Ministry of Higher Education and
Scientific Research
University of Babylon
College of Science for Women
Department of Computer Science

An Appropriate Security Mode for Data Ciphering in Life Applications

A Project Submitted to the College of Science / University of Babylon in Partial Fulfillment of the Requirements for the Degree of Higher Diploma in Computer Science

By

Mohammad Talib Hadi

Supervised by

Asst. Prof.Dr. Saif M. Kh. Al-Alak

2021 A.D. 1443 A.H

وزارة التعليم العالي والبحث العلمي

جامعة بابل / كلية العلوم للبنات

قسم علوم الحاسوب

وضع الأمان المناسب لتشفير البيانات في تطبيقات الحياة

بحث مقدم إلى كلية العلوم للبنات / جامعة بابل كجزء من متطلبات نيل درجة الدبلوم العالي في علوم الحاسوب

مقدم من قبل :

محمد طالب هادي

بإشراف :

أ. م. د. سيف محمود خلف العلاك

2021 م

1443 هـ

Ministry of Higher Education and
Scientific Research
University of Babylon
College of Science for Women
Department of Computer Science

An Appropriate Security Mode for Data Ciphering in Life Applications

A Project Submitted to the College of Science / University of Babylon in Partial Fulfillment of the Requirements for the Degree of Higher Diploma in Computer Science

By

Mohammad Talib Hadi

Supervised by

Asst. Prof.Dr. Saif M. Kh. Al-Alak

2021 A.D. 1443 A.H

بسم الله الرحمن الرحيم

قَالَ الَّذِي عِنْدَهُ عِلْمٌ مِنَ الْكِتَابِ أَنَا آتِيكَ بِهِ قَبْلَ أَنْ يَرْتَدَّ إِلَيْكَ طَرْفُكَ فَلَمَّا رَآهُ مُسْتَقِرًّا عِنْدَهُ قَالَ هَذَا مِنْ فَضْلِ رَبِّي لِيَبْلُوَنِي أَأَشْكُرُ أَمْ أَكْفُرُ وَمَنْ شَكَرَ فَإِنَّمَا يَشْكُرُ لِنَفْسِهِ وَمَنْ كَفَرَ فَإِنَّ رَبِّي غَنِيٌّ كَرِيمٌ ۝

صدق الله العلي العظيم

(النمل : 40)

Supervisor's Certification

I certify that this project entitled " **An Appropriate Security Mode for Data Ciphering in Life Applications** " Was done by (Mohammad Talib Hadi) under my supervision.

Signature:

Name: Asst.Prof.Dr. Saif M.Kh. Al-Alak

Date: / / 2021

Affiliation: University of Babylon/College of Science for Women

Head of the Department Certification

In view of the available recommendations, I forward the project entitled "**An Appropriate Security Mode for Data Ciphering in Life Applications**" for debate by the examination committee.

Signature:

Name : Dr. Farah Mohammed Hassan, Lecturer.

Date: / / 2021

Affiliation: University of Babylon/College of Science for Women

Dedication

Thanks to ALLAH in the first and last place, my Creator, to teacher and messenger, Mohammed (May Allah bless and grant him), who taught us the purpose of life, The great martyrs, the symbol of sacrifice, my family who encourage and support me, all the people in my life, I dedicate this research.

Mohammad, 2021

Acknowledgment

Praise and thanks to God who enabled me to complete my research and facilitated the difficulties for me.

Thanks to all my teachers in the college of science for women, particularly I am highly indebted to express my thanks to my supervisor on this research Dr. Saif Al-Alak for his excellent guidance and encouragement to complete my research.

I would like to thank very much my dear mother, father, brothers, sister for their love, patience, and understanding to spend my time on this research.

Special thanks to my wife, sons, and daughter, where this accomplishment would not have been possible without them.

Finally, I would like to thank all my friends and all the people who helped me during my Diploma study.

Mohammad,

2021

Abstract

Security at the present time is very important and highly effective for Internet and network applications, which are rapidly growing, therefore the data that is exchanged over the Internet or other media has increased in value and importance. Therefore, many ciphering-based algorithms have been proposed to provide the required protection against attacks. However, these algorithms differ in the degree of randomness and the time consumed to yield a secure ciphertext. This research aims to test a number of symmetric encryption algorithms (AES, DES, 3DES, RC4, Blowfish, Twofish) with security mode cipher block chaining (CBC). A comparison is then made between them based on evaluation criteria: encryption and decryption time tests are implemented by using Java programming language. The randomness test on ciphertext, which was implemented by using the Diehard statistical test to compute the most efficient algorithm to use in various life applications.

The results of the research showed that the 3DES algorithm is the most time-consuming, followed by DES, while RC4 is the algorithm that needs the least execution time, followed by AES, and both Twofish and Blowfish came between these two levels. As for the randomness criterion, 3DES was the highest compared to the rest of the algorithms, while RC4 and AES were the lowest in this criterion.

Keywords: *AES, DES, 3DES, RC4, Blowfish, Twofish, CBC, Encryption Time, Decryption Time, Randomness.*

Table Of Contents

Table of Figures

Table of Tables

Table of Abbreviations

Abbreviation	Meaning
AES	Advanced Encryption Standard
DES	Data Encryption Standard
3DES	Triple Data Encryption Standard
RC4	Rivest Cipher 4
SA	Security Algorithm
SM	Security Mode
CBC	Cipher Block Chaining
OFB	Output Feed Back
CFB	Cipher Feedback
ECB	Electronic Code Book
CTR	Counter
WEP	Wired Equivalent Privacy
WPA	Wi-Fi Protected Access
RSA	Rivest-Shamir-Adleman
DSA	Digital Signature Algorithm
ECC	Elliptic Curve Cryptography
HMAC	Hash-based Message Authentication Code
SEAL	Software-Optimized Encryption Algorithm
NIST	National Institute of Standards and Technology
KSA	Key Scheduling Algorithm
PRGA	Pseudo Random Generation Algorithm

CHAPTER ONE

GENERAL INTRODUCTION

CHAPTER TWO

THEORETICAL BACKGROUND

CHAPTER THREE

PROPOSED RESEARCH

CHAPTER FOUR

SYSTEM IMPLEMENTATION AND RESULTS

CHAPTER FIVE

CONCLUSIONS AND FUTURE WORKS

Chapter One : General Introduction

1.1 Overview

In digital communications, security is of great importance. Therefore, the encryption process is one of the most important areas of computer security. Encryption is defined as the process that sends data through unsecured channels so that only the authorized recipient and owner of the secret key can view and read the ciphertext, which may be pictures, documents, telephone conversations, or another form of data, as shown in figure (1.1) below.

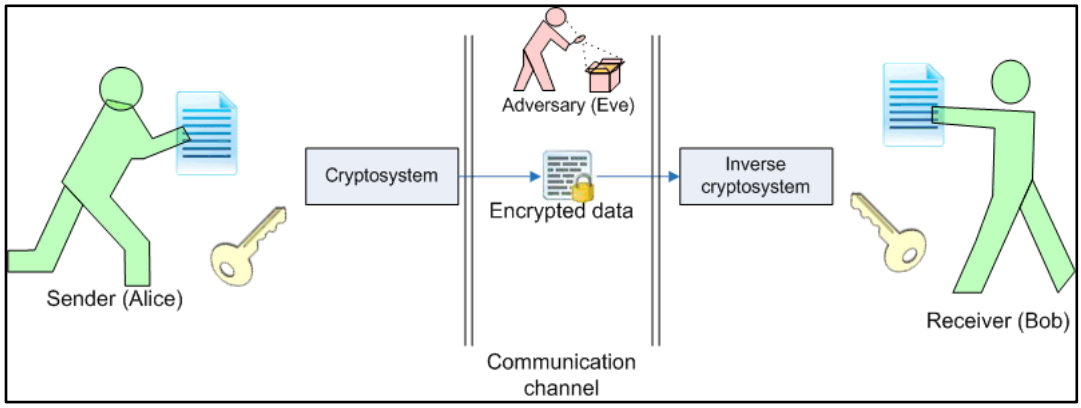

Figure (1.1) : Cryptosystem

To ensure that other users do not have access to the actual information, this information must be mixed using encryption systems. With privacy remaining one of the main goals, this field has expanded and includes a number of important goals that are not limited to the security of communications only, such as the authenticity of these communications and ensuring their safety and other more complex goals.

Secure cryptographic techniques are essential to protect confidential information. There are two main types of cryptographic systems: private (symmetric) algorithms and public (asymmetric) algorithms. Private key systems

require a master agreement over an existing secure channel while public key algorithms are very slow compared to private key algorithms[1].

In this research, different symmetric security algorithms (SA) are tested in a specific security mode (SM) such as Cipher Block Chaining (CBC). The research measures the randomness of ciphertext generated by these security algorithms using Diehard's statistical test, as well as calculating the execution time required for data encryption and decryption. Then a comparison is made between the results obtained for the purpose of determining the most efficient algorithm for use and according to the criteria specified by the various life applications.

1.2 Related Studies

This part, highlights on a number of works related to the topic of research. These works make comparisons between a variety of encryption algorithms and give results based on specific evaluation criteria.

Adolf Fenyi, et al . in [2] presents a comparison between the algorithms Blowfish, AES, and RC4 with security modes (CBC, CFB, ECB) in terms of evaluation criteria: encryption and decryption time, as well as memory usage. The mentioned algorithms were applied to files of different sizes. The results showed that RC4 was the fastest algorithm, followed by Blowfish, while AES came last in terms of speed and memory consumption than both Blowfish and RC4 algorithms. In terms of security modes, ECB was found to be the fastest mode, followed by CBC, while CFB was the slowest. This explains why the encryption speeds in ECB are higher than the rest at all file sizes selected for encryption.

R.venkateshwarlu, et al . in [3] made a comparison between common symmetric encryption algorithms such as (AES, DES, Blowfish, Twofish) and since the important thing is the performance of these algorithms in different configurations, so the comparison that was presented took into account the performance and behavior of the algorithms when using different sizes of data. The comparison was made based on a number of evaluation criteria such as key size, speed, and block size. He showed in the simulation outcomes that (Twofish) has a larger performance than the rest of the other used algorithms, and it can be considered as a standard encryption algorithm because it has no known weaknesses in terms of security so far. Regarding (AES) it showed weak performance results compared to the rest of the algorithms because it requires more processing capacity.

J.B.Awotunde, et al . in [4] consider the different key size, memory creation rate, CPU usage time, and encryption process speed for the four algorithms (DES, 3DES, AES, Blowfish) for the purpose of determining the amount of computer resources expended and the time it takes for each algorithm to complete its task. They showed in their results that there is a proportionality between the key length used in encryption algorithms and resource usage in most cases, so the (Blowfish) algorithm uses more time, memory and CPU usage in executing encryption operations because it uses a key length much higher than (448 bit). They also indicated in their results that the use of high key length encryption algorithms should not be recommended for memory and power sensitive devices, which are often small in size and do not perform well in such hot conditions.

Archisman Ghosh presented in [5] a comparison among three symmetric algorithms (Twofish, Blowfish and AES) in terms of throughput and data encryption and decryption time. He explained that the (Twofish) encryption algorithm has an advantage in terms of the evaluation criteria mentioned earlier

over the algorithms (AES, Blowfish) and as a result of the decrease in data encryption and decryption time and the increase in productivity in (Twofish) can be implemented in the security of all network protocols along with HMAC.

B. Nithya and P. Sripriya in [6] focused on four symmetric algorithms (AES, RC4, DES, 3DES) in terms of memory usage standards, throughput, encoding and decoding time. These tests were performed on text files of different sizes. The results showed that DES has less encoding time and takes less memory for decoding purpose, and on the other hand its throughput is low. While 3DES has a high decoding time in addition to using a large space for both encryption and decryption, its throughput is better than RC4 and DES. RC4 uses high encoding and decoding time, less memory, and low throughput when compared to other algorithms. AES has better throughput and requires less space for both encryption and decryption.

Masood Ahmad, et al. In [7] explained that there are strengths and weaknesses for all encryption algorithms, which are chosen depending on the requirements of the application. A comparison was made between a number of symmetric encryption algorithms (AES, Blowfish, DES, 3DES), and through the comparison, it was found that Blowfish is the best choice in the case of memory and time as it records the least time compared to the rest of the proposed algorithms. But if integrity and confidentiality are the factors required by the application, it is better to choose AES. Whereas, if the network bandwidth is the application request, then DES can be chosen from among the algorithms proposed in this work.

1.3 Research Problem

Security at the present time is very important and highly effective for Internet and network applications. They are rapidly growing and therefore the data that is exchanged over the Internet or other media has increased in value and importance. Furthermore, the process of searching for the best solutions for the purpose of providing the required protection against illegal attacks with the provision of these services in a timely manner and required robustness is one of the most interesting topics in security-related communities.

1.4 Research Objectives

The objectives of this research are:

(1) Finding the best algorithm among the symmetric encryption algorithms (AES, DES, 3DES, RC4, Blowfish, Twofish) in the security mode (CBC) in terms of the encryption and decryption time.

(2) Finding the best algorithm among the symmetric encryption algorithms (AES, DES, 3DES, RC4, Blowfish, Twofish) in the security mode (CBC) in terms of the randomness.

1.5 Research Structure

This research is organized as follows:

Chapter Two : offers some basic information about encryption and types of encryption algorithms: symmetric and asymmetric algorithms, security modes, as well as evaluation criteria among the proposed algorithms.

Chapter Three : presents the proposed research.

Chapter Four : show the system implementation and computed results of the research.

Chapter Five : presents the conclusions and suggests some works that could be applied in the future.

Chapter Two : Theoretical Background

2.1 Introduction

This chapter provides some basic information about encryption and its main techniques are used to convert information into an unreadable form. There are two types of encryption algorithms: symmetric algorithms and asymmetric algorithms. To implement security algorithms (SA) with high performance and good security level it could be implemented with different Security Modes (SM).

Cryptography is one of the important tools used to protect transmitted data, as it changes the format of this data into a form that only the intended recipient can read, understand and use.

Cryptography is the science and art of preserving confidential and sensitive information from unauthorized accesses [8].

2.2 Cryptographic Primitives

Cryptographic Primitives are those algorithms that employ mathematical functions to conduct encryption or decryption on secret messages. In general, an encryption process is a set of steps for converting *plaintext* into *ciphertext*, which is a confused form of *plaintext*. Whereas the process of retrieving the original *plaintext* from the *ciphertext* is called decryption. Both of these processes rely on a piece of data, called *a key*, which allows only the parties that hold it to perform these operations on a message [9].

2.2.1 Block Ciphers

It is a type of symmetric key encryption that breaks a message into blocks and then encrypts those blocks in sequence. Its fixed-length plaintext block is equal to

the length of the ciphertext block. Block ciphers are usually more secure and complicated but slow compared to stream ciphers. The examples of this type of encryption include: AES, DES, Blowfish [1].

2.2.2 Stream Ciphers

In this type of private key algorithm the key stream is created by the stream cipher algorithm . The key stream bits and plaintext bits are combined, frequently using the exclusive-or (XOR) operation, and the result is the ciphertext. Stream ciphers are quicker than block cipher methods. The examples of this type of encryption are : RC4, SNOW, and SEAL [10].

2.3 Symmetric Algorithms

Symmetric encryption algorithms are called (secret key or private key algorithms). This type of algorithm uses the same secret key for encryption and decryption, where this key is shared by the sender and recipient. When the encryption process is applied to the hard disk data, then only the user can access the secret key, while when encrypting the transmitted data, each partner can receive a copy of the shared key [11].

2.3.1 Advanced Encryption Standard (AES)

A block cipher method was first published in 2000 by the " National Institute of Standards and Technology (NIST) " in order to achieve a higher security rate than the previous algorithm (DES). AES used the Rijndael algorithm with a fixed block size of 128 bits, and key lengths (128, 192, 256) bits. This solution has been adopted as a standard for the purpose of the exchange of sensitive and non-confidential data by the United States government. The Rijndael algorithm was the initial name for the AES algorithm. This term, however, has not become a widespread name for this algorithm; instead, it is known as the Advanced Encryption Standard (AES) algorithm around the world [12].

The AES algorithm operates on a state array which is arranged as 4 × 4 main columns of bytes, although some versions have additional state columns as well as a larger block size. Most Rijndael calculations are done in a specific field.

The size of the key used in AES determines the number of iterations of conversion rounds responsible for converting plaintext to ciphertext. The number of these iterations was:

- 128 bit keys = 10 rounds of repetition.
- 192 bit keys = 12 round of repetition.
- 256 bit keys = 14 round of repetition.

Each of these rounds consists of several processing steps, each of which contains four similar but different stages, including one based on the same encryption key.

For the purpose of converting the ciphertext to the original plaintext, a collection of reverse cycles is applied utilizing the very encoding key. as showed in figure (2.1) [13].

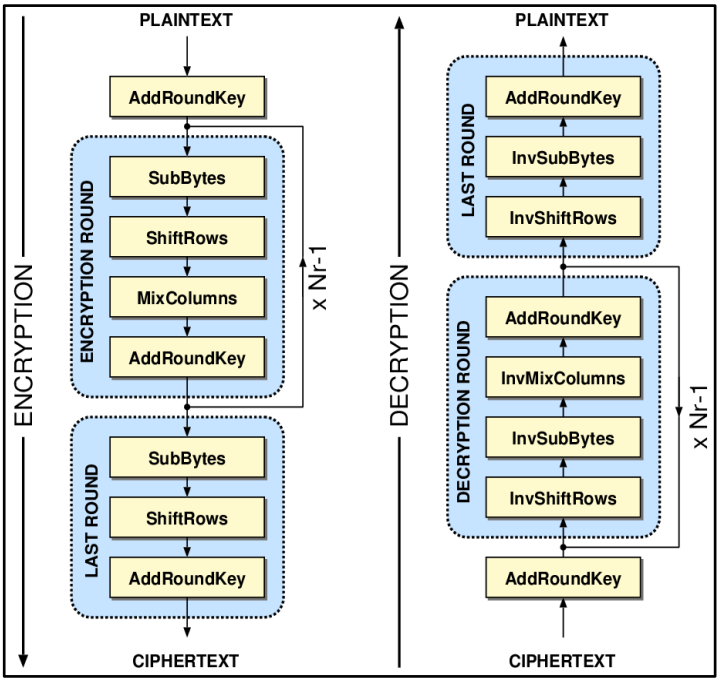

Figure : (2.1) AES Implementation Scheme.

2.3.2 Data Encryption Standard (DES)

Electronic data is protected using the Data Encryption Standard. The DES algorithm encrypts and decrypts data using a symmetric block cipher. The input to DES is 64 bits, and the output is also 64 bits. A second input, a secret key with a length of 64 bits, is required for the process. A message is separated into blocks of bits using the block cipher technique. Substitution, transposition, and other mathematical functions are used to put these blocks of bits together as illustrated in figure (2.2) [14].

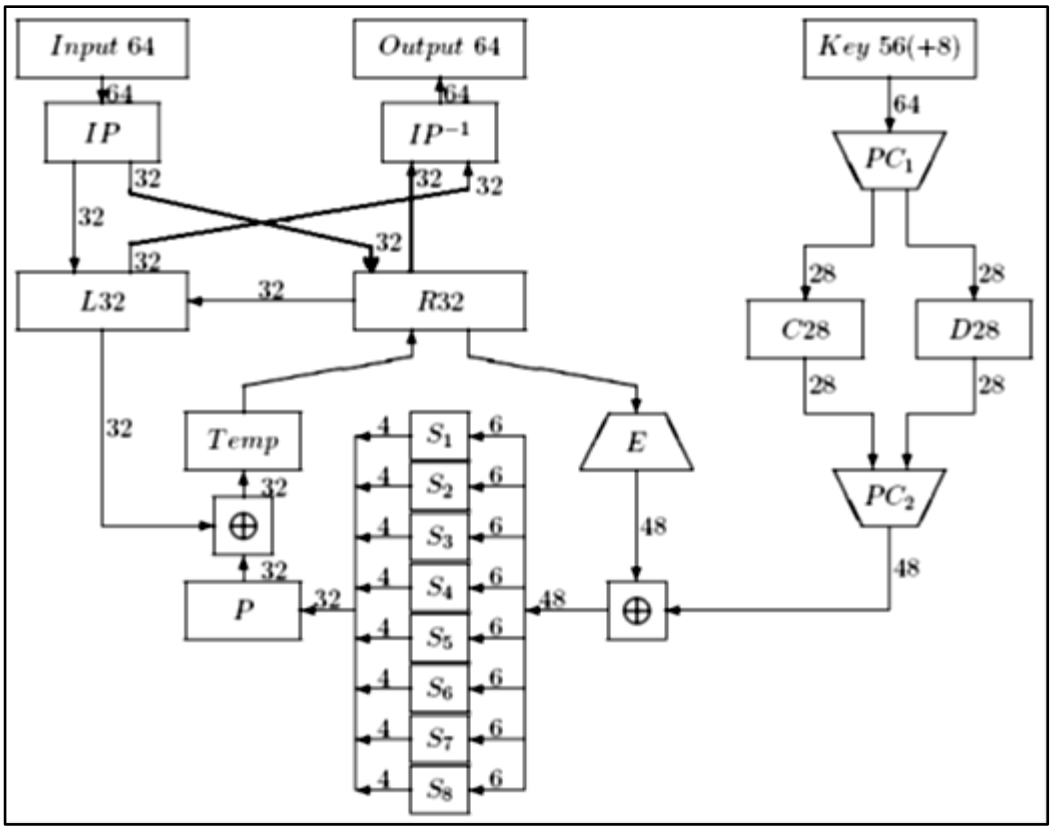

Figure (2.2) : DES Scheme.

Because the DES algorithm works with different security modes (CBC, Output Feed Back (OFB),.....etc.) it is considered more flexible. The main drawback of this algorithm is that it uses a 56-bit key which was broken in 22 hours in 1998 and because of that the DES algorithm was modified to 3DES [15].

2.3.3 Triple Data Encryption Standard (3DES)

3DES is a block cipher type private encryption algorithm. The 3DES algorithm is a safer version when compared to the DES method, where runs the DES algorithm three times on each data block, yielding a key strength of 112 or 168 bits. In the encryption and decryption procedure, the 3DES method requires three keys. By employing the same one key, two different keys, or three different keys to each other, the key variants in 3DES can be grouped into three categories. For current use, 3DES encryption with two or three distinct keys is still deemed robust as showed in figure (2.3) [16].

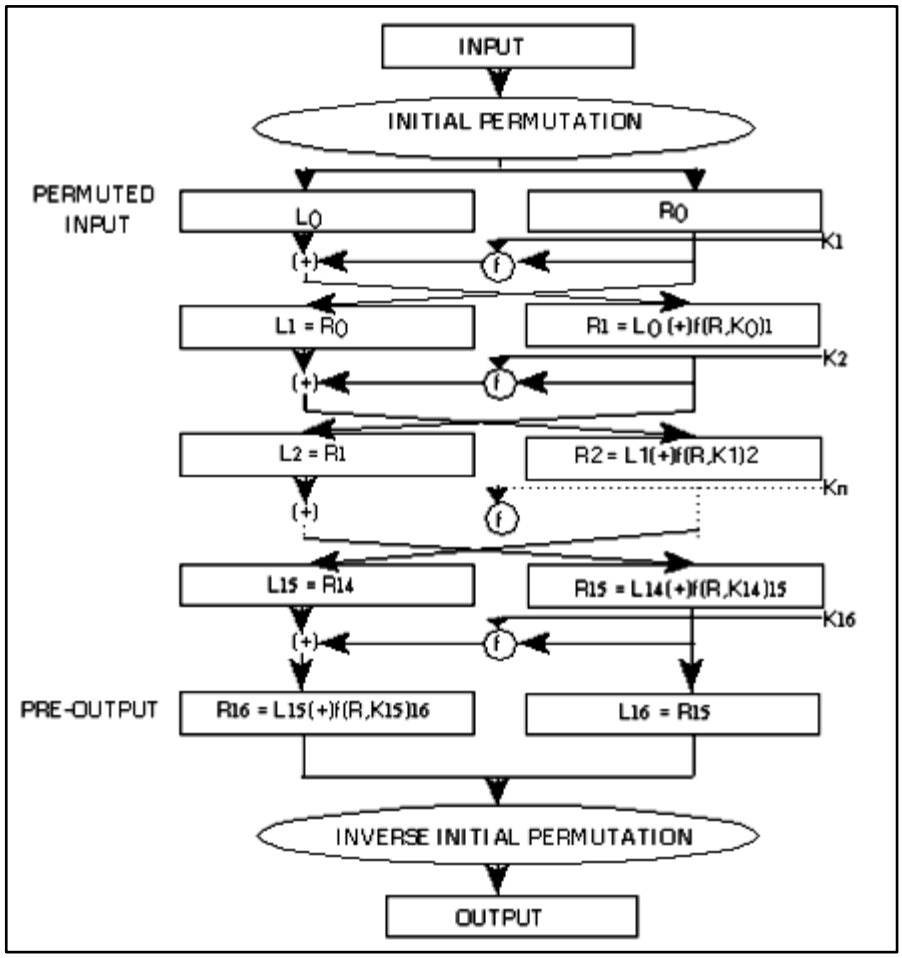

Figure (2.3) : 3DES Scheme using permutations

2.3.4 Rivest Cipher 4 (RC4)

It is a stream cipher 128-bit often used in wireless security protocols such as WEP, WPA. Although RC4 is still widely used today, most security professionals prefer the more modern RC5 and RC6 [17].

At one time, the RC4 processes unit or input data. A byte, or even bits, is a unit of data. Encryption or decryption can be performed on the length of the variable in this manner [18]. This algorithm does not require a particular quantity of data input to wait for it before it is processed, nor does it require extra bytes to encrypt as showed in figure (2.4) .

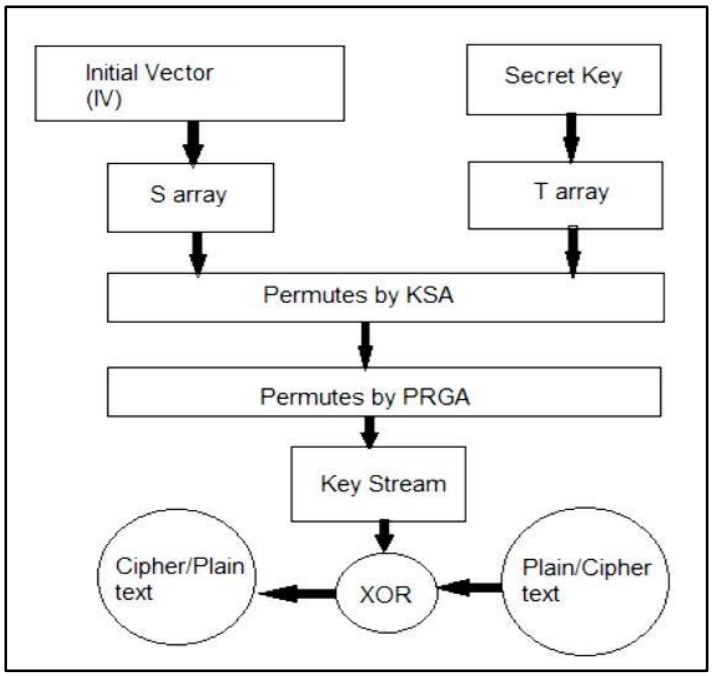

Figure (2.4) : RC4 Encryption and Decryption Scheme.

2.3.5 Blowfish Algorithm

It is a private key block cipher technique that protects encrypted data with a block size of 64 bits and a variable key cipher length of 32 to 448 bits. The fiestal network is the algorithm's structure. Bruce Schneier first proposed the technique in

1993, and it has yet to be cracked. Because of its compactness, it can be optimized in hardware applications [19]. As the lengths of the keys greater than 128 bits , it can be a reliable encryption method. Its work showed in figure (2.5).

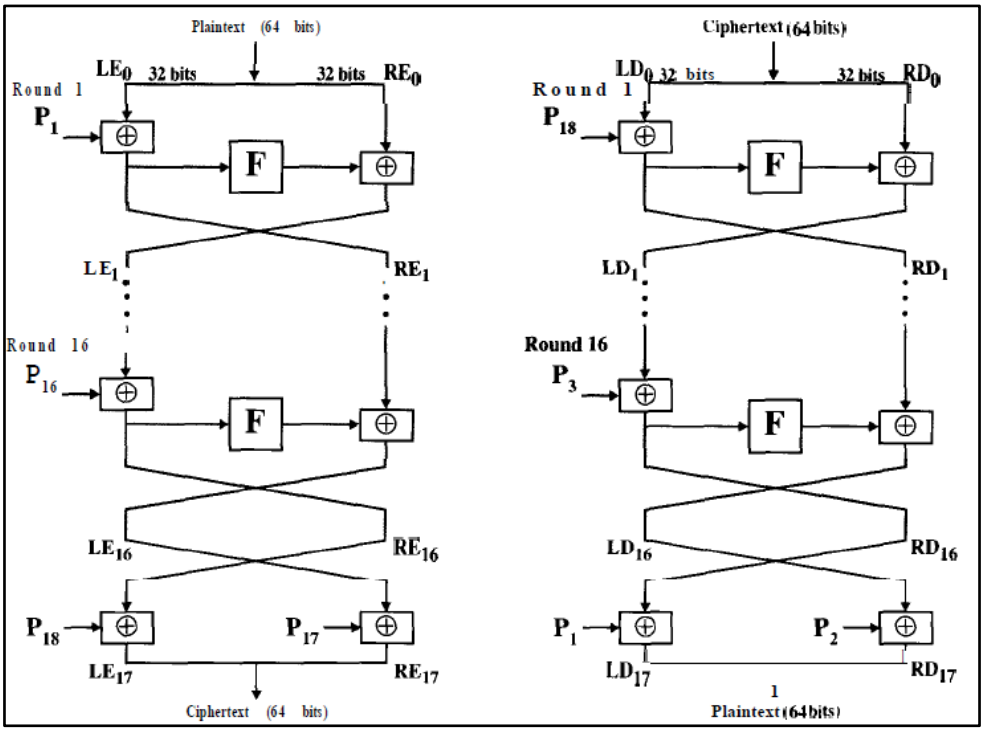

Figure (2.5) : Blowfish Encryption and Decryption Scheme.

2.3.6 Twofish Algorithm

B.Schneier proposed Twofish, a 128-bit symmetric key block cipher. Twofish takes keys of any length up to 256 bits. The cipher is a 16-round Feistel network. Key-dependent S-boxes, maximum distance separable (MDS) matrice, pseudo-Hadamard transform (PHT), and an extremely complex key schedule are all notable aspects of the design Twofish. This algorithm works as illustrated in figure (2.5) [20].

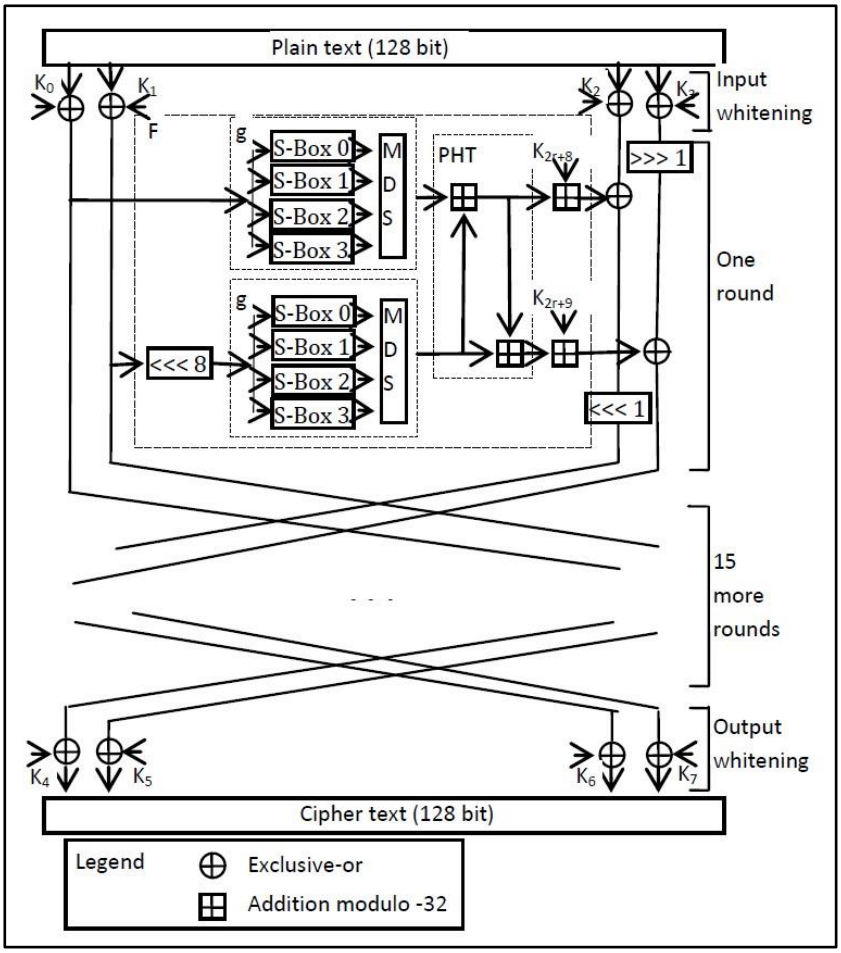

Figure (2.6) : Twofish Scheme.

2.4 Cipher Security Mode

Cipher security mode is an algorithm that is used to encrypt a block in order to provide security for information such as authenticity and confidentiality. This cipher, by itself, means that it is only suitable for secure cipher conversion (encryption or decryption) for a single set of fixed-length block of bits. NIST has specified in its own publication (800-38A) a number of security modes such as (CBC, Electronic Code Book (ECB), and Counter (CTR)) [21].

2.4.1 CBC Mode

The cryptography Security Mode (CBC) includes a feedback process for the ciphertext in the encryption process, as shown in figure (2.7). Each ciphertext block is fed into this method again and used in the next plaintext block cipher.

However, CBC security mode is also vulnerable to parser attacks since similar text blocks will produce the same ciphertext block, and repeated text formats will provide evidence to the intruder. For the purpose of solving this problem, the initialization vector (IV) contains a randomly generated number that is used as the first block of ciphertext (C0) for cipher randomization, thus different ciphertexts will be produced even if the same plaintext cipher is performed a number of times independently and with the same key. This will ensure that any plaintext is encoded in various methods. An initialization vector (IV) usually needn't be secret and can be passed as a common value for the purpose of synchronizing the encryption and decryption processes. Security mode (CBC) is characterized by its ability to use the same key to encrypt more than one text. The main weakness of CBC is that the encryption is sequential. Decryption using a false (IV) causes the first plaintext block to be corrupt but subsequent blocks of plaintext will be correct. This is because the plaintext block can be decrypted using adjacent blocks of the ciphertext [22]. The encryption process using CBC security mode as shown in equations (1a, 1b):

$$C_0 = IV\,, \qquad \text{.....(1a)}$$

$$C_i = E(P_i \oplus C_{i-1}), \quad 1 \le i \le m\,, \qquad \text{.....(1b)}$$

Where IV is the initialization vector, Pi stands for the i-plaintext, Ci stands for the equivalent ciphertext, and m stands for the inclusive number of the text blocks. E stands for an encryption technique, while $\oplus$ stands for a bitwise XOR

operation. As for decrypting the ciphertext, it is done as illustrated in equation (2).

$$P_i = D(C_i) \oplus C_{i-1}, \quad 1 \le i \le m \qquad \text{.....(2)}$$

Where D describes the corresponding decrypt system.

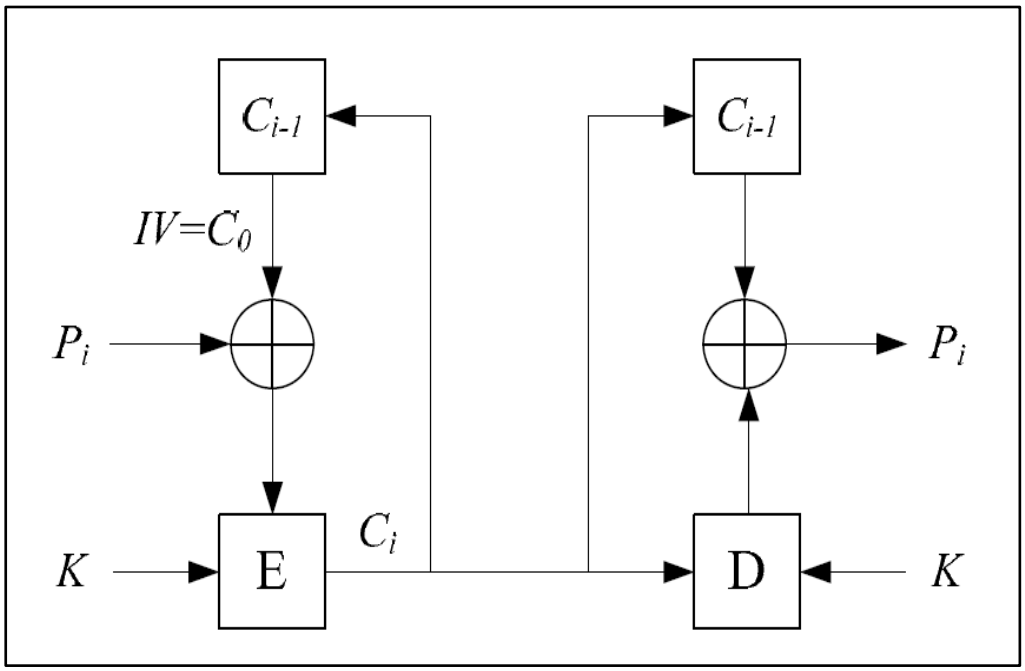

Figure (2.7) : CBC Mode.

2.5 Asymmetric Algorithms

For both encryption and decryption, this approach uses two different keys. The terms "public key" and "private key" are used to describe them. The sender and recipient both know the public key, Whereas only the sender and receiver know their private key. The plain text is encrypted with a public key, and the cipher text is decrypted with a private key. Asymmetric key encryption needs further processing as a different key is used for encryption and decryption [23].

2.5.1 Rivest-Shamir-Adleman (RSA)

Rivest, Shamir, and Adleman invented the first workable public-key cryptosystem, which they named RSA. It provides for the encoding and decoding of the data used primarily while transportation between two parties across a communication medium using two keys, the public key and the private key, which are used respectively to lock and unlock data. The data is encrypted and sent using the sender's public key, while the decryption process will be done using the receiver's private key. As a result, data can only be decrypted using a private key. RSA encryption is widely used for both information and network security, as it is implemented in browsers and web servers [24].

2.5.2 Digital Signature Algorithm (DSA)

The DSA algorithm is distinct from the Schnorr also ElGamal algorithms. Rather than encrypting data, it creates 320 bits digital signatures with key strengths ranging from 512 to 1024 using a discrete logarithm (multiples of 64). There are a large number of digital signature algorithms. This method generally specifies two complementary algorithms, the first for signature and the second for verification, while the output for the signature process is called digital signature [25].

2.6 Evaluation Criteria

This part includes three criteria that will be adopted in this research.

2.6.1 Encryption Time

The time that demands the encryption algorithm to convert the plain text to the ciphertext is named (encryption time), it is measured in milliseconds and depends on the size of both the data block and the key used. This criterion is considered as an efficiency indicator for encryption algorithms, as the less time it takes, the higher the percentage of using this algorithm and including it in different life applications such as banking services, e-commerce, and others [26].

2.6.2 Decryption Time

The time taken to convert the ciphertext into plain text data by the encryption algorithm is called (decryption time) and the unit of milliseconds is used to measure this time. The efficiency of the algorithm increases as the decryption time decreases [5].

2.6.3 Randomness

One of the important criteria used to evaluate encryption algorithms is the randomness of the resulting ciphertext. The higher the value of this criterion, the more secure the algorithm used to encrypt data in life applications. The randomness value is calculated by Diehard Test [27]. This program performs a set of statistical tests on ciphertext of size 11-12 MB to measure the quality of the random number generator, which was developed by George Marsaglia and published in 1995 for the first time [28].

Chapter Three : Proposed Research

3.1 Introduction

The process of searching for the best solutions for the purpose of timely delivery of data while ensuring the required protection against illegal attacks is one of the most interesting topics in security-related communities.

One of the ways to obtain the above requirements is to encrypt data exchanged between life applications. There are two main types of cryptographic systems: private (symmetric) algorithms which use the selfsame key for encode and decode the data, and public (asymmetric) algorithms which used different keys : public and private keys to get its work done. Also, to ensure higher performance and better security, these algorithms must be implemented in one of the security modes (SM). There are strengths and weaknesses for every encryption algorithm. In order to apply a suitable coding technique in life applications, there must be a complete knowledge of the strengths, weaknesses and performance of these algorithms.

3.2 Research Design

In this proposed research, three algorithms were designed to fulfill the research objectives, which are the encryption time calculation algorithm, the decryption time calculation algorithm, and the randomness calculation algorithm for ciphertext.

In addition to six cases will be implemented in this research which are commonly used symmetric encryption algorithms (AES, DES, 3DES, RC4, Blowfish, Twofish) using Java language in security mode (CBC) as illustrated in figure (3.1).

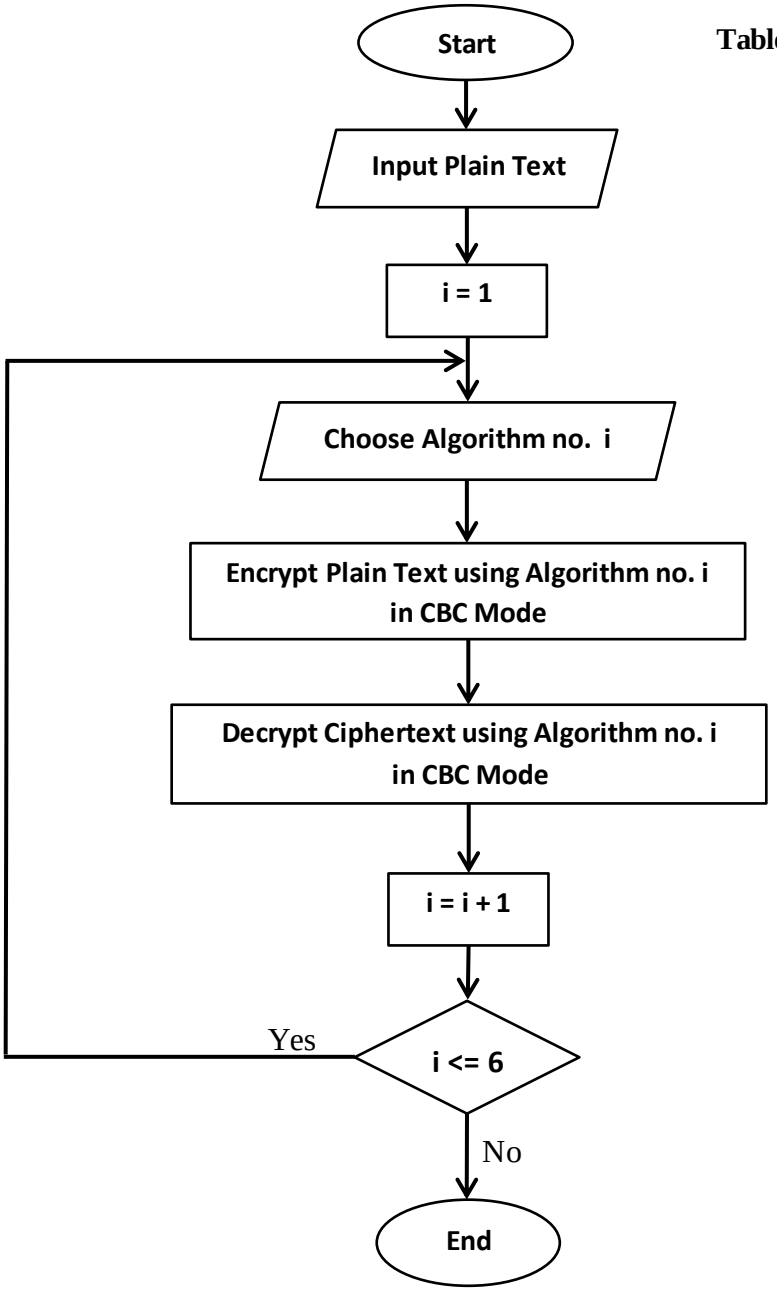

Table (3.1) : Sequence of Algorithm

i	Algorithm
1	AES
2	DES
3	3DES
4	RC4
5	Blowfish
6	Twofish

Figure (3.1) : Research Design.

3.3 Steps of Research Design :

As illustrated in figure (3.1), the working mechanism of the research design consists of the following steps:

(1) The first proposed algorithm (Algorithm no i) is chosen as illustrated in table (3.1) for plaintext encryption with security mode (CBC). The result of this step is the ciphertext.

(2) In this step, the ciphertext in step (1) is decrypted with the same encryption algorithm used and with the same security mode.

(3) The counter (i) is incremented by one and the previous two steps are repeated to test the remaining six proposed algorithms according to the sequence in the table (3.1).

The final output of the above three working steps will be six ciphertexts as well as six files after the decryption processes.

3.4 Ciphering / Deciphering Time

To compute the time of encryption and decryption, two subtests are implemented. First one is implemented to compute encryption time, while the second one is implemented to compute decryption time.

3.4.1 Encryption Time

The total execution time that required for encryption the plaintext is calculated as in the following steps and as illustrated in figure (3.2) :

(1) In the first step, plaintext is entered and the start time is recorded in milliseconds by calling the function (System.currentTimeMillis( )).

(2) The plaintext is encrypted using one of the encryption algorithms proposed in this research.

(3) After completing the plaintext encoding process, the final time is recorded in milliseconds by calling the same function (System.currentTimeMillis( )).

(4) The total time required to encrypt the plaintext with any of the proposed encryption algorithms is calculated as illustrated in equation (3.1).

Stored Encryption Time = "End Time – Start Time" …… (3.1)

Where :

End Time = Time recorded after the encryption process is completed.

Start Time = Time recorded before the encryption process was started.

Stored Encryption Time = Total time required to complete the encryption process.

The previous four steps were repeated using a loop of 100 cycles to increase the accuracy of the calculated time required to encode the plaintext.

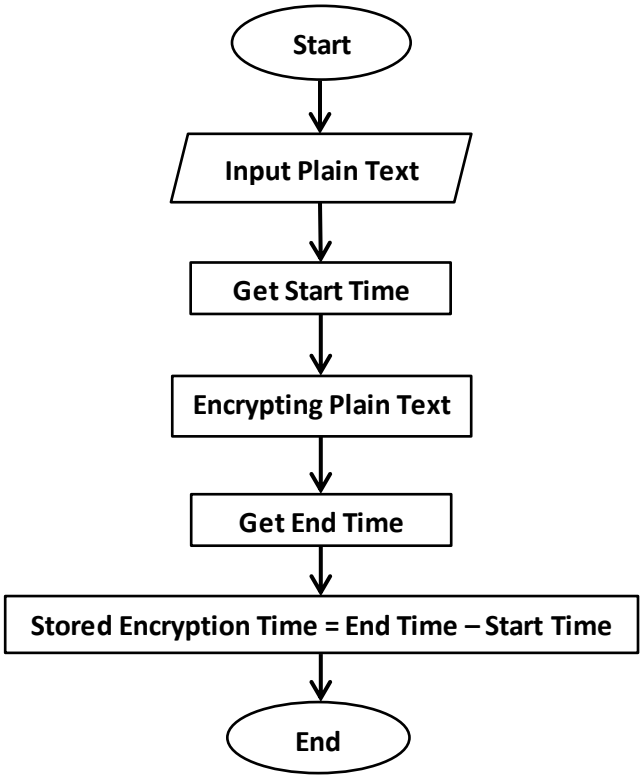

Figure (3.2) : Encryption Time.

3.4.2 Decryption Time

The total execution time required for ciphertext decoding is calculated as in the following steps and as illustrated in figure (3.3) :

(1) In the first step, ciphertext is entered and the start time is recorded in milliseconds by calling the function (System.currentTimeMillis( )).

(2) The ciphertext is decrypted using one of the encryption algorithms proposed in this research.

(3) After completing the ciphertext decoding process, the final time is recorded in milliseconds by calling the same function (System.currentTimeMillis( )).

(4) The total time required to decrypt the ciphertext with any of the proposed encryption algorithms is calculated as illustrated in equation (3.2).

Stored Decryption Time = "End Time – Start Time" …….(3.2)

Where :

End Time = Time recorded after the decryption process is completed.

Start Time = Time recorded before the decryption process was started.

Stored Decryption Time = Total time required to complete the decryption process.

The previous four steps were repeated using a loop of 100 cycles to increase the accuracy of the calculated time required to decode the ciphertext.

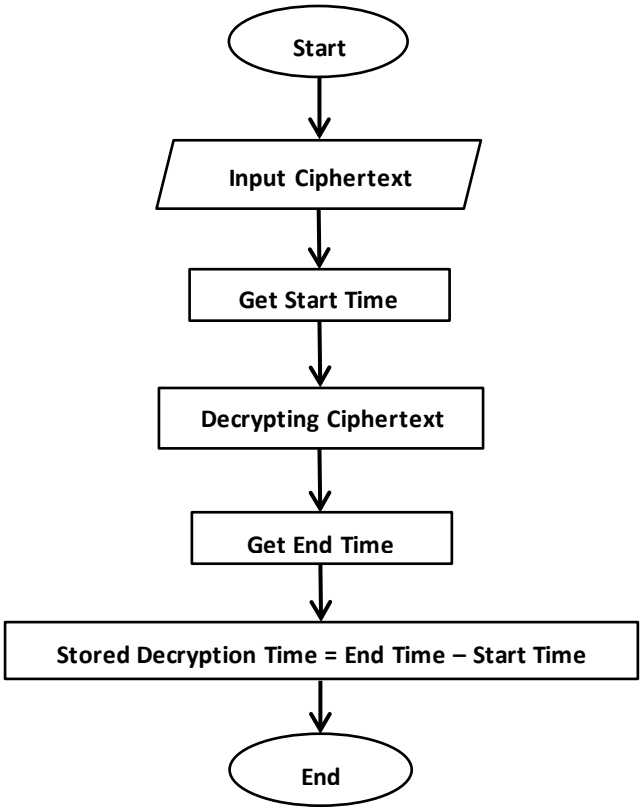

Figure (3.3) : Decryption Time.

3.5 Randomness of Security Algorithms

The second criterion for evaluating the encryption algorithms proposed in this research is related to the randomness of the resulting ciphertext, which were calculated using the Diehard Tests program as illustrated in figure (3.4) and according to the following steps:

(1) In the first step, the ciphertext generated by the proposed encryption algorithms is fed into the Diehard test program.

(2) Diehard test program performs set of different statistical tests to produce 215 p-values for each ciphertext.

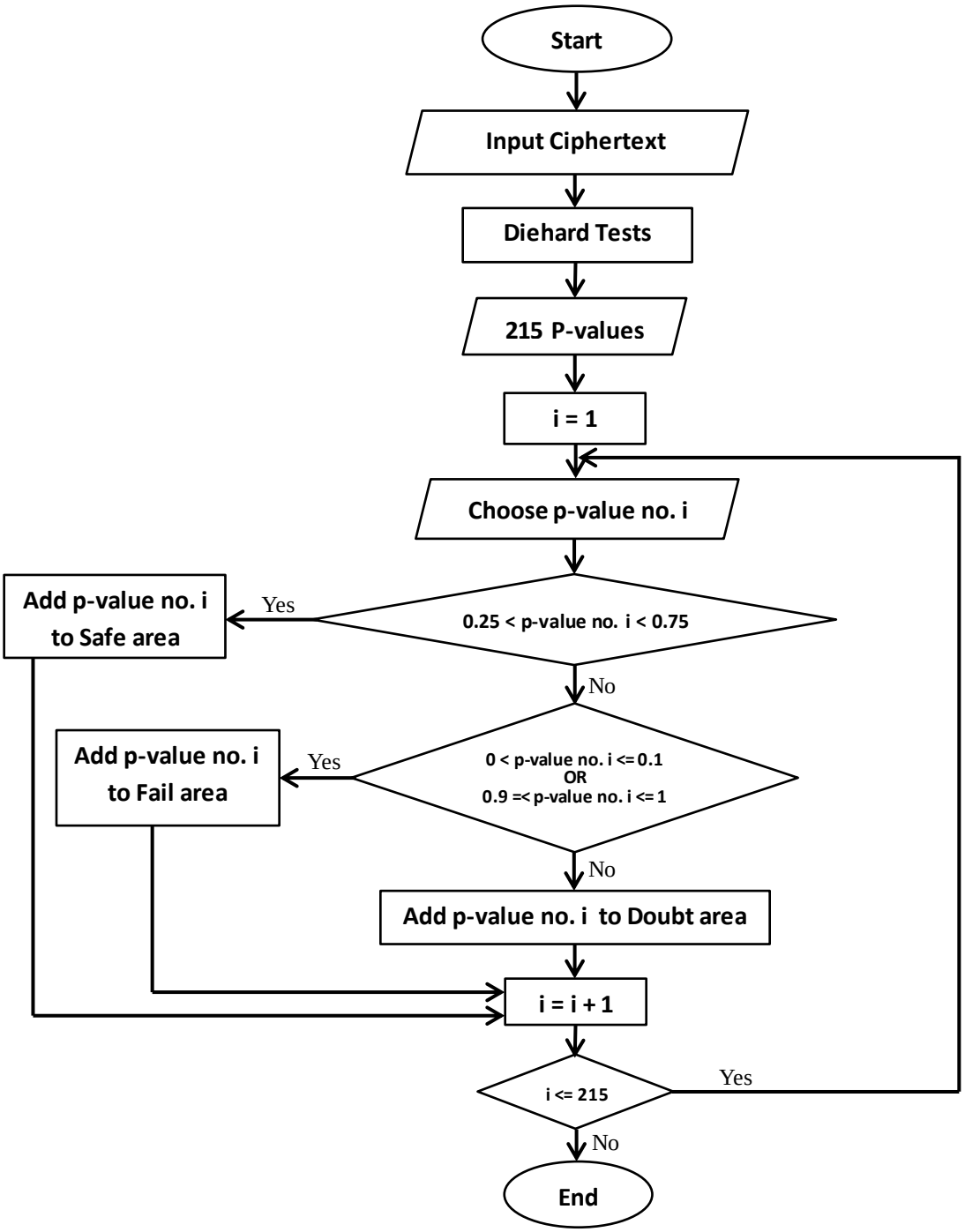

Figure (3.4) : P-values Classification from Diehard Tests.

(3) All p-values are belong to the interval [0,1) and classified according to their values into three areas: safe, failure and doubt areas, as illustrated in table (3.2).

Table (3.2) : Bounds of Safe, Failure, and Doubt areas

Safe region	"0.25 < p – value < 0.75"
Fail region	"0 < p – value ≤ 0.1 OR 0.9 ≤ p – value ≤ 1"
Doubt region	"0.1 < p – value ≤ 0.25 OR 0.75 ≤ p – value < 0.9"

When most of the p-values are included in the safe region then the tested ciphertext has better randomness, while when the fail region includes most of the p-values, then this means the tested ciphertext deviates from randomness.

Chapter Four : System Implementation and Results

4.1 Introduction

This chapter highlight two important parts, the first part relates to the implementation mechanism of the system, the second part relates to the results obtained from the execution of the proposed research, and at the end of this chapter a comparison made between the proposed symmetric encryption algorithms (AES, DES, 3DES, RC4, Blowfish, Twofish) are within the security mode (CBC) in terms of the evaluation criteria that have been adopted in this research.

4.2 System Requirements

This proposed research was designed and implemented based on the basic requirements, and these requirements are divided into hardware requirements and software requirements.

4.2.1 Hardware Requirement

Specifications of the computer through which the system is implemented according to the following points: Device name Laptop-Lenovo ThinkPad , installed RAM 8.00 GB, "system type 64-bit operating system, x64-based processor, processor intel(R) core(TM) i7-4700 MQ CPU @ 2.40GHz 2.39 GHz" .

4.2.2 Software Requirement

All operations of this proposed research for encoding and decoding plaintext as well as calculating the execution time were implemented in Java using IntelliJ IDE 2021 programming language compatible with Windows 10 pro. For calculating the ciphertext randomness, the Diehard Test program and Microsoft Excel 2010 was used.

4.3 System Implementation

In this research two main tests were implemented :

4.3.1 Cipher / Decipher Tests Time

This part explains the main interface used for the encrypting and decrypting of plaintext, as well as how the encryption and decryption time is calculated.

4.3.1.1 Selection of an encryption and decryption algorithm

The first step in this test, select the plaintext and selection one of the proposed symmetric encryption algorithms, as shown in figure (4.1).

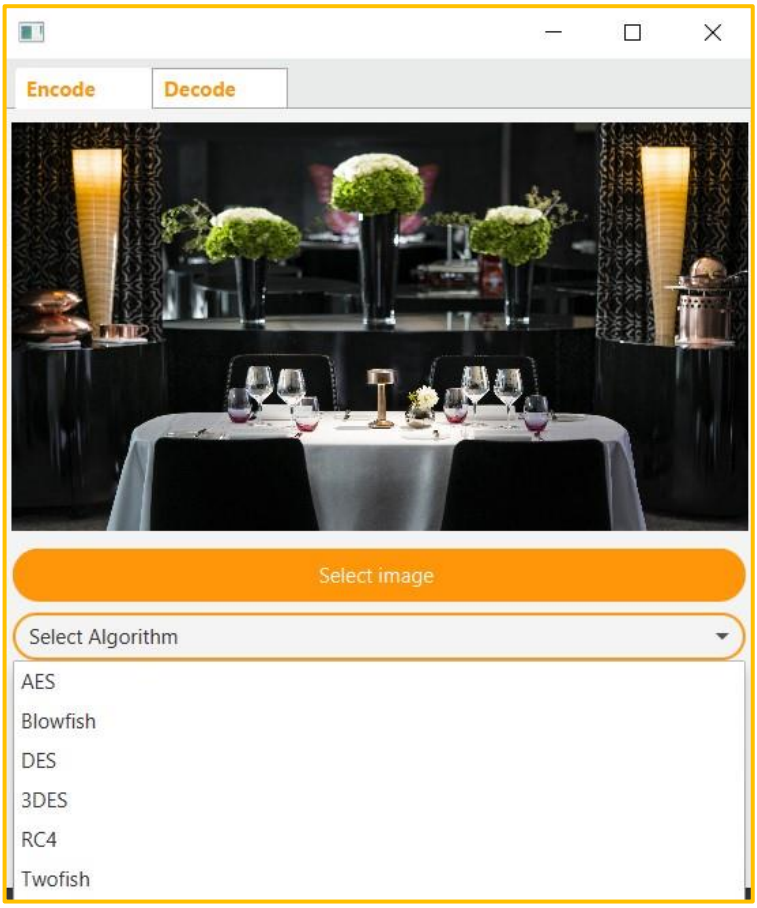

Figure (4.1) : Selecting one of the proposed algorithms.

4.3.1.2 Cipher Test Time

For the purpose of starting the encrypting of the plaintext as well as calculating the encrypting time, the Encode command is selected as shown in figure (4.2).

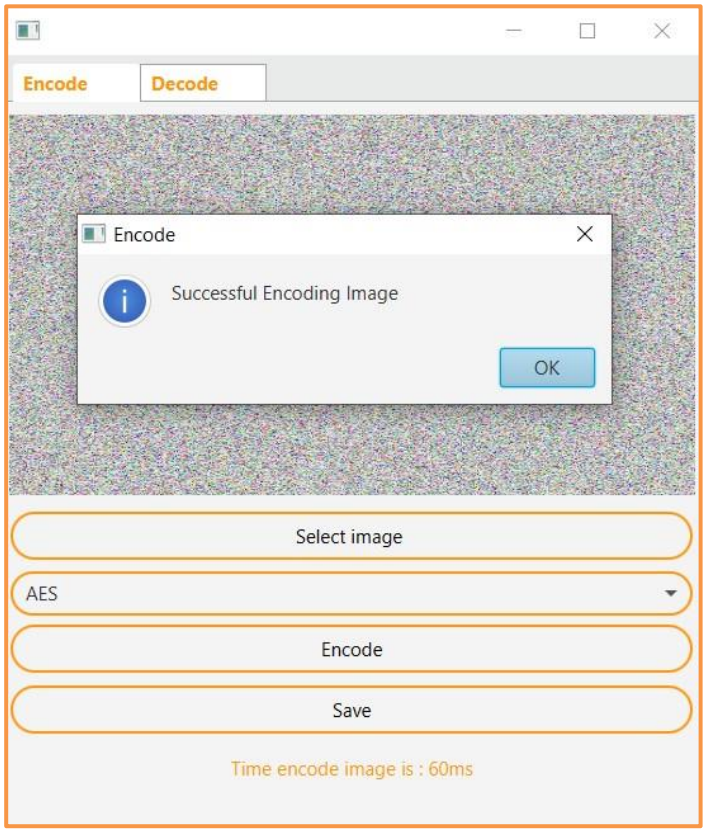

Figure (4.2) : Encrypting Test.

To store the resulting ciphertext, the command (Save) is selected, as illustrated in figure (4.2).

4.3.1.3 Decipher Test Time

As for starting the decryption of the ciphertext as well as calculating the decryption time, first the ciphertext resulting from the previous process is selected as shown in figure (4.3).

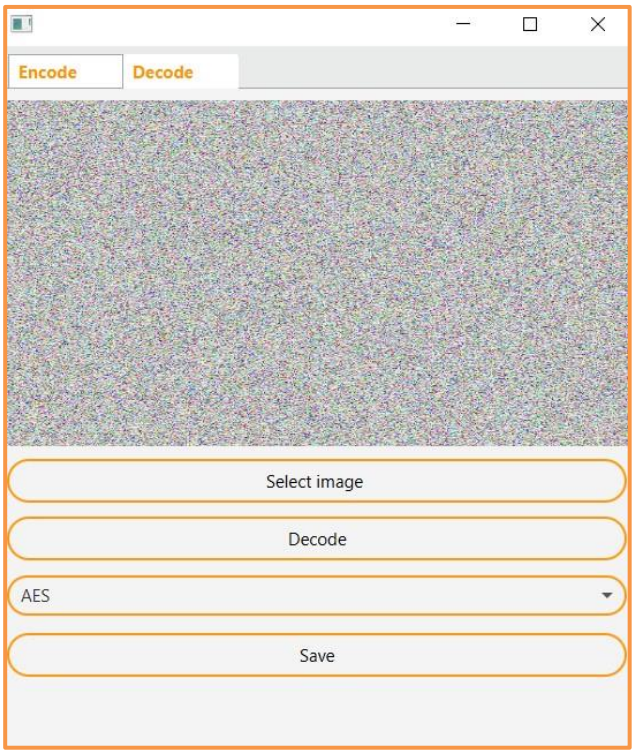

Figure (4.3) : Decrypting Test.

To display the plaintext after decryption in this window, in addition to the total time required for decryption, the command (Decode) is selected as shown in figure (4.4).

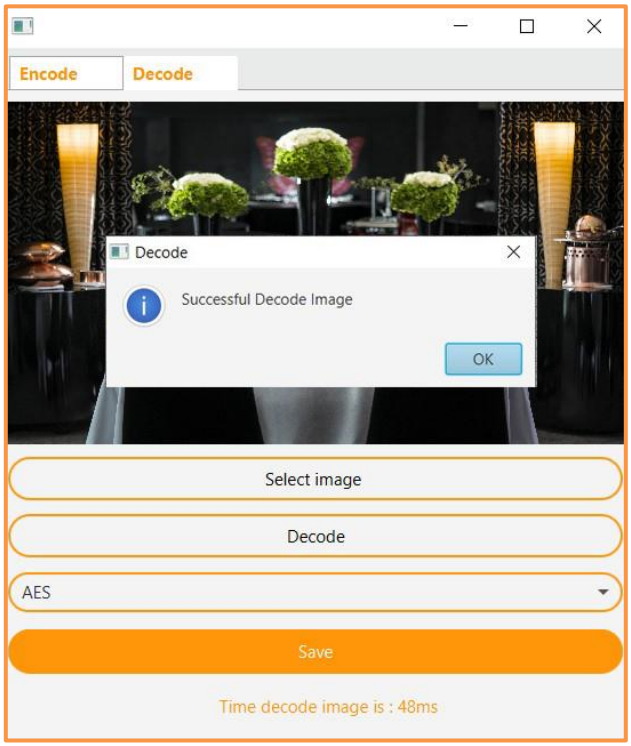

Figure (4.4) : Computed Result of Decrypting Time Test.

To store the resulting plaintext, the command (Save) is selected, as shown in figure (4.4).

The same previous three steps are repeated with the rest of the proposed algorithms to calculate the encryption and decryption time on the same plaintext used.

4.3.2 Randomness Test

This part discusses how to calculate the randomness of the encryption algorithms proposed in this research, using the Diehard Tests program, as shown in the following steps:

(1) The name of the ciphertext generated by the six proposed algorithms, as well as its extension, is entered into the Diehard Test program as shown in figure (4.5).

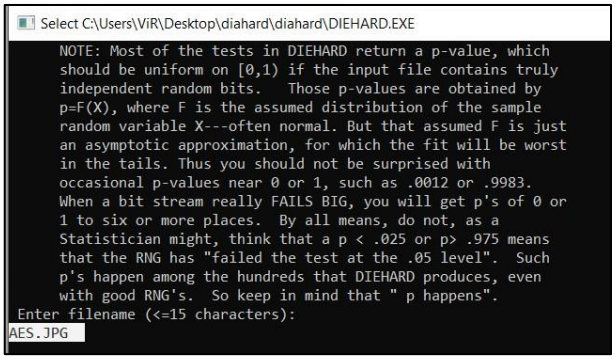

Figure (4.5) : First Step of Randomness Test

(2) A different name is chosen for the output file from this program, with the extension (.txt), as shown in figure (4.6).

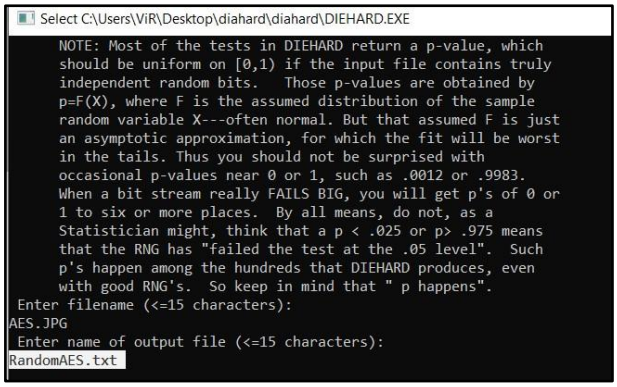

Figure (4.6) : Second Step of Randomness Test

(3) Diehard program contains set of different statistical tests, each test producing a set of p-values and for the purpose of activating all these required tests fifteen numbers are entered with the value one as shown in figure (4.7).

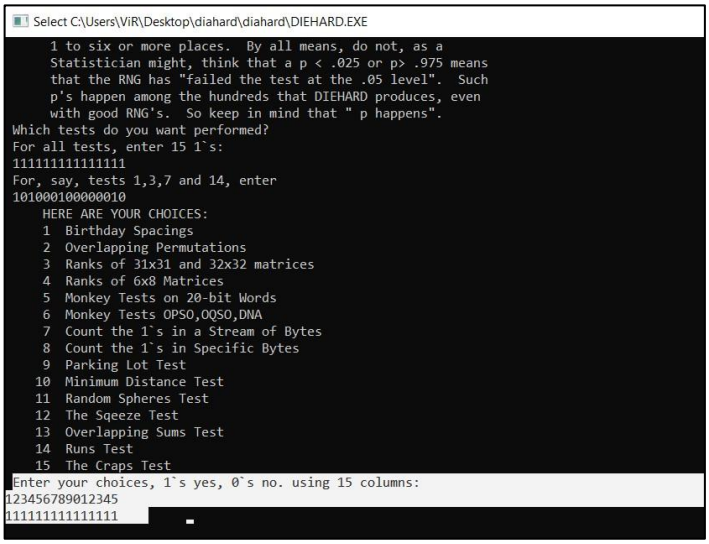

Figure (4.7) : Third Step of Randomness Test

4.4 Research Results

This part presents the results obtained after implementing the proposed research and consists of two section :

4.4.1 Results of Cipher / Decipher Tests Time

4.4.1.1 Encryption Test Time

As mentioned earlier, encryption time is the total time an algorithm needs to convert data from plaintext to ciphertext depending on the size of the data block and the length of the key used.

By applying the six symmetric encryption algorithms proposed in this research on a file with size 11.7MB and comparing these algorithms in terms of data

encryption time. The results of the research showed that the 3DES algorithm is the most time-consuming, preceded by DES, while RC4 is the algorithm that needs the least encryption time, followed by AES, and both Twofish and Blowfish came between these two levels as shown in figure (4.8).

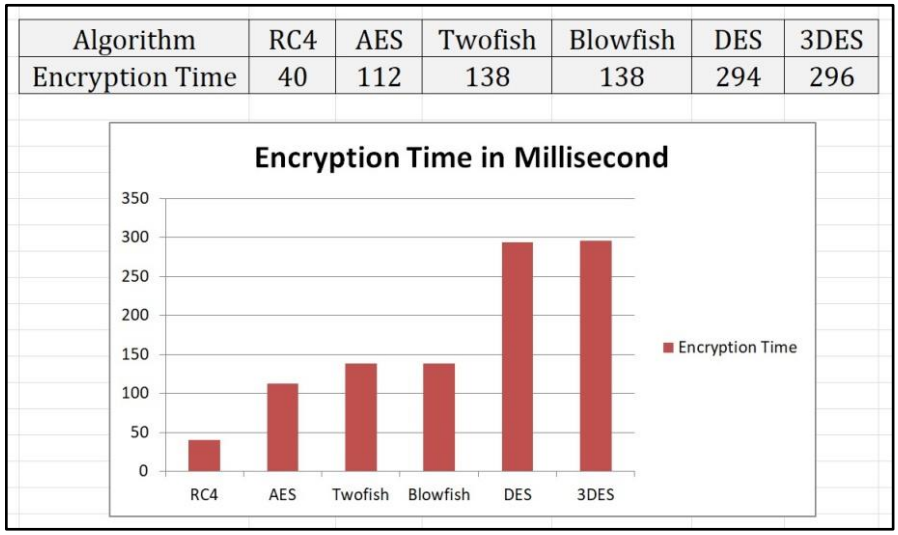

Figure (4.8) : Encryption Time for proposed algorithms.

4.4.1.2 Decryption Test Time

As mentioned earlier, the time required by the encryption algorithm for the purpose of converting the ciphertext to the original plaintext is called the decryption time, it is considered one of the important criteria for measuring the efficiency of the algorithm.

After implementing the proposed algorithms on the ciphertext and making a comparison based on the decryption time, the results showed that RC4 is the least time consuming, followed by AES, while 3DES is the most time consuming, and it is preceded by DES with a very close percentage. As for the two algorithms, Blowfish and Twofish, they came between these two levels, and the decoding time is very close between them, as shown in figure (4.9).

Algorithm	RC4	AES	Twofish	Blowfish	DES	3DES
Decryption Time	41	109	155	156	312	314

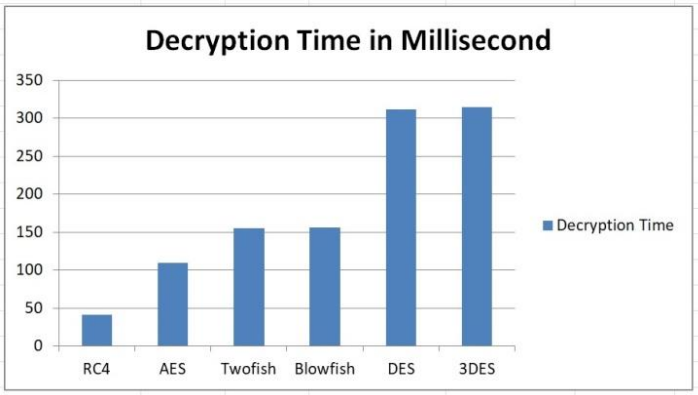

Figure (4.9) : Decryption Time of the proposed algorithms.

4.4.2 Results of Randomness Test

Using Diehard tests on the ciphertexts resulting from the proposed algorithms, 215 p-values were obtained for each tested algorithm, which was distributed according to their values into three areas: the safe area, the failure area, and the doubt area as shown in the steps below:

(1) For the AES algorithm with 128 bits key length, the randomness of the ciphertext as exhibited in the figure (4.10).

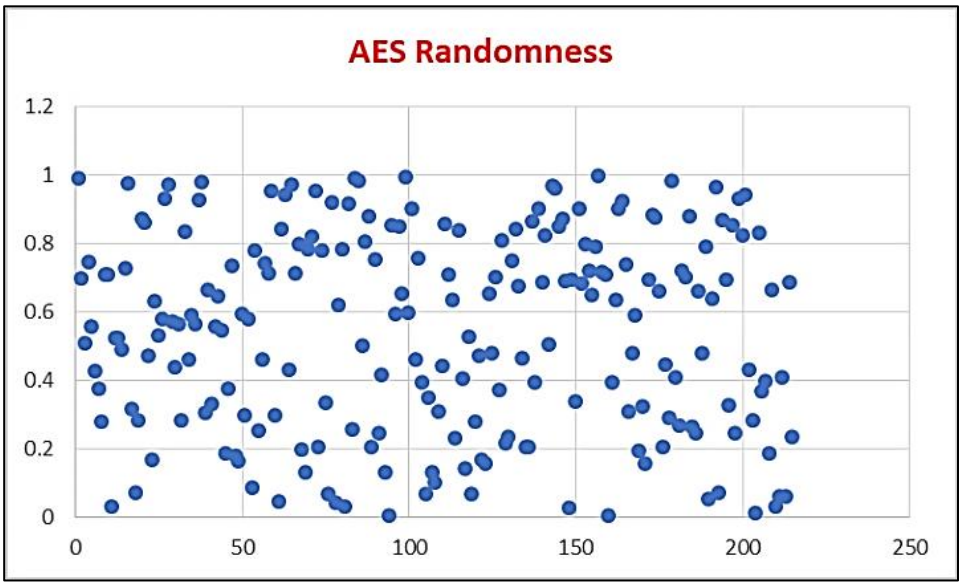

Figure (4.10) : P-values For AES – Ciphertext.

The number of p-value in the safety region was higher than the number of the doubt and almost twice the number of the failure regions, and also the number of p-value in the doubt region was more than the number in the failure region as exhibited in the figure (4.11).

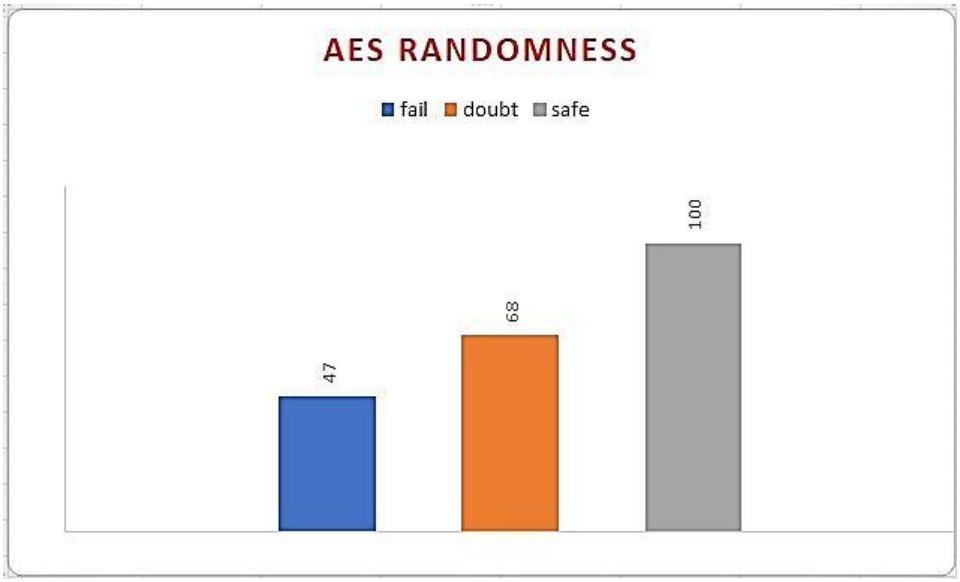

Figure (4.11):Distributed p-value over the safe, doubt,and fail area for AES –Ciphertext.

(2) For the DES algorithm with 64 bits key length, the randomness of the ciphertext as exhibited in the figure (4.12).

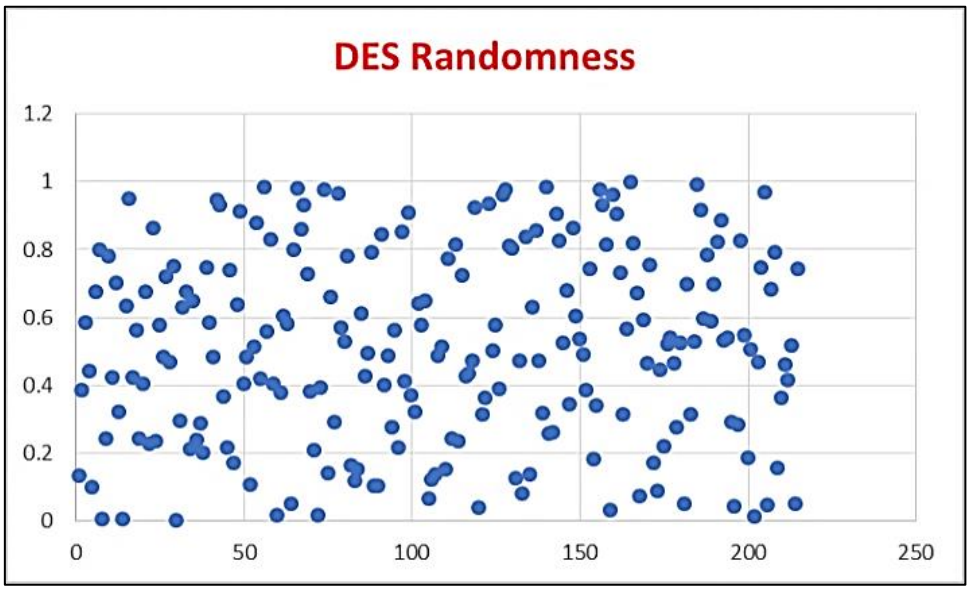

Figure (4.12) : P-values For DES – Ciphertext.

The greater proportion of p-values (Almost half of the total number) tends to be in the safety region and much more than the doubt and failure regions. Also, the numbers of p-value in the doubt region were more than in the failure region as shown in figure (4.13).

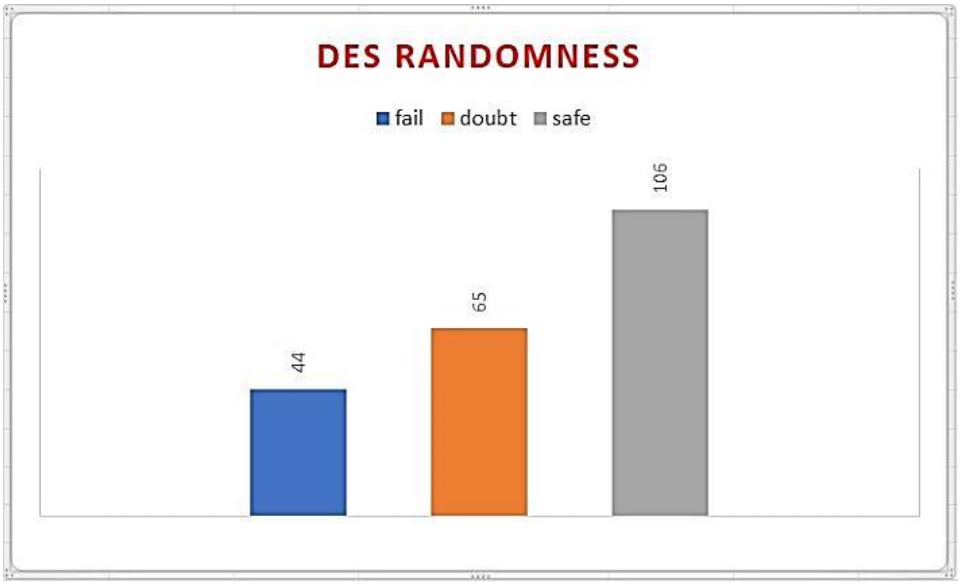

Figure (4.13):Distributed p-value over the safe, doubt, and fail area for DES – Ciphertext.

(3) For the 3DES algorithm with 112 bits key length, the randomness of the ciphertext as shown in figure (4.14).

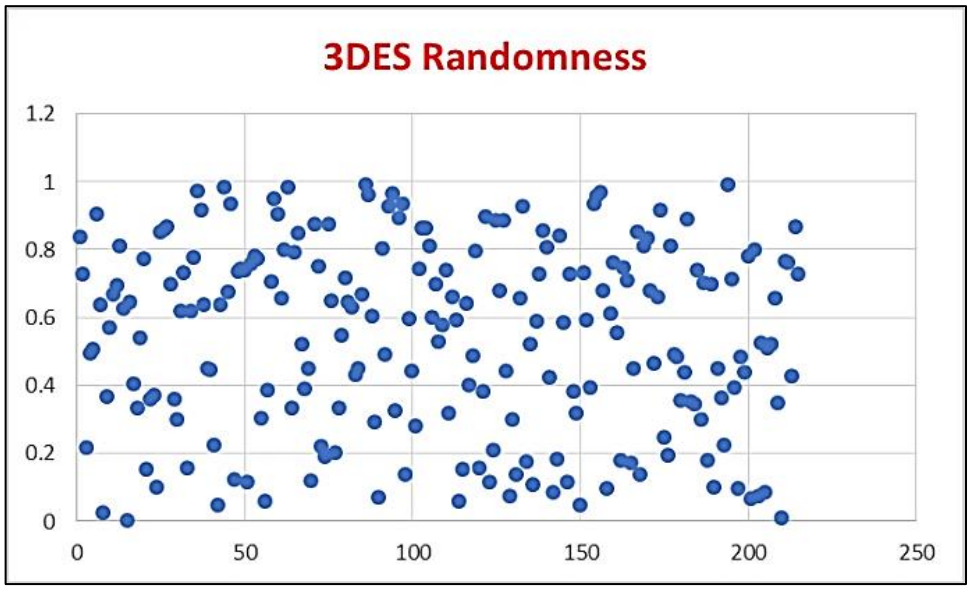

Figure (4.14) : P-values For 3DES – Ciphertext.

As in the previous two algorithms, the ratio of p-value in the safety region was significantly more than the doubt and failure regions, which were close in number despite the slight increase in the doubt region as shown in figure (4.15).

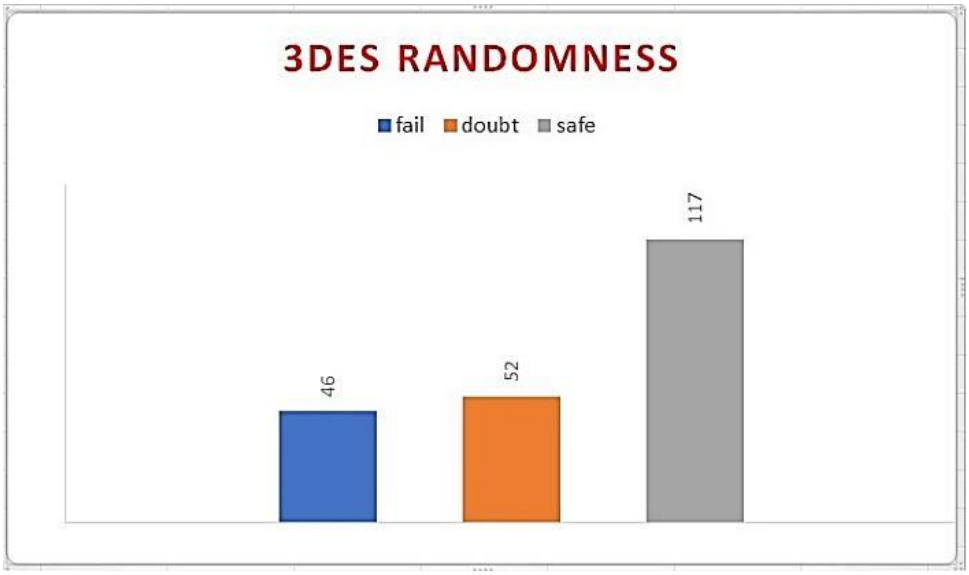

Figure (4.15) : Distributed p-value over the safe,doubt, and fail area for 3DES – Ciphertext.

(4) For the RC4 algorithm with 128 bits key length, the randomness of the ciphertext as shown in figure (4.16).

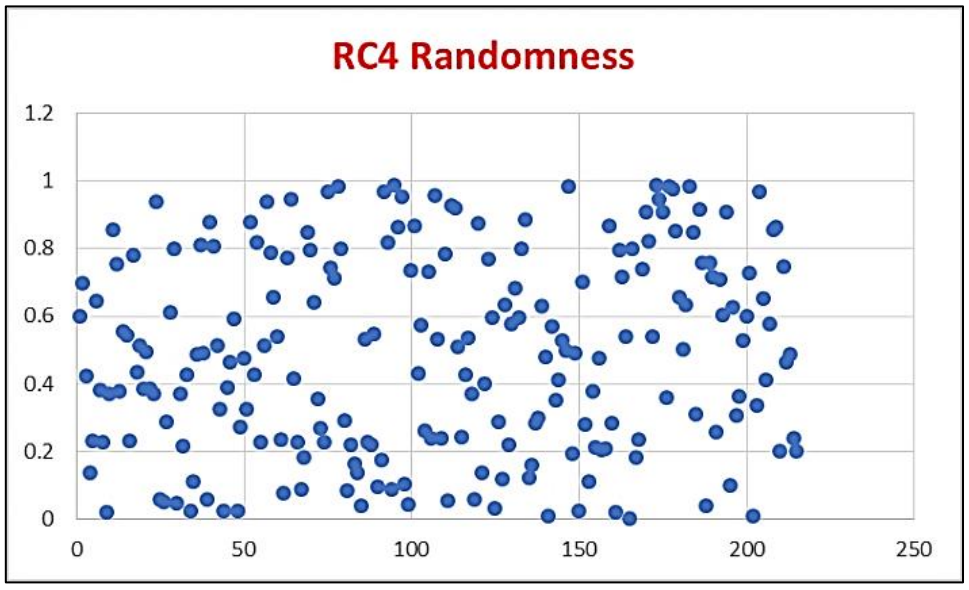

Figure (4.16) : P-values For RC4 – Ciphertext.

Here, the number of p-value in the safety region was almost twice as many as in the failure region and also more than the number in the doubt region. As for the numbers of p-value in the doubt region, it was significantly more compared to the failure region as shown in figure (4.17).

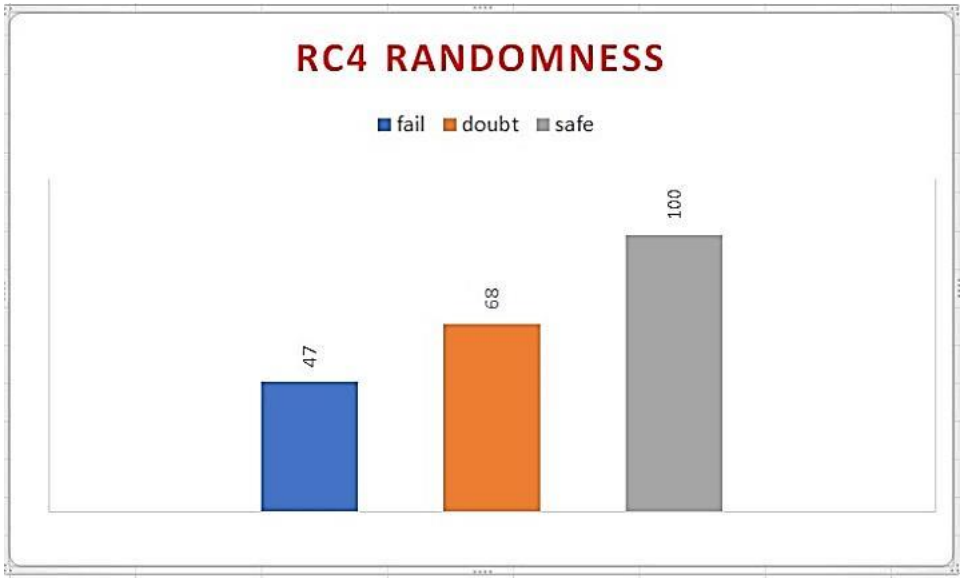

Figure (4.17) : Distributed p-value over the safe, doubt, and fail area for RC4 – Ciphertext.

(5) For the Blowfish algorithm with 256 bits key length, the randomness of the ciphertext as shown in figure (4.18).

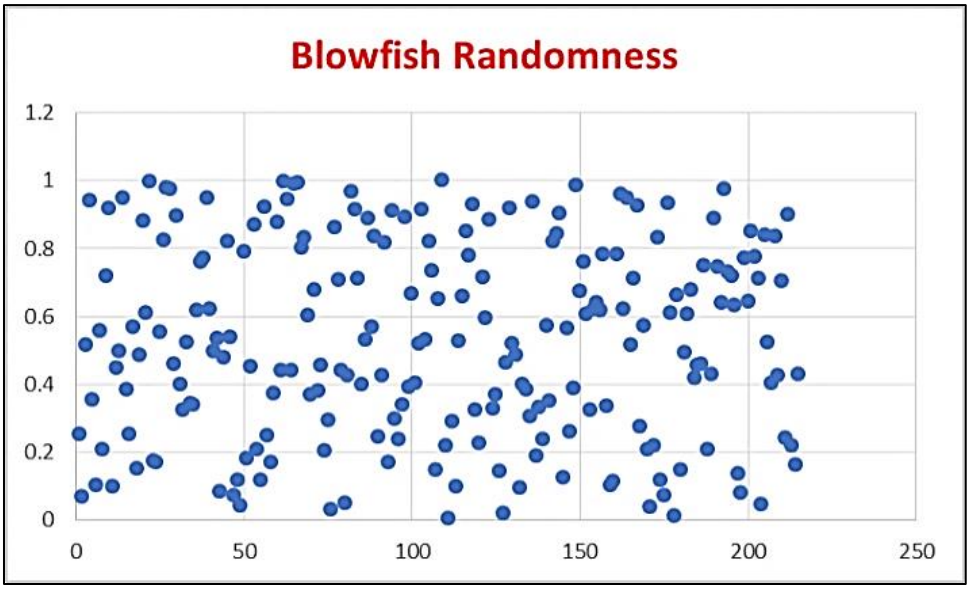

Figure (4.18) : P-values For Blowfish – Ciphertext.

In this algorithm, the number of p-value in the safety region was almost twice as many as in the doubt region, which in turn had a few more p-value numbers than what was in the failure region as shown in figure (4.19).

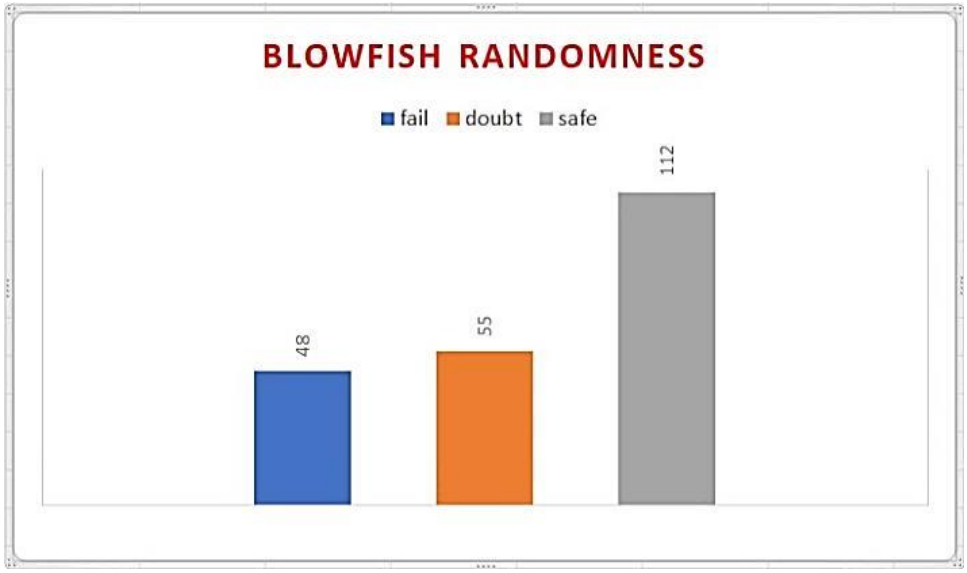

Figure (4.19):Distributed p-value over the safe,doubt,and fail area for Blowfish–Ciphertext

(6) For the Twofish algorithm with 256 bits key length, the randomness of the ciphertext as shown in figure (4.20).

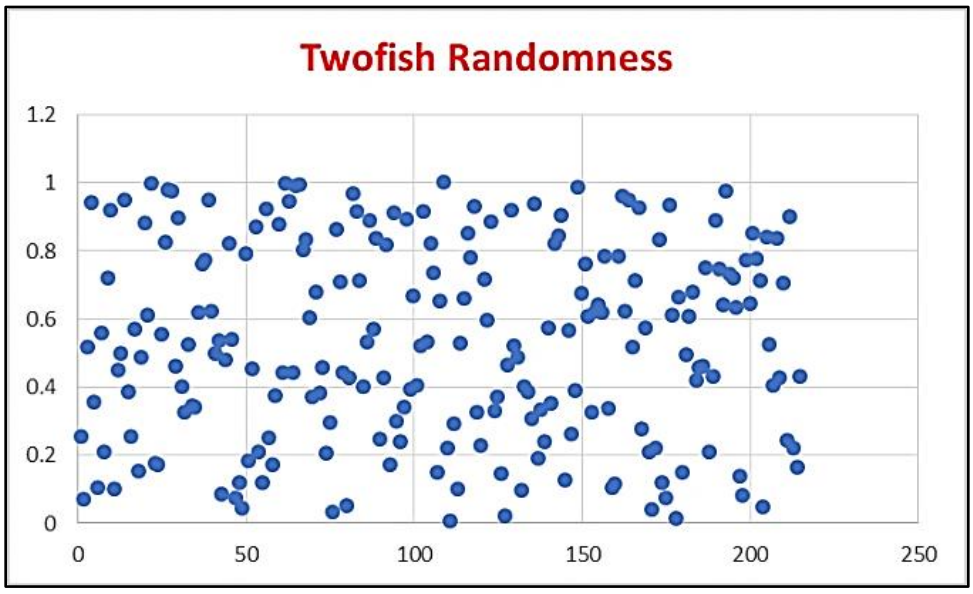

Figure (4.20) : P-values For Twofish – Ciphertext.

In this algorithm, all p-value numbers in the three regions were exactly the same as what is found in Blowfish's algorithm, as shown in figure (4.21).

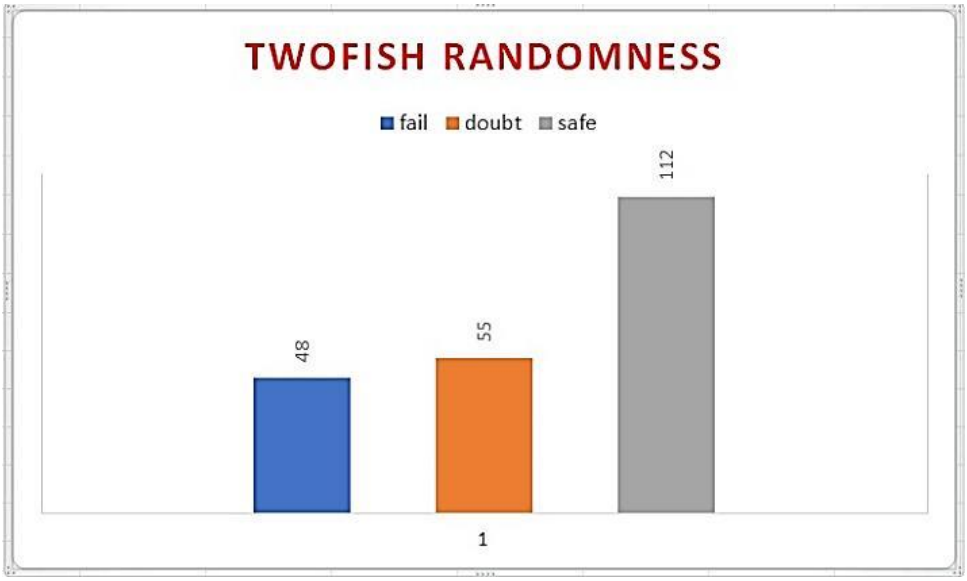

Figure (4.21):Distributed p-value over the safe,doubt, and fail area for Twofish–Ciphertext.

(7) As for the randomness of the original file, it is shown in figure (4.22).

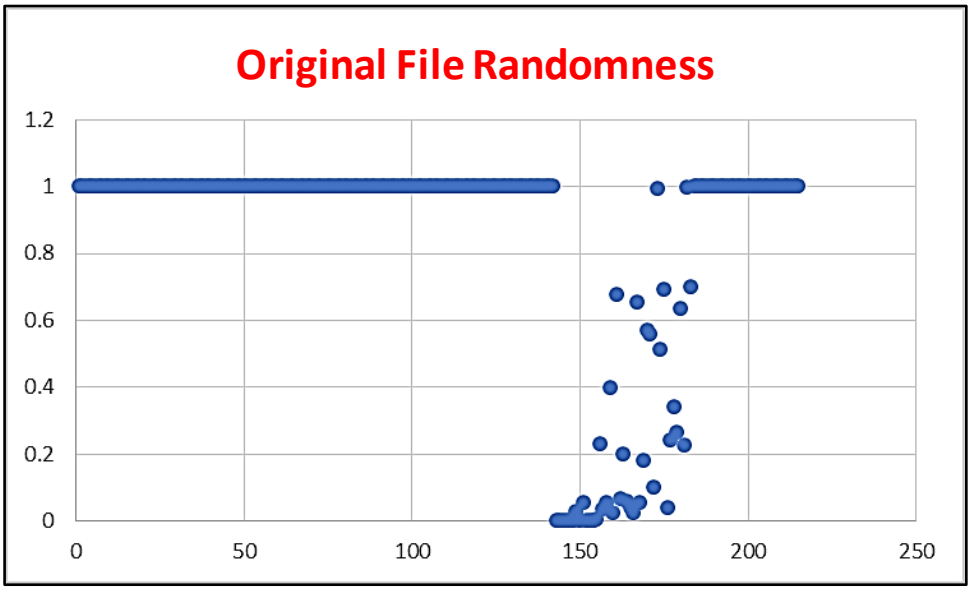

Figure (4.22) : P-values For the Original File.

Most of the p-value numbers were in the failure region except for a very small percentage that was divided between the safety and doubt regions. The number of p-value in the safety region was twice the number in the doubt region, as shown in figure (4.23).

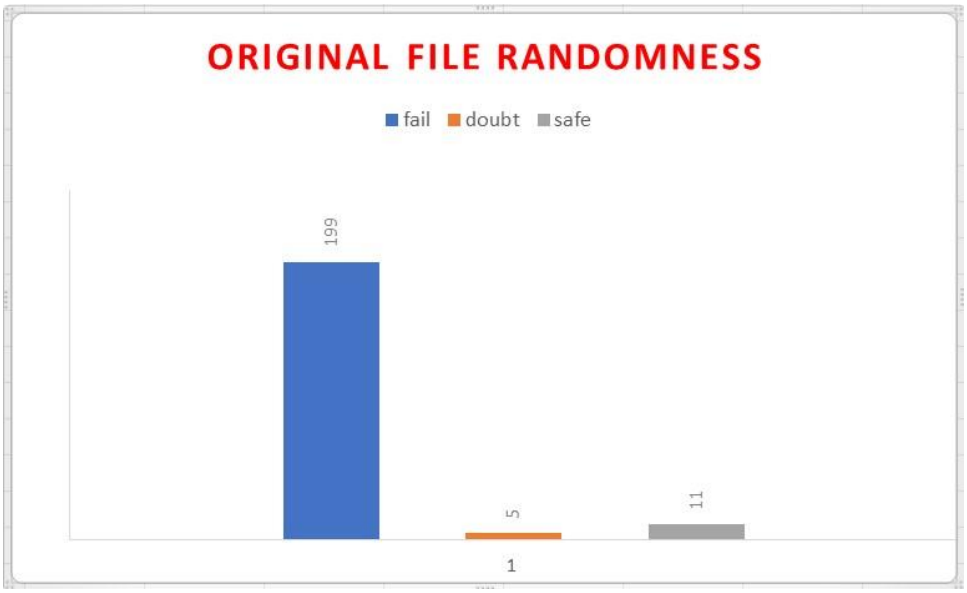

Figure (4.23) : Distributed p-value over the safe, doubt, and fail area for the Original File.

It is clear from the above results that the original file had the lowest level of randomness criterion as it contained the majority of p-value within the failure region, but by using the proposed encryption algorithms, the randomness percentage was clearly increased in the resulting ciphertexts.

And when comparing the proposed six algorithms according to the randomness criterion, the results were that 3DES contained the most numbers of p-value within safety region in addition to the least numbers of p-value within doubt region compared to all the proposed algorithms, while all numbers of p-value within failure region and for all the proposed algorithms are very close. In the second rank came both Blowfish and Twofish, which contained identical numbers for p-values and for all three areas. DES followed and finally came the AES and RC4

algorithms with identical p-values for each of the three areas as shown in figure (4.24)..

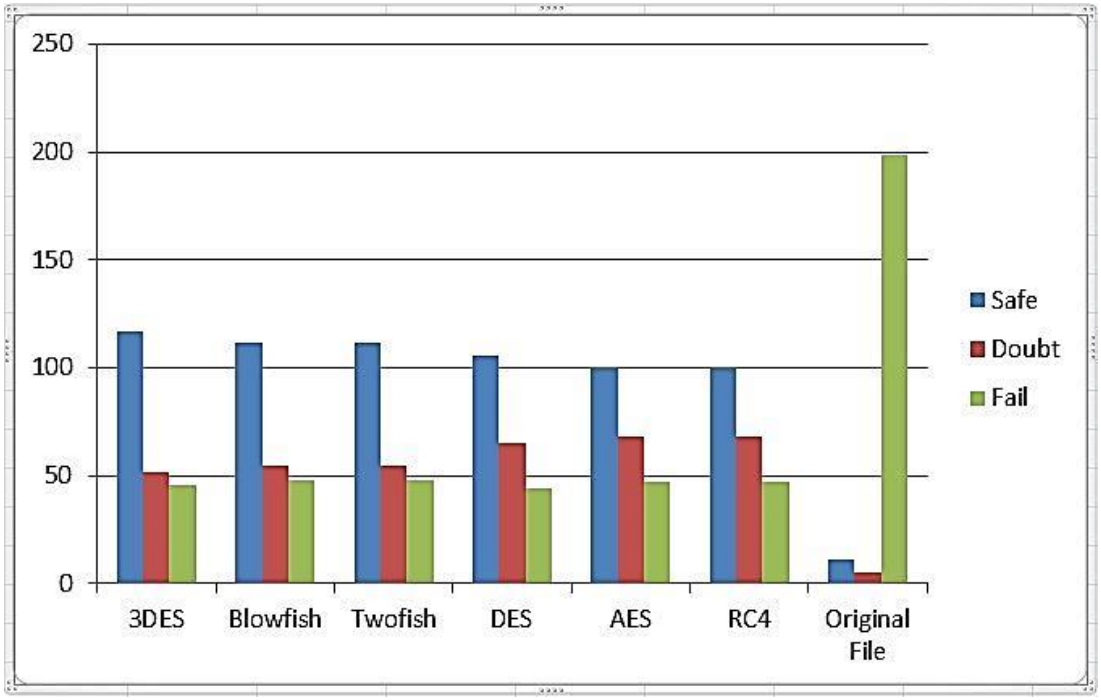

Figure (4.24) :Comparison between the six proposed algorithms in randomness criterion.

Chapter Five : Conclusions and Future Works

5.1 Introduction

Information security is the science that protects data from all security breaches and threats through a set of procedures, one of the most prominent of which is encryption. Encryption is defined as the process of sending data over insecure channels so that only the authorized person and owner of the secret key can view and read this encrypted data by one of the encryption methods. There are two main types of cipher: public cipher and private cipher.

This chapter discusses two parts, the conclusions related to the research and future works.

5.2 Conclusions

Through the comparison made in this research between the proposed symmetric algorithms, the following was concluded:

(1) The RC4 and AES algorithms have high performance as they are the least time-consuming to execute compared to the rest of the proposed algorithms.
(2) 3DES is better than AES, DES, RC4, Blowfish, Twofish in terms of randomness.
(3) There is an inverse relationship between the performance and randomness for test algorithms.
(4) The more complex algorithm, the more execution time it consumes for the data encryption and decryption.

5.3 Future Works

This research have suggested some future works to be adopted for further studies :

(1) Suggesting a comparison between a number of asymmetric encryption algorithms such as (RSA and Elliptic Curve Cryptography (ECC)).

(2) Adopting more than one security mode among the proposed algorithms such as (ECB, CTR) for the purpose of comparing the results.

(3) Adopting different evaluation criteria in comparing the proposed algorithms, such as (Entropy value, Memory used, Avalanche effect).

References

[1] H. A. M. Abu Ghali, "Novel Hybrid Cryptosystem Based on Quasi Group, Chaotic and ElGamal Cryptography," 2011.

[2] A. Fenyi, J. G. Davis, and K. Riverson, "Comparative analysis of advanced encryption standard, blowfish and rivest cipher 4 algorithms," *International Journal of Innovative Research and Development,* vol. 3, no. 11, 2014.

[3] R. Venkateshwarlu and J. Ramalingam, "Comparison of DES, AES, Blowfish and Twofish Symmetric Key Cryptography Algorithms," *International Journal of Innovative Research in Science Engineering and Technology,* vol. 6, pp. 639-647, 03/01 2019.

[4] j. h. c. Awotunde Joseph Bamidele, A. Oloduowo, I. D. Oladipo, R. Tomori, and M. AbdulRaheem, "Evaluation of Four Encryption Algorithms for Viability, Reliability and Performance Estimation," *Nigerian Journal of Technological Development,* vol. 13, p. 74, 03/13 2017, doi: 10.4314/njtd.v13i2.5.

[5] A. Ghosh, *Comparison of Encryption Algorithms: AES, Blowfish and Twofish for Security of Wireless Networks*. 2020, doi: 10.13140/RG.2.2.31024.38401

[6] B. Nithya and S. Priya, "Comparative Analysis of Symmetric Cryptographic Algorithms on .Net Platform," *Indian Journal of Science and Technology,* vol. 9, 07/28 2016, doi: 10.17485/ijst/2016/v9i27/86580.

[7] M. Ahmad and R. P. Singh, "A Survey on Comparison of Various Encryption Algorithms for secured data Communication," 2019, Print ISSN: 2395-1990 | Online ISSN : 2394-4099

[8] M. T. Gençoğlu, "Importance of Cryptography in Information Security," 03/10 2019, doi: 10.9790/0661-2101026568.

[9] A. Menezes, P. Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*. 1996, pp. 1-780, ISBN 9780849385230.

[10] M. J. Robshaw, "Stream ciphers," *RSA Labratories,* vol. 25, 1995.

[11] S. M. A. Elkourd, "Data Encryption Using the Dynamic Location and Speed of Mobile Phone," 2010.

[12] A. Abdullah, "Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt Data," 06/16 2017.

[13] B. Y. Abusalim, "An Efficient Approach For Data Encryption Using Two Keys," 2015.

[14] I. Saikumar, "DES-Data Encryption Standard," *International Research Journal of Engineering and Technology,* vol. 4, no. 3, 2017.

[15] C. Paar and J. Pelzl, *Understanding cryptography: a textbook for students and practitioners*. Springer Science & Business Media, 2009.

[16] A. Sari, E. Rachmawanto, and C. Haryanto, "Cryptography Triple Data Encryption Standard (3DES) for Digital Image Security," *Scientific Journal of Informatics,* vol. 5, pp. 105-117, 11/29 2018, doi: 10.15294/sji.v5i2.14844.

[17] A. P. U. Siahaan, "RC4 Technique in Visual Cryptography RGB Image Encryption," 2017.

[18] A. P. U. Siahaan, "Blum Blum Shub in generating key in RC4," 2017.

[19] T. Nie and T. Zhang, *A study of DES and Blowfish encryption algorithm*. 2009, pp. 1-4.

[20] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, and N. Ferguson, "Two sh: a 128-bit block cipher," *AES submission,* 1998.

[21] S. K. Gil, S. H. Jeon, J. R. Jung, and N. Kim, "Optical design of cipher block chaining (CBC) encryption mode by using digital holography," in *Practical Holography XXX: Materials and Applications*, 2016, vol. 9771: International Society for Optics and Photonics, p. 97710Y.

[22] M. Dworkin, "Recommendation for block cipher modes of operation. methods and techniques," National Inst of Standards and Technology Gaithersburg MD Computer security Div, 2001.

[23] G. Simmons, "Symmetric and Asymmetric Encryption," *ACM Comput. Surv.*, vol. 11, pp. 305-330, 12/01 1979, doi: 10.1145/356789.356793.

[24] S. J. Muhammad, H. Chiroma, and M. Mahmud, "Cryptanalytic attacks on Rivest, Shamir, and Adleman (RSA) cryptosystem: issues and challenges," *Journal of Theoretical and Applied Information Technology*, vol. 61, no. 1, 2014.

[25] H. Alajbegović, D. Zečić, and H. Jamak, *DIGITAL SIGNATURE ALGORITHM (DSA)*. 2006.

[26] M. E. MANAA and R. H. JWDHA, "A PROACTIVE DATA SECURITY SCHEME OF FILES USING MINHASH TECHNIQUE," *Journal of Theoretical and Applied Information Technology*, vol. 96, no. 24, 2018.

[27] M. M. Alani, "Testing randomness in ciphertext of block-ciphers using DieHard tests," *Int. J. Comput. Sci. Netw. Secur*, vol. 10, no. 4, pp. 53-57, 2010.

[28] J. Bellamy, "Randomness of D sequences via diehard testing," *arXiv preprint arXiv:1312.3618*, 2013.

Appendix(A) Diehard Tests

- **Birthday spacing** : select sites at random from a wide range of possibilities. The distances between the points should be exponentially distributed asymptotically. The birthday paradox inspired the name.
- **Overlapping permutations** : Examine five different random number concatenations. in a row. The 120 potential orderings should have an equal statistical likelihood of occurring.
- **Ranks of matrices** : To generate a matrix with values ranging from 0 to 1, select a large number of bits from a set of random numbers, then compute the matrix's rank. The rank count must be distributed in a specified way.
- **Monkey tests** : Bit sequences of any length can be referred to as "words." Count the number of phrases that overlap in a stream. The number of "terms" that do not appear should be evenly distributed. The term comes from the endless monkey theorem.
- **Count the ones** : Count the number of one-bits in each of the bytes or a random sample. Convert the counts to "letters," and then count how many five-letter "words" appear.
- **Parking lot test** : In a 100*100 square, arrange unit circles at random. A circle is correctly parked if it does not overlap another properly parked circle. The amount of successfully parked circles should follow a specific pattern after 12,000 attempts.
- **Minimum distance test** : Determine the shortest path between 8000 random locations in a 10000*10000 grid. This distance's square should have an exponential distribution with a known mean.

• **Random spheres test** : Select 4000 random points from a cube with 1000 edges. Place a sphere with a radius equal to the shortest distance between two points at each point. The volume of the smallest sphere should be distributed exponentially with a defined mean.

• **The squeeze test** : hit 2^{31} by a random float from 0 to 1 until you reach 1. This should be repeated a100000 times over . The number of floats required to reach 1 should be distributed in a specific order.

• **Overlapping sums test** : Make a long string of random floats between range 0 and 1. Make a column of 100 float sequences. The sums should have an evenly distributed mean and variance.

• **Runs test** : Make a lengthy random series of [0,1] floats. Count the number of rising and descending runs. The counts must adhere to a strict pattern.

• **The craps test** : play 200,000 games of craps counting the wins and number of throws per game. each count should follow a certain distribution.

Ministry of Higher Education and Scientific Research
University of Babylon
College of Science for Women
Department of Computer Science

An Appropriate Security Mode for Data Ciphering in Life Applications

A Project Submitted to the College of Science / University of Babylon in Partial Fulfillment of the Requirements for the Degree of Higher Diploma in Computer Science

By

Mohammad Talib Hadi

Supervised by

Asst. Prof.Dr. Saif M. Kh. Al-Alak

2021 A.D. 1443 A.H

وزارة التعليم العالي والبحث العلمي

جامعة بابل / كلية العلوم للبنات

قسم علوم الحاسوب

وضع الأمان المناسب لتشفير البيانات في تطبيقات الحياة

بحث مقدم إلى كلية العلوم للبنات / جامعة بابل كجزء من متطلبات نيل درجة الدبلوم العالي في علوم الحاسوب

مقدم من قبل :

محمد طالب هادي

بإشراف :

أ. م. د. سيف محمود خلف العلاك

2021 م

1443 هـ

الخلاصة

يعد الأمن في الوقت الحاضر مهمًا جدًا وفعالًا للغاية لتطبيقات الإنترنت والشبكات ، والتي تنمو بسرعة ، وبالتالي زادت قيمة وأهمية البيانات التي يتم تبادلها عبر الإنترنت أو الوسائط الأخرى. لذلك تم اقتراح العديد من الخوارزميات القائمة على التشفير لتوفير الحماية المطلوبة ضد الهجمات. ومع ذلك ، تختلف هذه الخوارزميات في درجة العشوائية والوقت المستغرق لإنتاج نص مشفر آمن. يهدف هذا البحث إلى اختبار عدد من خوارزميات التشفير المتماثل (AES، DES ، 3DES ، RC4 ، Blowfish ، Twofish) مع وضع أمان تسلسل كتلة التشفير (CBC) ثم يتم إجراء مقارنة بينهم بناءً على معايير التقييم : اختبارات وقت التشفير وفك التشفير والتي يتم تنفيذها باستخدام لغة برمجة Java . اختبار العشوائية على النص المشفر ، والذي تم تنفيذه باستخدام برنامج Diehard Tests الإحصائي لحساب الخوارزمية الأكثر كفاءة لاستخدامها في تطبيقات الحياة المختلفة.

أظهرت نتائج البحث أن خوارزمية 3DES هي الأكثر استهلاكا للوقت ، تليها DES ، بينما RC4 هي الخوارزمية التي تحتاج إلى أقل وقت تنفيذ ، تليها AES ، وجاءت كل من Twofish و Blowfish بين هذين المستويين. أما بالنسبة لمعيار العشوائية ، فقد كان 3DES هي الأعلى مقارنة بباقي الخوارزميات ، بينما كانت كل من RC4 و AES الأدنى في هذا المعيار.