JOURNAL OF UNIVERSITY OF BABYLON
For Pure and Applied Sciences (JUBPAS)

ARTICLE
Vol.29; No.3. October-December | 2021

Symmetric Security Algorithms Implementation in CBC Mode

Mohammad Talib Hadi[1] **Saif Al-Alak[2]**

1College of Science for Women, University of Babylon, mohammad.hmood.gsci37@student.uobabylon.edu.iq, Babylon, Iraq.
2 College of Science for Women, University of Babylon, saif.mahmood@uobabylon.edu.iq, Babylon, Iraq.

Corresponding author Email: mohammad.hmood.gsci37@student.uobabylon.edu.iq ,

saif.mahmood@uobabylon.edu.iq

Received: 25/10/2021 **Accepted:** 29/11/2021 **Published:** 1/11/2021

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, 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 is one of the most interesting topics in security-related communities. This paper 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 paper 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 best compared to the rest of the algorithms, while RC4 and AES were the worst in this criterion.

Key words:

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

Citation:

Mohammad Talib Hadi[1], Saif Al-Alak[2]. Symmetric Security Algorithms Implementation in CBC Mode. Journal of University of Babylon for Pure and applied science (**JUBPAS**). October-December , 2021. Vol.29; No.3; p: 126-144

INTRODUCTION

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) below.

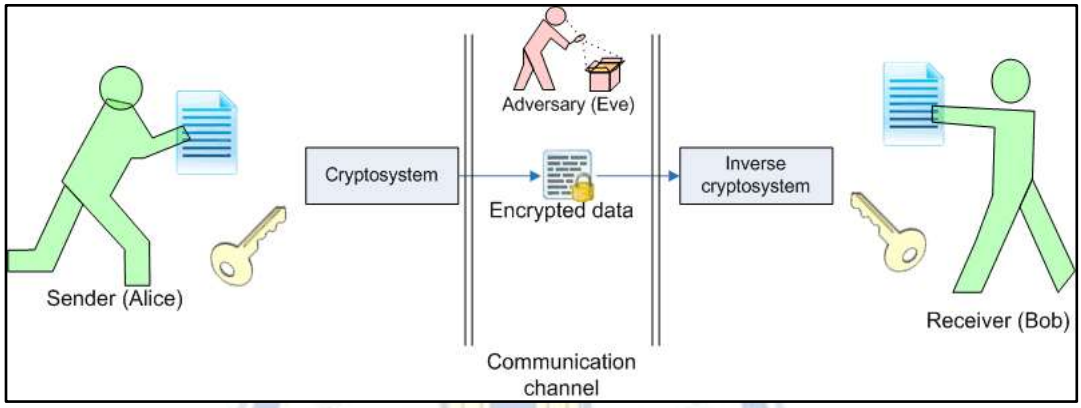


Figure (1) : Cryptosystem

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 paper, different symmetric security algorithms (SA) are tested in a specific security mode (SM) such as Cipher Block Chaining (CBC). The paper 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.

- **Related Studies**

In this part, highlight a number of works related to the topic of paper. 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.

In [3], R.venkateshwarlu 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] they consider with 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.

In [5] Archisman Ghosh presented 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) it can be implemented in the security of all network protocols along with HMAC.

In [6], B. Nithya and P. Sripriya 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.

Hasan Dibas, et al . in [8] conducted a performance comparison between four symmetric encryption algorithms: Blowfish, Twofish, 3DES, and AES. They evaluated memory usage, execution time, and text size in the encryption and decryption operations. The comparison was made with different file sizes by creating a .NET application using the C# language. Analyzing the results, they noticed that AES had the lowest execution time for both encryption and decryption, while Twofish had the highest execution time. Both AES and 3DES encryption uses less memory than Blowfish and Twofish and uses a very close amount of memory. While the AES algorithm uses less memory for decryption. Finally, Blowfish and Twofish have the largest ciphertext size.

- 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, here each partner can receive a copy of the shared key [9].

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 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 around the world as the Advanced Encryption Standard (AES) algorithm [10].

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. Encryption and decoding are done with the block cipher. Substitution, transposition, and other mathematical functions are used to put these blocks of bits together [11].

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 [12].

4) **Rivest Cipher 4 (RC4)**

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 [13].

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 [14]. 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.

5) **Blowfish Algorithm**

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 [15]. As the lengths of the keys greater than 128 bits , it can be a reliable encryption method.

6) **Towfish 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 that incorporates elements from the ciphers Khufu, Square, and SAFER. 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 [16].

- Cipher Block Chaining Mode (CBC)

Cryptography Security Mode (CBC) includes a feedback process for the ciphertext in the encryption process. 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 [17].

- Paper 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.

- Paper Objectives

Objectives of this paper are: 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. In addition to finding the best algorithm among the symmetric encryption algorithms (AES, DES, 3DES, RC4, Blowfish, Twofish) in the security mode (CBC) in terms of randomness.

Materials and Methods

1) **Paper Design**

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

In addition to six study cases will be implemented in this paper which are commonly used symmetric encryption algorithms (AES, DES, 3DES, RC4, Blowfish, Twofish) using Java language in security mode (CBC) as illustrated in figure (2).

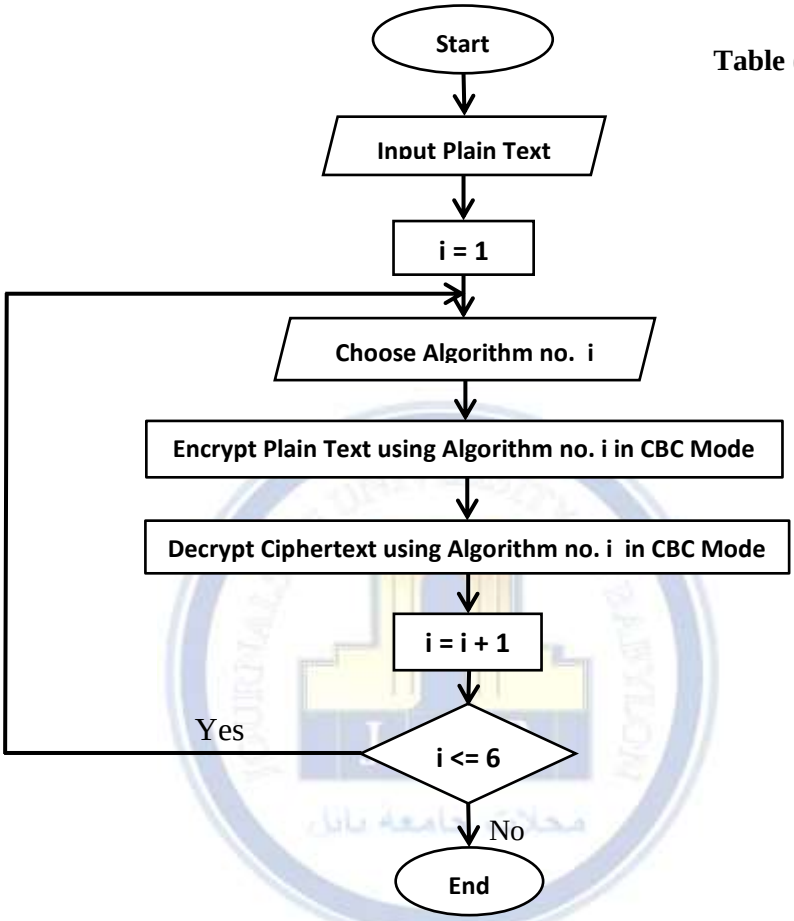


Figure (2) : Paper Design.

Table (1) : Sequence of Algorithm

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

2) Steps of Paper Design :

As illustrated in figure (2), the working mechanism of the paper design consists of the following steps:

(1) The first proposed algorithm (Algorithm no i) is chosen as illustrated in table (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 (1).

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

3) Ciphering / Deciphering Time

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

3.1 Encryption Time

The total execution time required for encryption the plaintext is calculated as in the following steps:

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

(2) The plain text is encrypted using one of the encryption algorithms proposed in this paper.

(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 (1).

Stored Encryption Time = "End Time – Start Time" …….…… (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.

3.2 Decryption Time

The total execution time required for ciphertext decoding is calculated as in the following steps:

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

(2) The cipher text is decrypted using one of the encryption algorithms proposed in this paper.

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

(4) The total time required to decrypt the cipher text with any of the proposed encryption algorithms is calculated as illustrated in equation (2).

Stored Decryption Time = "End Time – Start Time" …….(2)

Where :

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

Start Time = Time recorded before the encryption 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.

4) Randomness of Security Algorithms

The second criterion for evaluating the encryption algorithms proposed in this paper is related to the randomness of the resulting ciphertext, which were calculated using the Diehard Tests program as illustrated in figure (3) 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.

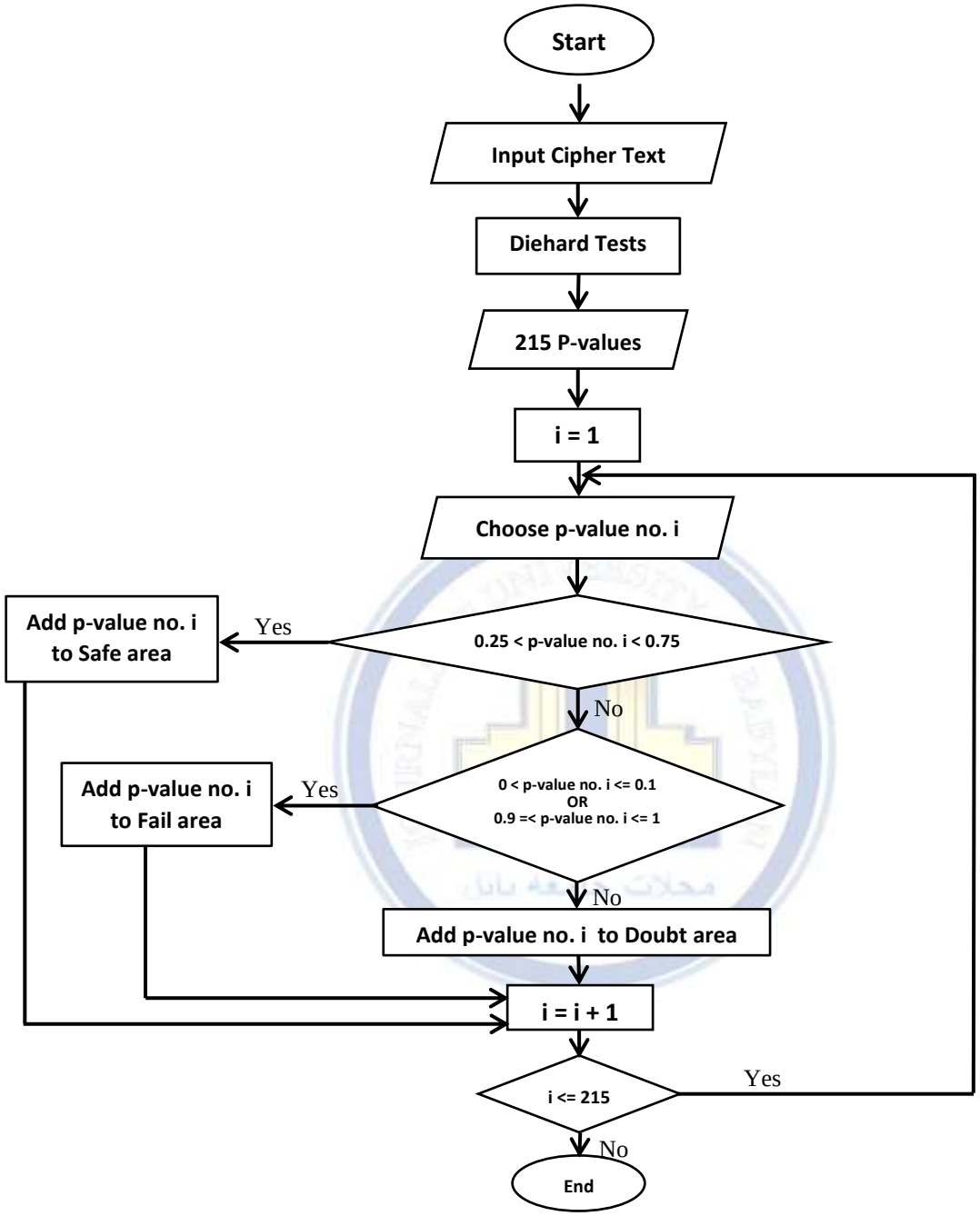


Figure (3) : P-values Classification from Diehard Tests.

(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 (2).

Table (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"

The higher the number of p-values in safe region, then this ciphertext and thus the proposed algorithm has better randomness, while the higher the number of p-values in fail region , it indicates that the proposed algorithm deviates from randomness.

Results and Discussion

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

A) Results of Cipher / Decipher Tests Time

1) Encryption Test Time

Encryption time is the total time an algorithm needs to convert data from plain text 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 paper on a file with size 11.7MB and comparing these algorithms in terms of data encryption time. The results of the paper showed that the 3DES algorithm is the most time-consuming, followed 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).

Algorithm	RC4	AES	Twofish	Blowfish	DES	3DES
Encryption Time	40	112	138	138	294	296

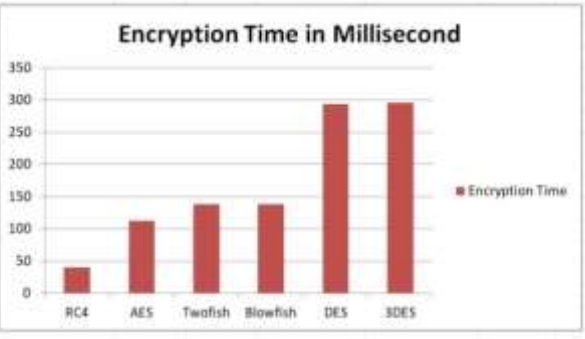


Figure (4) : Encryption Time for proposed algorithms.

2) Decryption Test Time

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 followed 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 (5).

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

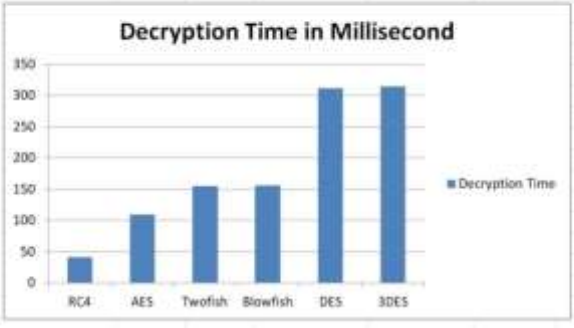


Figure (5) : Decryption Time of the proposed algorithms.

B) 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 (6).

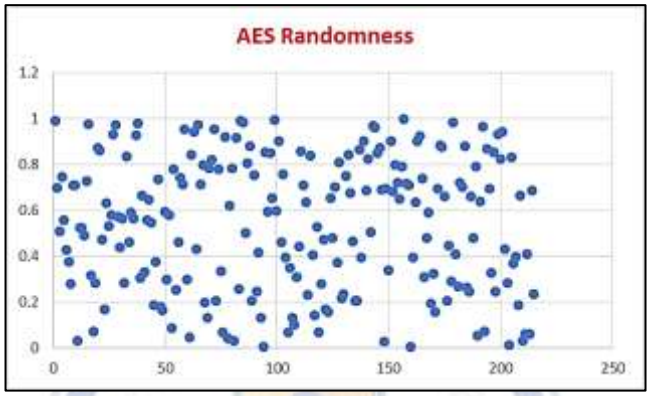


Figure (6) : P-values For AES – Ciphertext.

(2) For the DES algorithm with 64 bits key length, the randomness of the ciphertext as exhibited in the figure (7).

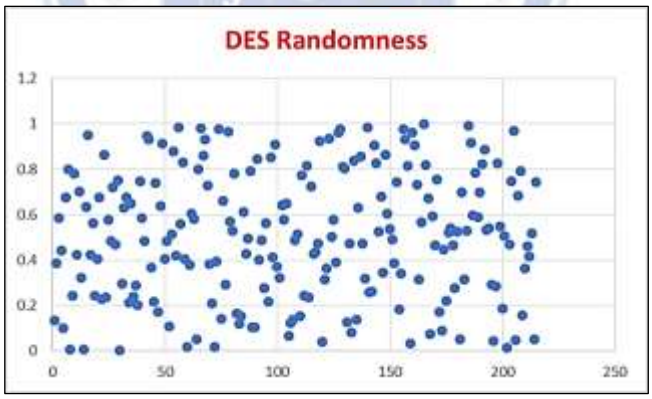
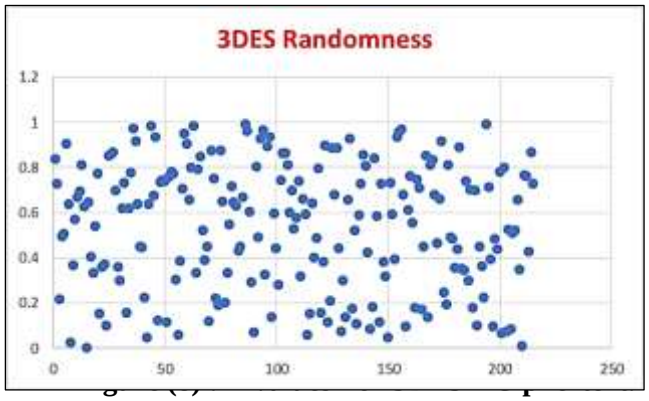


Figure (7) : P-values For DES – Ciphertext.

(3) For the 3DES algorithm with 112 bits key length, the randomness of the ciphertext as shown in figure (8).



(4) For the RC4 algorithm with 128 bits key length, the randomness of the ciphertext as shown in figure (9).

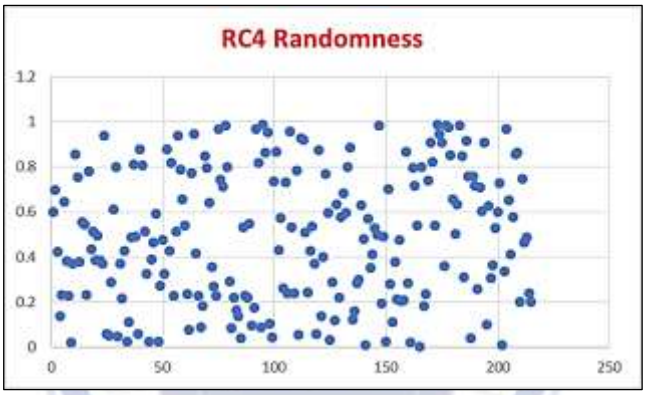


Figure (9) : P-values For RC4 – Ciphertext.

(5) For the Blowfish algorithm with 256 bits key length, the randomness of the ciphertext as shown in figure (10).

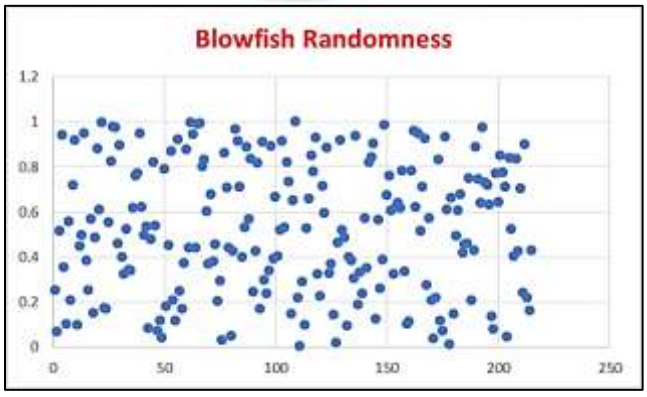


Figure (10) : P-values For Blowfish – Ciphertext.

(6) For the Twofish algorithm with 256 bits key length, the randomness of the ciphertext as shown in figure (11).

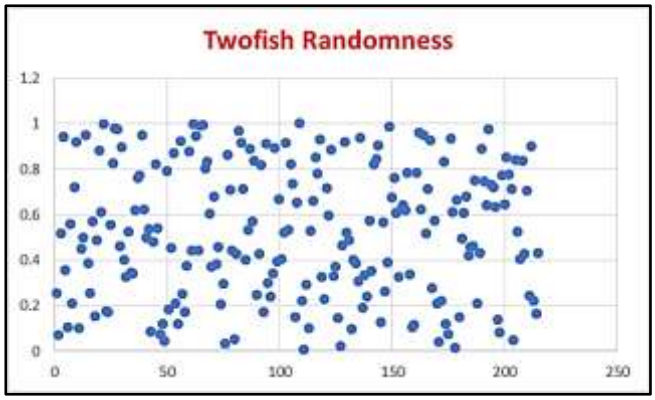


Figure (11) : P-values For Twofish – Ciphertext.

(7) As for the randomness of the original file, it is shown in figure (12).

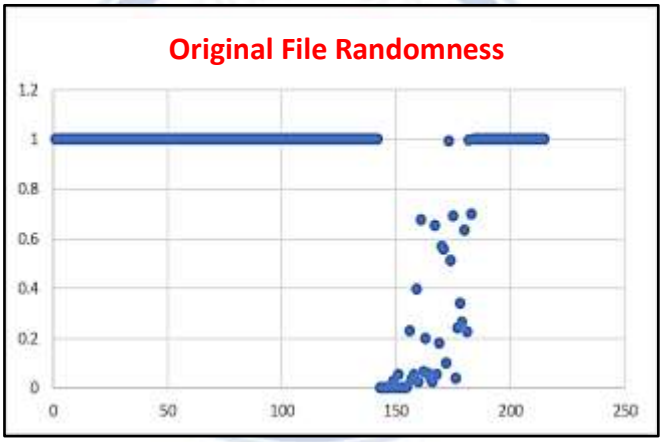


Figure (12) : P-values For the Original File.

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 (13).

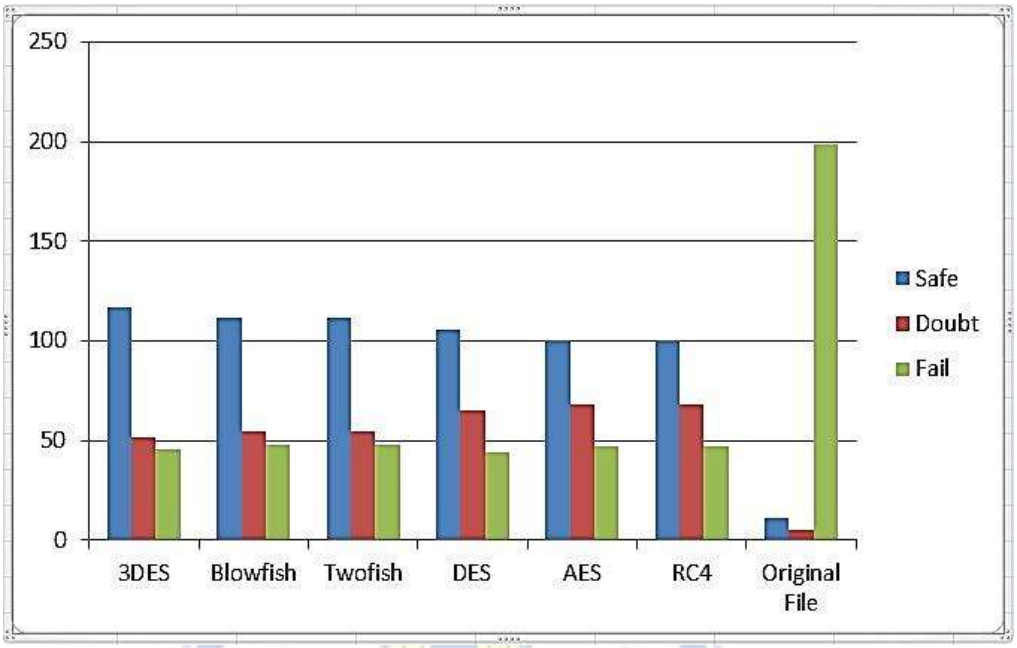


Figure (13) :Comparison between the six proposed algorithms in randomness criterion.

Conclusions

Through the comparison made in this paper between the proposed symmetric algorithms, the following was concluded: 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. 3DES is better than AES, DES, RC4, Blowfish, Twofish in terms of randomness. There is an inverse relationship between the performance and randomness for test algorithms. The more complex the algorithm, the more execution time it consumes for the data encryption and decryption.

Conflict of interests.

There are non-conflicts of interest.

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.

[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.

[8] H. Dibas and K. E. Sabri, "A comprehensive performance empirical study of the symmetric algorithms:AES, 3DES, Blowfish and Twofish," 2021 International Conference on Information Technology (ICIT), pp. 344-349, 2021, doi: 10.1109/ICIT52682.2021.9491644.

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

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

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

[12] 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.

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

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

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

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

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

الخلاصة

يعد الأمان في الوقت الحالي مهمًا للغاية وفعالًا للغاية لتطبيقات الإنترنت والشبكات ، والتي تنمو بسرعة ، وبالتالي زادت قيمة وأهمية البيانات التي يتم تبادلها عبر الإنترنت أو الوسائط الأخرى. لذلك ، فإن عملية البحث عن أفضل الحلول لغرض توفير الحماية المطلوبة ضد الهجمات غير القانونية مع توفير هذه الخدمات في الوقت المناسب هي واحدة من أكثر الموضوعات إثارة للاهتمام في المجتمعات ذات الصلة بالأمن. تهدف هذه الورقة إلى اختبار عدد من خوارزميات التشفير المتماثل (AES، DES، 3 DES، RC4 ، السمكة المنتفخة ، Twofish) باستخدام تسلسل كتل التشفير في وضع الأمان (CBC). ثم يتم إجراء مقارنة بينهما بناءً على معايير التقييم: يتم تنفيذ اختبارات وقت التشفير وفك التشفير باستخدام لغة برمجة Java. اختبار العشوائية على النص المشفر ، والذي تم تنفيذه باستخدام اختبار Diehard الإحصائي لحساب الخوارزمية الأكثر كفاءة لاستخدامها في تطبيقات الحياة المختلفة.

أظهرت نتائج البحث أن خوارزمية DES3 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 DES3 هو الأفضل مقارنة بباقي الخوارزميات ، بينما كان RC4 و AES الأسوأ في هذا المعيار.

الكلمات الدالة:

AES، DES، 3 DES، RC4 ، السمكة المنتفخة ، Twofish، CBC ، وقت التشفير ، وقت فك التشفير ، العشوائية