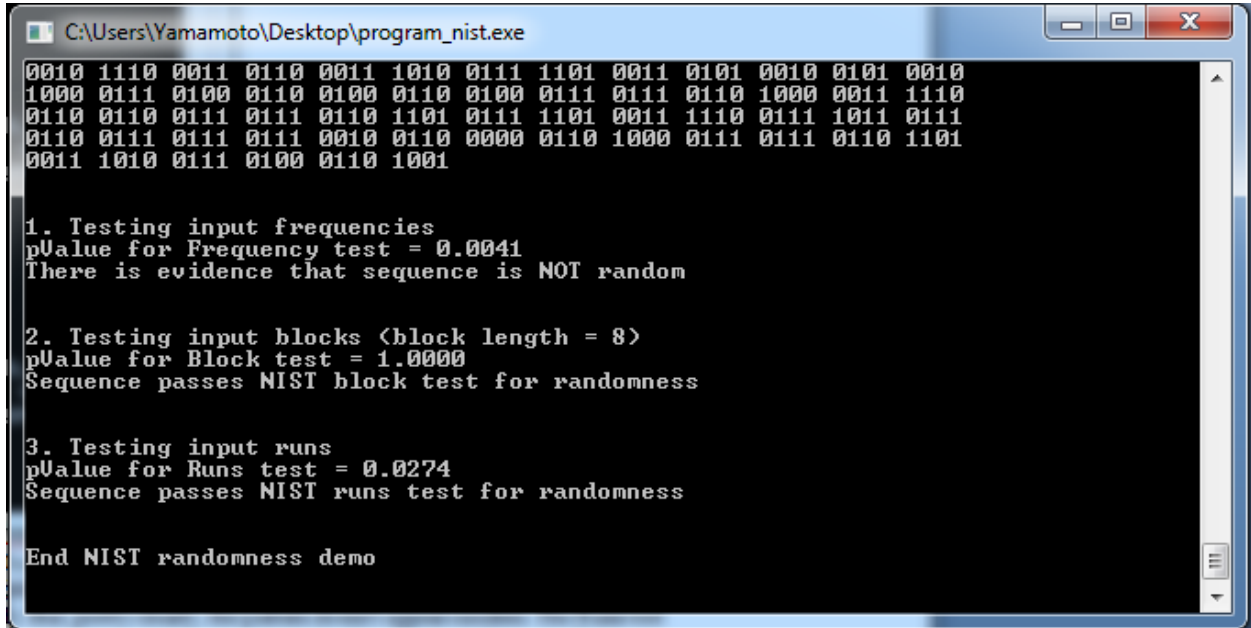


Figure 0.3. Frequency Test on E-DES

Frequency Test (Mono-bit test) pass on (E-DES) using CRYPTOOOL

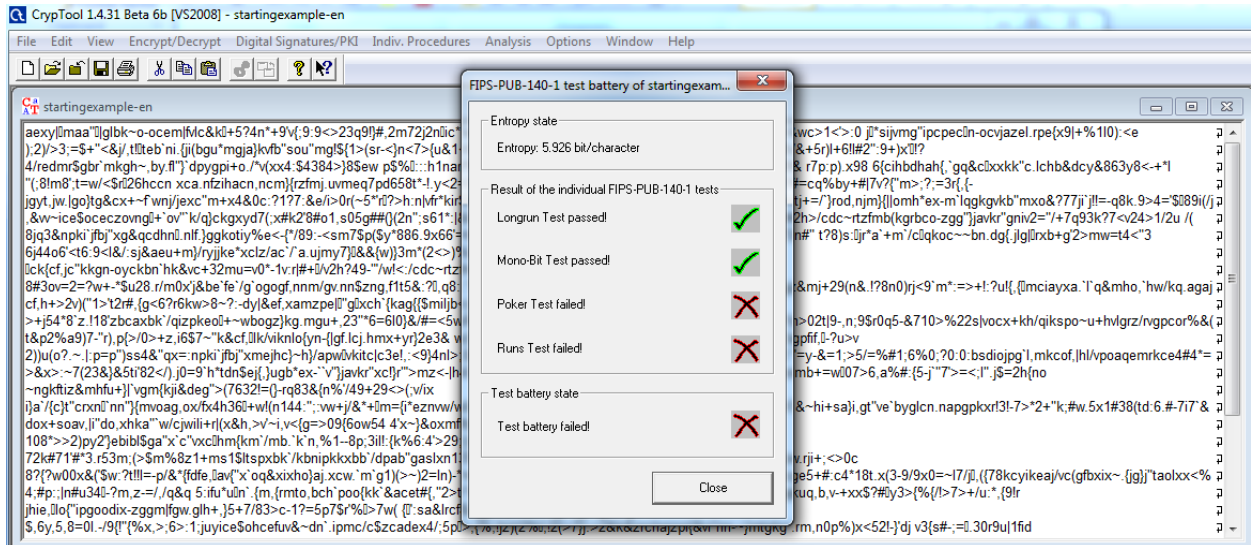


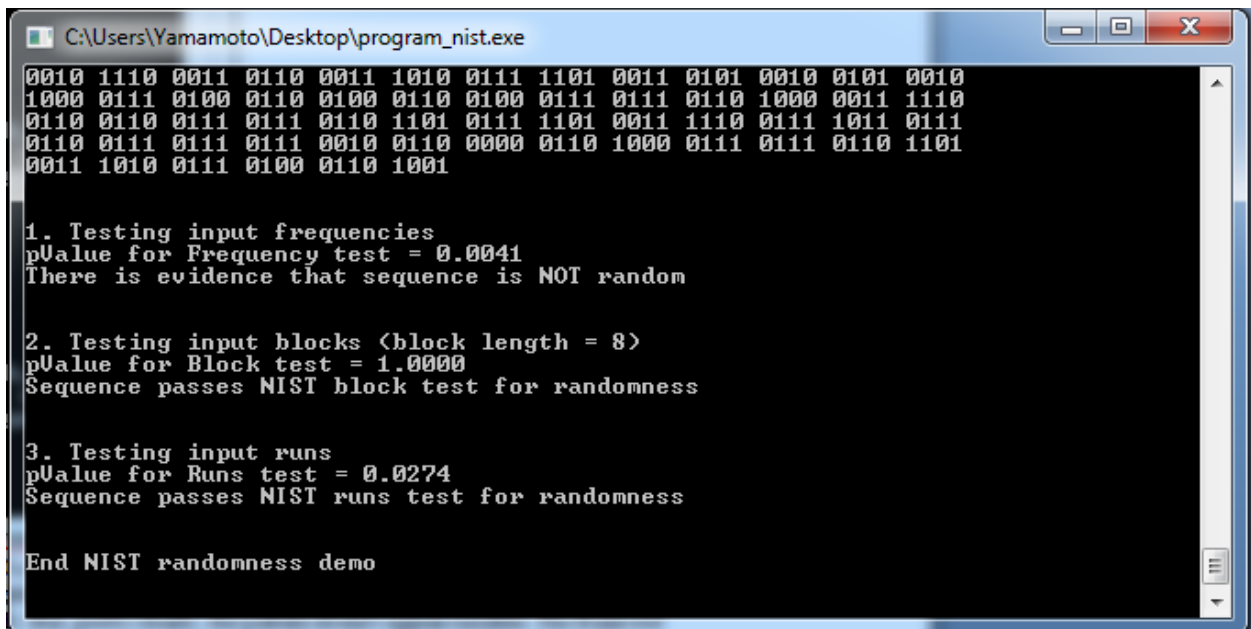
Figure 0.4 Data Encryption Standard [DES] Test Result with Cryptool

4.4.2. The Block Test on E-DES

Consider the pattern 00000000 11111111. This pattern would pass the Frequency test because there are equal numbers of 0s and 1s, but clearly the pattern looks

suspicious. The Block test is designed to address this type of non-randomness. The Block test divides a pattern into blocks and examines the number of 1s in each block. A random pattern would be expected to have about 50 percent 1s in every block.

Block Test pass on (E-DES) using National Institute of Standards and Technology [NIST] tools



```
C:\Users\Yamamoto\Desktop\program_nist.exe
0010 1110 0011 0110 0011 1010 0111 1101 0011 0101 0010 0101 0010
1000 0111 0100 0110 0100 0110 0100 0111 0111 0110 1000 0011 1110
0110 0110 0111 0111 0110 1101 0111 1101 0011 1110 0111 1011 0111
0110 0111 0111 0111 0010 0110 0000 0110 1000 0111 0111 0110 1101
0011 1010 0111 0100 0110 1001

1. Testing input frequencies
pValue for Frequency test = 0.0041
There is evidence that sequence is NOT random

2. Testing input blocks <block length = 8>
pValue for Block test = 1.0000
Sequence passes NIST block test for randomness

3. Testing input runs
pValue for Runs test = 0.0274
Sequence passes NIST runs test for randomness

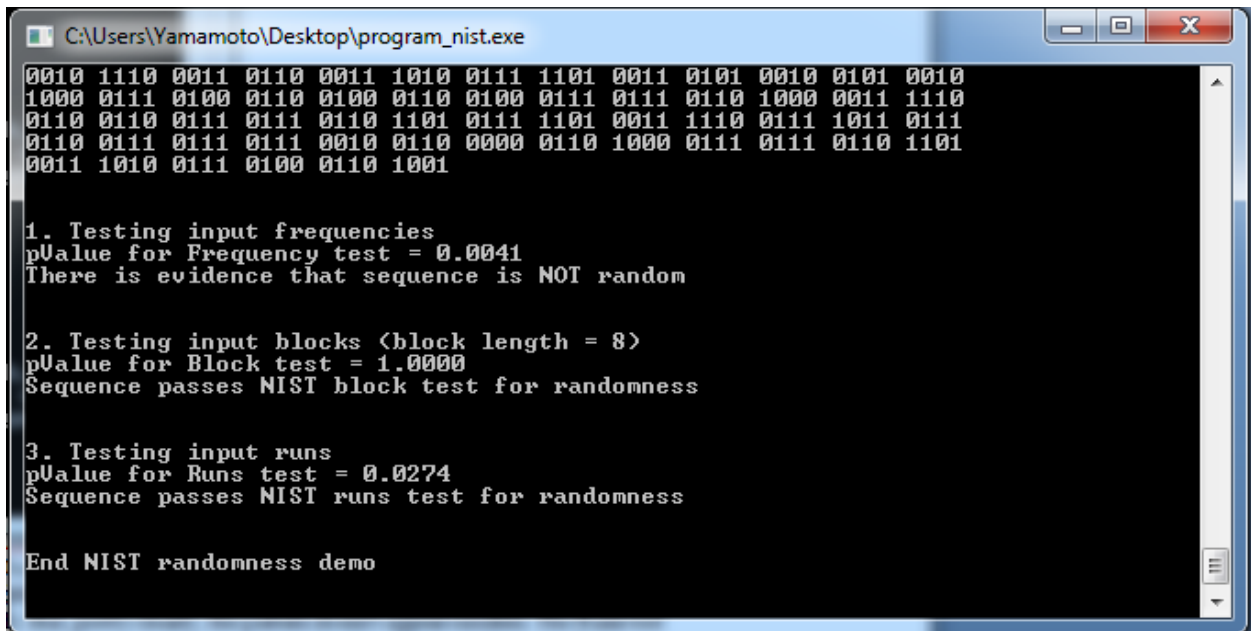
End NIST randomness demo
```

Figure 0.5. Block Test on E-DES

4.4.3. The Runs Test on E-DES

Consider the pattern 0101 0101 0101 0101. This pattern would pass the Frequency test because there are equal numbers of 0s and 1s. The pattern would also likely pass the Block test because each block would have roughly 50 percent of 0 bits and 50 percent of 1 bits (depending on whether the block size is even or odd). But, pretty clearly, the pattern doesn't appear random. The Runs test catches patterns like this. A run is a sequence where consecutive bit tokens are the same. For example, the pattern 00100011 has four runs: 00, 1, 000 and 11. If a pattern is randomly generated, it's possible to compute the expected number of runs.

Runs Test pass on (E-DES) using National Institute of Standards and Technology [NIST] tools



```
C:\Users\Yamamoto\Desktop\program_nist.exe
0010 1110 0011 0110 0011 1010 0111 1101 0011 0101 0010 0101 0010
1000 0111 0100 0110 0100 0110 0100 0111 0111 0110 1000 0011 1110
0110 0110 0111 0111 0110 1101 0111 1101 0011 1110 0111 1011 0111
0110 0111 0111 0111 0010 0110 0000 0110 1000 0111 0111 0110 1101
0011 1010 0111 0100 0110 1001

1. Testing input frequencies
pValue for Frequency test = 0.0041
There is evidence that sequence is NOT random

2. Testing input blocks (block length = 8)
pValue for Block test = 1.0000
Sequence passes NIST block test for randomness

3. Testing input runs
pValue for Runs test = 0.0274
Sequence passes NIST runs test for randomness

End NIST randomness demo
```

Figure 0.6. Runs Test on E-DES

Test on Data Encryption Standard DES algorithm

4.4.4. The Frequency Test on Data Encryption Standard [DES]

Consider the pattern 0100 0000 0000 0010. The pattern doesn't appear to be random because there are too many 0s. The most fundamental test for randomness is a frequency test. If a pattern is generated randomly, you'd expect the number of 0s and 1s to be roughly the same. Too many 0s or too many 1s suggest non-randomness.

Frequency Test failed on Data Encryption Standard [DES] using National Institute of Standards and Technology [NIST] tools

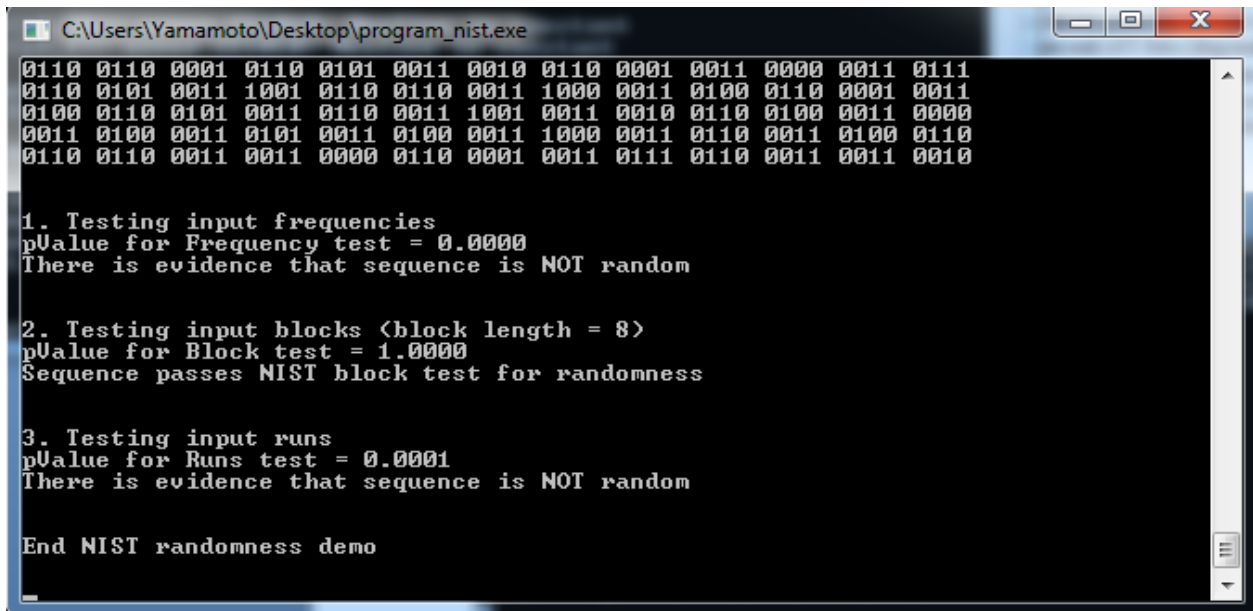


Figure 0.7. Data Encryption Standard [DES] Test Result With Nist
 Frequency Test (Mono-bit test) failed on Data Encryption Standard [DES] using
 CRYPTOOOL

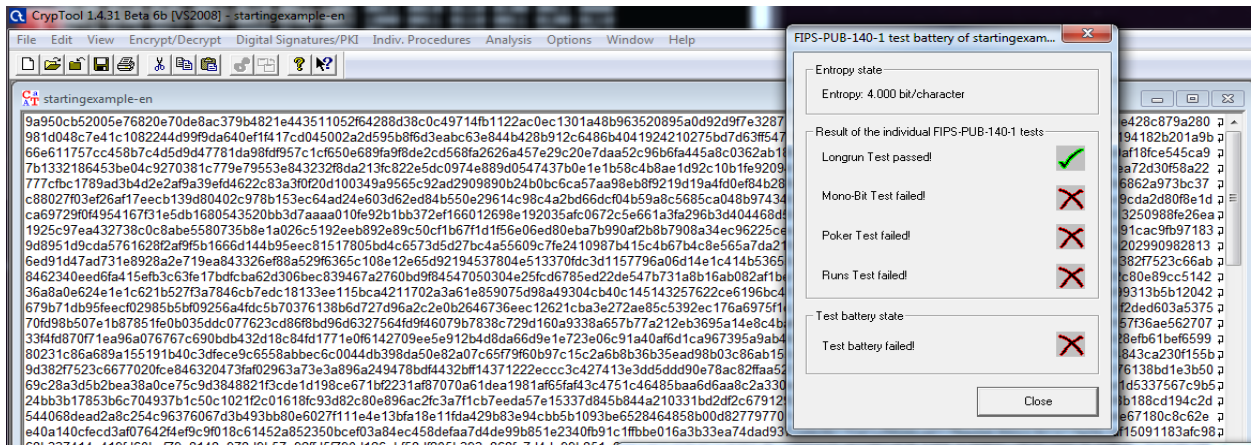


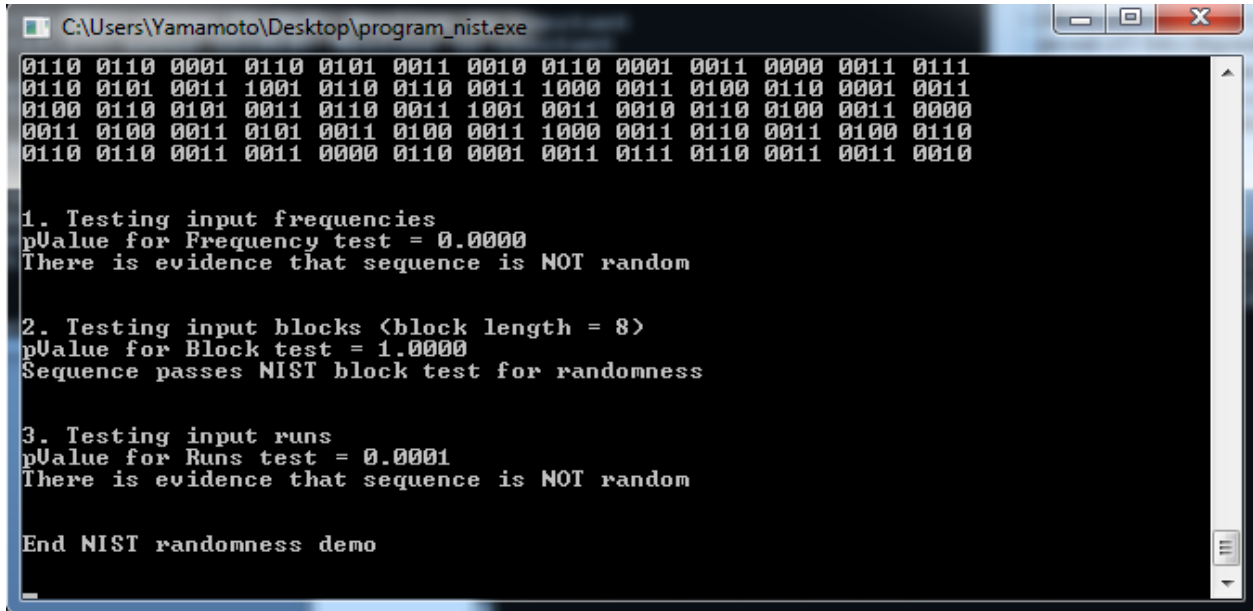
Figure 0.8. E-DES Test Result with Cryptool

4.4.5. The Block Test on Data Encryption Standard [DES]

Consider the pattern 00000000 11111111. This pattern would pass the Frequency test because there are equal numbers of 0s and 1s, but clearly the pattern looks suspicious. The Block test is designed to address this type of non-randomness. The Block test divides a pattern into blocks and examines the number of 1s in each

block. A random pattern would be expected to have about 50 percent 1s in every block.

Block Test pass on Data Encryption Standard [DES] using National Institute of Standards and Technology [NIST] tools



```
C:\Users\Yamamoto\Desktop\program_nist.exe
0110 0110 0001 0110 0101 0011 0010 0110 0001 0011 0000 0011 0111
0110 0101 0011 1001 0110 0110 0011 1000 0011 0100 0110 0001 0011
0100 0110 0101 0011 0110 0011 1001 0011 0010 0110 0100 0011 0000
0011 0100 0011 0101 0011 0100 0011 1000 0011 0110 0011 0100 0110
0110 0110 0011 0011 0000 0110 0001 0011 0111 0110 0011 0011 0010

1. Testing input frequencies
pValue for Frequency test = 0.0000
There is evidence that sequence is NOT random

2. Testing input blocks (block length = 8)
pValue for Block test = 1.0000
Sequence passes NIST block test for randomness

3. Testing input runs
pValue for Runs test = 0.0001
There is evidence that sequence is NOT random

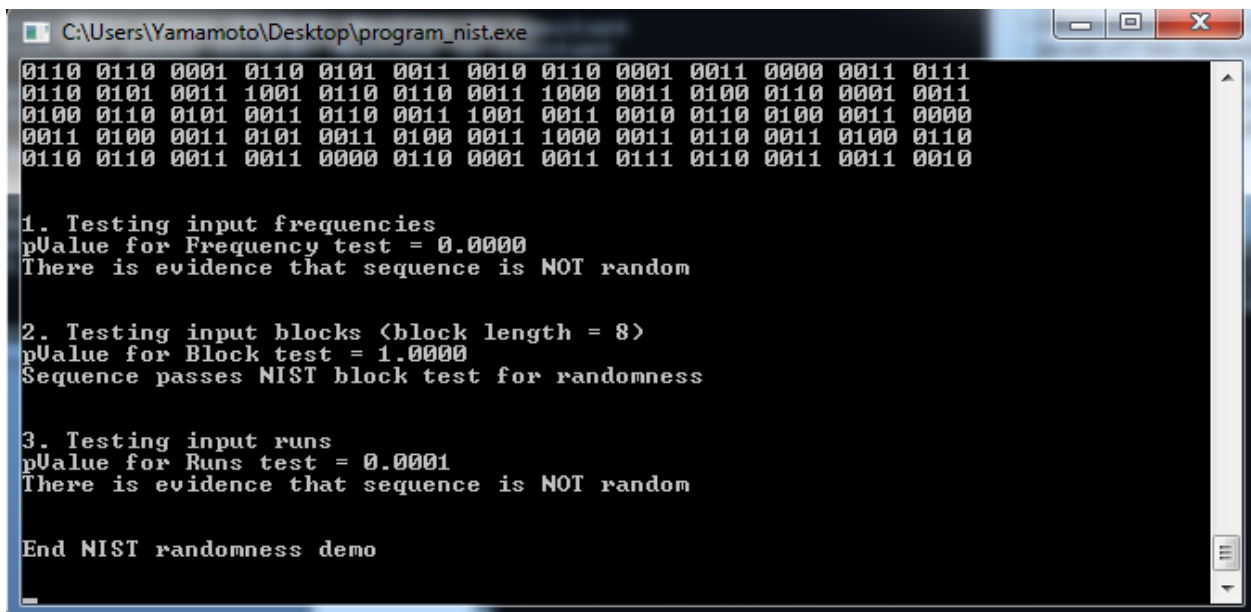
End NIST randomness demo
```

Figure 0.9. Block Test on Data Encryption Standard [DES] With Nist

4.4.6. The Runs Test on Data Encryption Standard [DES]

Consider the pattern 0101 0101 0101 0101. This pattern would pass the Frequency test because there are equal numbers of 0s and 1s. The pattern would also likely pass the Block test because each block would have roughly 50 percent of 0 bits and 50 percent of 1 bits (depending on whether the block size is even or odd). But, pretty clearly, the pattern doesn't appear random. The Runs test catches patterns like this. A run is a sequence where consecutive bit tokens are the same. For example, the pattern 00100011 has four runs: 00, 1, 000 and 11. If a pattern is randomly generated, it's possible to compute the expected number of runs.

Runs Test failed on Data Encryption Standard [DES] using National Institute of Standards and Technology [NIST] tools



```
C:\Users\Yamamoto\Desktop\program_nist.exe
0110 0110 0001 0110 0101 0011 0010 0110 0001 0011 0000 0011 0111
0110 0101 0011 1001 0110 0110 0011 1000 0011 0100 0110 0001 0011
0100 0110 0101 0011 0110 0011 1001 0011 0010 0110 0100 0011 0000
0011 0100 0011 0101 0011 0100 0011 1000 0011 0110 0011 0100 0110
0110 0110 0011 0011 0000 0110 0001 0011 0111 0110 0011 0011 0010

1. Testing input frequencies
pValue for Frequency test = 0.0000
There is evidence that sequence is NOT random

2. Testing input blocks (block length = 8)
pValue for Block test = 1.0000
Sequence passes NIST block test for randomness

3. Testing input runs
pValue for Runs test = 0.0001
There is evidence that sequence is NOT random

End NIST randomness demo
```

Figure 0.10. Runs Test on Data Encryption Standard With Nist

CHAPTER V

CONCLUSIONS

5.1. CONCLUSIONS

Network security is becoming more and more crucial as the volume of data being exchanged on the internet increases. A more practical way to protect information is to alter it so that only an authorized receiver can understand it. There is some weakness in Data Encryption Standard [DES] Algorithm, new algorithm has been introduced.

the results obtained and analyzed for the proposed algorithm using National Institute of Standards and Technology [NIST] statistical test and cryptool it was good. Hence the time for encryption and decryption of proposed algorithm is lesser than existing approaches. As the complexity of the encryption algorithm increases, security also increases and speeds. The main advantage is that it is having 256 length key approaches to make it difficult to break.

References

1. W. Diffie and M. Hellman, "New directions in cryptography", *IEEE Transaction on Information Theory*, 1976, pp. 644–654.
2. Behrouz A Forouzan, "Data Communications and networking", McGraw-Hill, 4th Edition.
3. M Matsui, "Linear Cryptanalysis Method for DES Cipher", in *Advances in Cryptology 7 Eurocrypt 93*, Springer LNCS v 765 pp 3867397
4. N. Ferguson and B. Schneir, "Cryptanalysis of Akelarre", 23 July 1997.
5. Biryukov, J. Nakahara Jr, B. Preneel, J. Vandewalle, "New Weak-Key Classes of IDEA", *4th International Conference Information and Communications Security, ICICS 2002*, *Lecture Notes in Computer Science 2513*, Springer-Verlag, 2002, pp. 315-326.
6. E. Biham, O. Dunkelman, and N. Keller, "A New Attack on 6-Round IDEA", *Proceedings of Fast Software Encryption*, *Lecture Notes in Computer Science*, Springer-Verlag, 2007
7. D. Khovratovich, G. Leurent, and C. Rechberger, "Narrow-Bicliques: Cryptanalysis of Full IDEA", *Advances in Cryptology, EUROCRYPT 2012*, LNCS 7237, Springer-Verlag, 2012, pp. 392–410.
8. G. Álvarez, D. de la Guía, F. Montoya, and A. Peinado, "Akelarre: a new Block Cipher Algorithm", *Third Annual Workshop on Selected Areas in Cryptography, SAC 96*, Kingston, Ontario, 15-16 August 1996, pp. 1-14.
9. G. Álvarez, D. de la Guía, F. Montoya, and A. Peinado, "Description of the new Block Cipher Algorithm Akelarre", <http://www.iec.csic.es/~fausto/papers/akelarre1.ps>
10. R. L. Rivest, "The RC5 Encryption Algorithm", *Proceedings of the Second International Workshop on Fast Software Encryption*, 1994, pp. 86-96.
11. D. Hong, J. K. Lee, D. C. Kim, D. Kwon, K. H. Ryu, and D. G. Lee, "LEA: A 128-Bit Block Cipher for Fast Encryption on Common Processors", *WISA 2013*, LNCS 8267, Springer International Publishing Switzerland 2014.
12. X. Lai and J. Massey, "A Proposal for a New Block Encryption Standard", *Advance in Cryptography, EUROCRYPT 90*, Springer Verlag, Berlin 1991, pp. 389-404.
13. . Tingyuan Nie, Chuanwang Song and Xulong Zhi, "Performance Evaluation of DES and Blowfish Algorithms", *IEEE International Conference on Biomedical Engineering and Computer Science (ICBECS- 2010)*, pp. 1-4, 23-25 Apr 2010.