Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

A Systematic Method for Tackling Data Encryption Standard Vulnerabilities

Farah Al-Shareefi[1*]

[1] Babylon University,College of Science for Women, Computer Science Department, 51002, Iraq
wsci.farah.mohammed@uobabylon.edu.iq

[*] Corresponding author Email: wsci.farah.mohammed@uobabylon.edu.iq

https://doi.org/10.46649/fjiece.v4.1.22a.25.3.2025

Abstract. *Data Encryption Standard (DES) is a broadly used symmetric-key algorithm for ciphering digital data. However, there are several vulnerabilities related to this technique, including key distribution, namely, the key must be securely passed between the associated parties, and susceptibility, that is, the DES is defenceless against brute force and cryptanalysis attacks upon the key and the message, respectively. This paper attempts to tackle the first vulnerability by using the Diffie-Hellman protocol, as it enables the parties to interchange unshared secret. In addition, the second vulnerability is alleviated through: (1) performing the Columnar Transposition Cipher on the message before DES, (2) replacing the classical shifting process of the key with a type of shifter called Barrel. These improvements are vital for safeguarding against breaches and sophisticated attacks. The findings of conducted experiments have revealed that the proposed method satisfies the key and plaintext avalanche with a 50% output bit flipped in the ciphertext, and the average time required to crack passwords handled by the enhanced DES is more than that for classical DES. Therefore, the proposed method of this paper has raised the security requirements of DES.*

Keywords: *Data Encryption Standard (DES); Diffie-Hellman Protocol; Columnar Transposition Cipher; Barrel Shifter.*

1. INTRODUCTION

With the Internet continually evolving, the security concerns of the transmitted data over the network increases dramatically. Cryptography plays a key role at providing secrecy for data being sent over public network. Typically, Cryptography utilizes mathematical algorithms to implement encryption, decryption, or hashing processes on travelled data, also called messages [1], [2]. These processes sometimes require a data piece called key. Based on the number of the required keys, the cryptography is categorised into three types: **symmetric-key cryptography**, **public-key cryptography**, and **hashing function** [3], [4].

In symmetric-key style, only one key is employed for encryption and decryption, while two different keys are used with the public-key cryptography. Different from these cryptography types, no key is used with the hashing function. The symmetric-key cryptography is quite simpler than the public-key, and it is almost used when a large amount of data need to be commuted, but the public-key is the opposite of that. Hashing function produces a unique, fixed length and irreversible value for a data of a variable length [5], [6].

In addition, the cryptography algorithms are applied by protocols, called **cryptographic protocols** [7]. These protocols attain security service(s), such as: key distribution, authentication, secrecy, and so on. One of these protocols is **Diffie Hellman Key Exchange (DHKE)** protocol [8], [9], which is initially designed to distribute secret key between the sender and the receiver without a prior secret key interchanged between them. Utilizing this protocol for the symmetric-key cryptography needs to be paid a particular attention.

One of the most common used symmetric-key algorithm is Data Encryption Standard (DES) [10]. DES is known for its simplicity and quick processing. Despite of that, there is a great debate about several vulnerabilities, including but not limited to: key distribution and DES susceptibility. With respect to the first one, the DES entails that the key is inviolably distributed between parties and an implicit trust is confirmed in them. While regarding the last vulnerability, the DES is vulnerable to the brute force search and cryptanalysis attacks [11]. Accordingly, this paper aims at tackling the above mentioned problems by:

1. Employing the DHKE protocol to distribute the secret key, since this protocol allows to reliably exchange messages between parties without a need for prior keys sharing between them. Furthermore, the Argon2 hashing function is utilized to derive the DES symmetric key from the distributed one to the integrity of the key.

2. Increasing the complexity level of the DES key via replacing the shifting function with the Barrel Shifter (BS) one during the key generation operation. The reason for that, the BS function produces a less predictable value by the attackers comparing with that produced by the classical shifting function.

3. Raising the required efforts and time to break the DES algorithm through implementing further permutation on the plain message by using the Columnar Transposition Cipher.

The rest of this paper is structured as follows: Section 2 covers the main literature that technically relates to the proposed method of this paper. Section 3.1 surveys the background information about techniques and algorithms employed in this paper. Section 3.2 illustrates the elaboration of the proposed method, while Section 4 discusses the obtained results and findings. Finally, Section 5 derives a conclusion of this paper.

2. RELATED WORK

This section discusses related work that focuses on improving the security level of the DES algorithm. In [12], two techniques from image processing field have been utilized to improve the DES algorithm, such that it will become more resistant against cryptanalysis attacks. The first technique, which is the average pooling, has been used to generate 16 sub-keys after performing the shifting operation. While the second technique, which is filtering and striding, was employed in the f-function to increase both the avalanche results and the difference between the plaintext and the encrypted text. To the best of our knowledge, these techniques are irreversible, i.e., the decryption process cannot correctly be performed.

The authors in [13], have presented the enhanced multi states DES algorithm. In this algorithm, a key is composed of a combination of 16 states (halves) rather than the original 2 states (two halves). In addition, the output of each S-BOX operation is XORed with the next S-BOX output. The main reason for these enhancements is to make it more difficult to achieve brute force attack.

A technique that blends between DES and Diffie Hellman protocol was suggested [14]. This protocol was used as a tool for distributing a key for the DES method which relies on a shared secret key.

Therefore, the possibility of existing a deceptive participant will be decreased. This technique is somewhat similar to that one in this paper as they both utilize DHKE protocol, but this paper adds ensuring that the distributed key remains unaltered through Argon2 hashing function.

Some modifications were proposed in [15] to DES. Firstly, the expanded right key (48 bits) is divided into two equal parts each of 24 bits length. Secondly, two functions are implemented on both of these parts. Lastly, the length of the key is increased to 112 bits by employing two different keys. The analytical results have shown that these modification enhance the diffusion of DES but they require three halves the time of the original DES.

The concept of TKE (Two Key) was proposed in [16]. The essential advantage of this concept is that it makes the DES is less laborious with respect to the hardware. However, working with two keys requires more time for achieving the encryption and hardware is not the focus of this paper.

Different from the previous literature, this paper tries to enhance the security of DES algorithm by focusing on both the key distribution and the susceptibility challenges. For the key distribution challenge, this paper attempts to distribute it safely using DHKE protocol and Argon2 technique, and for the susceptibility, the key generation way is complicated via BS function, and an extra layer of encryption for the message is added.

3. MATERIALS AND METHODS

3.1. MATERIALS

This section surveys some background information on the cryptographic primitives and cryptographic protocols.

- **Data Encryption Standard**

Data Encryption Standard (DES) is one of the widespread symmetric-key encryption paradigm that was originally developed by National Bureau of Standards (NBS) in 1972, as a tool for protecting sensitive and uncategorized electronic government data. Following that, exactly in 1975, DES has been approved as an official encryption algorithm [10].

DES requires two inputs: 64-bits plain text message and 56-bit key, in order to yield 64-bit encrypted text [17]. Its principle work depends on two characteristics of cryptography: substitution (also called confusion) and transposition (also known as diffusion). The encryption process of the DES can be described in the following steps, which are schematically illustrated in Figure 1.

Step1: The 64-bit plain text is subjected to the initial permutation in order to obtain two 32-bit permuted block, which are termed Left Plain Text (LPT) and Right Plain Text (RPT).

Step 2: Each permuted block passes through16 rounds to achieve the encryption process, such that there is a specific key (48-bit) for each round. The sixteen rounds can be summarized through the following:

Step 2.1: The 64-bit key undergoes its key permutation to yield 56-bit reordered key.

Step 2.2: The obtained 56-bit key is partitioned into two halves. Next, the left shifting operation is conducted on every half part of the key, such that it is shifted to the left direction one or more positions relying on the round number. Following that, the two shifted halves are combined and subjected to the second permutation to gain 48-bit key.

Step 2.3: The 32-RPT is expanded to 48 bits using an expansion permutation operation.

Step 2.4: The output of step 2.3 is XORed with the 48-bit key (obtained from step 2.2).

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

Step 2.5: The result of step 2.4 is fed to eight S-boxes to produce 32 substituted bits, which are subsequently XORed with the LPT. The xoring result becomes the new RPT, while the new LPT equals to the old RPT. The new LPT and RPT are given to the next round and so on.

Step3: After completing the sixteen rounds, the final permutation, which represents the inverse of the initial permutation, is performed on the LPT and the RPT that are yielded from the last round to produce the 64-bit encrypted text.

The above steps are used to decrypt the cipher text as well, but in a reverse order.

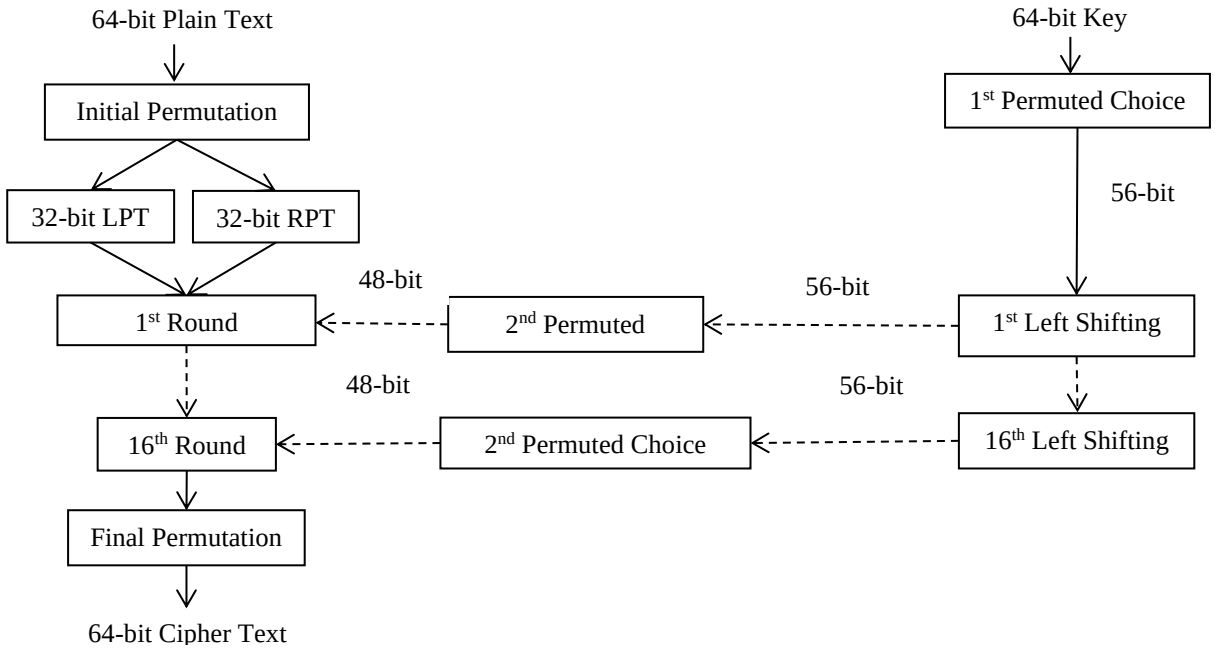


Fig. 1. The Traditional DES Encryption Process

- **Columnar Transposition Cipher**

The Columnar Transposition Cipher (CTC), as its name suggests, an algorithm for arranging the letters of the plaintext in rows of a rectangle, and then reading them off in columns according to an identified order [18], [19].

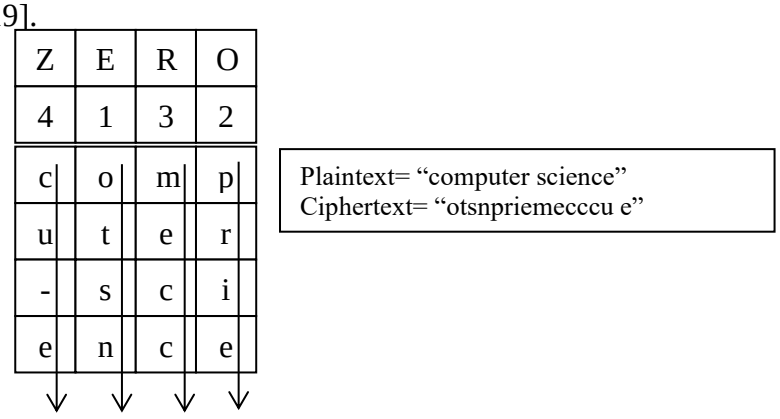


Fig. 2. An Illustrated Example of the CTC Encryption Process

The encryption algorithm for this cipher can be summarized into the following steps:

Step 1: The number of rows and columns is determined, such that:

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

$$no.of\ columns = length\ of\ the\ key \quad \ldots\ldots\ldots\ldots(1)$$
$$no.of\ rows = \frac{length\ of\ the\ message}{no.of\ columns} \quad \ldots\ldots\ldots\ldots(2)$$

Step 2: The permutation of the columns is identified according to the alphabetic order of the key's letters. For example, if we have a key that is equal to "ZERO", then the permutation should be "4 1 3 2".

Step 3: The letters of the message are arranged in rows, and then read out column by column according to the identified permutation order, see Figure 2 that illustrates this operation for the message="computer science".

The decryption process is similar to the encryption one, the letters should be read row by row.

- **Barrel Shifter**

Barrel Shifter (BS) is a cryptographic technique that was designed as a logical hardware circuit for shifting and rotating multiple bits in only single cycle [20], [21]. It allows data to be only shifted to left or to right direction, i.e., it is a bidirectional process. The main inputs for this shifter is the number of bits which must be shifted, the shifting direction and position. This shifter is widely used as it is very flexible and efficient and has a moderate level of security and speed.

- **Argon2**

Argon2 is a member of password hashing (PH) schemes that was announced winner of the PH Competition [22], [23]. It is designed for the x86 architecture, configurable in both time and memory amount, underpins variable number of used threads. It well-suits applications which require high resistance against brute force attacks. The length of produced hashing is customizable ranging from 32 bytes (256 bits) to 64 bytes (512 bits).

- **Diffie Hellman Key Exchange Protocol**

The Diffie Hellman Key Exchange (DHKE) protocol is initially designed to securely produce a private key over an unreliable communication network, without the requirement of broadcasting any secret key [8], [9]. In DHKE, the sender A, and the receiver B, concur on two public keys P and Q in advance. Next, each party selects and possesses its own private key (without participation), for example a for A and b for B. Following that, two messages are exchanged. Firstly, A calculates the public value $S = P^a\ mod\ Q$ and sends it to B. Secondly, B calculates another public value $T = P^b\ mod\ Q$ and sends it to A. Finally, A and B generate the new session secret key via the following calculations: $k_A = T^a mod\ Q$ and $k_B = S^b mod\ Q$. Note that, $k_A = k_B$. The protocol is as follows:

1. $A \rightarrow B: S = P^a\ mod\ Q$
2. $B \rightarrow A: T = P^b\ mod\ Q$

3.2. SPECIFICATION OF THE PROPOSED METHOD

In this section, we present a method for developing a more secure specification of the DES algorithm. Typically, it extends the classical DES via: Firstly, adopting authentication protocol for distributing the DES key in a safe way, and adding further permutation to the message by implementing the CTC algorithm. Secondly, replacing the shifting operation on the key value with the BS one. Figure 3 schematically shows the whole method, which is elucidated below.

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

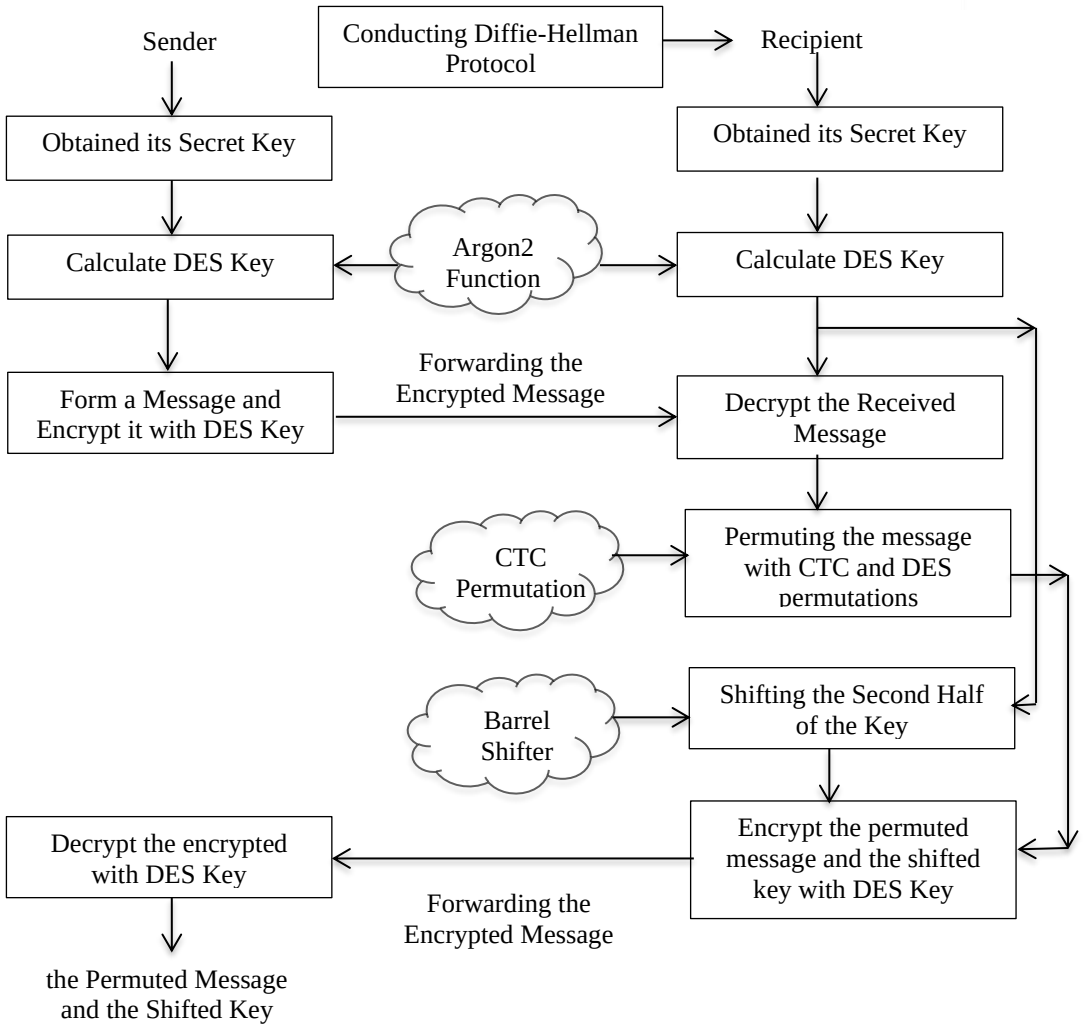


Fig. 3. A Schematic Representation of the Proposed Method

The proposed method incorporates two stages:

- *THE KEY DISTRIBUTION STAGE*

This stage is devoted to the accomplishment of the DHKE protocol in order to exchange the key and the message of the DES in a secure manner. Figure 4 shows an overview of this stage which is made up of the following:

First: in this step, the sender with an identity A and the receiver with an identity B publicly agree on two arbitrary values: P and Q. Next, two secret keys : a and b are chosen by A and B, respectively. Later, A calculates its public key using the following formula: $S = P^a \ mod \ Q$ and sends the result to B.

Second: B receives A's public key and saves it. After that, B computes it's public key: $T = P^b \ mod \ Q$, and forwards it to A.

Third: A uses the received public key to mathematically create a private key: $k_A = T^a mod \ Q$. On the other hand, B converts the received public key into a secret one via the following formula: $k_B = S^b mod \ Q$.

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

Fourth: Both parties A and B derive the key of DES method through hashing the calculated private key with Argon2 technique. The produced Des key is 56 bits (7 bytes).
Fifth: A determines the DES message and encrypts it with DES key, and forwards the result to B.
Sixth: If B can successfully decrypt the received message then A is an authenticated party and the DES key is valid.

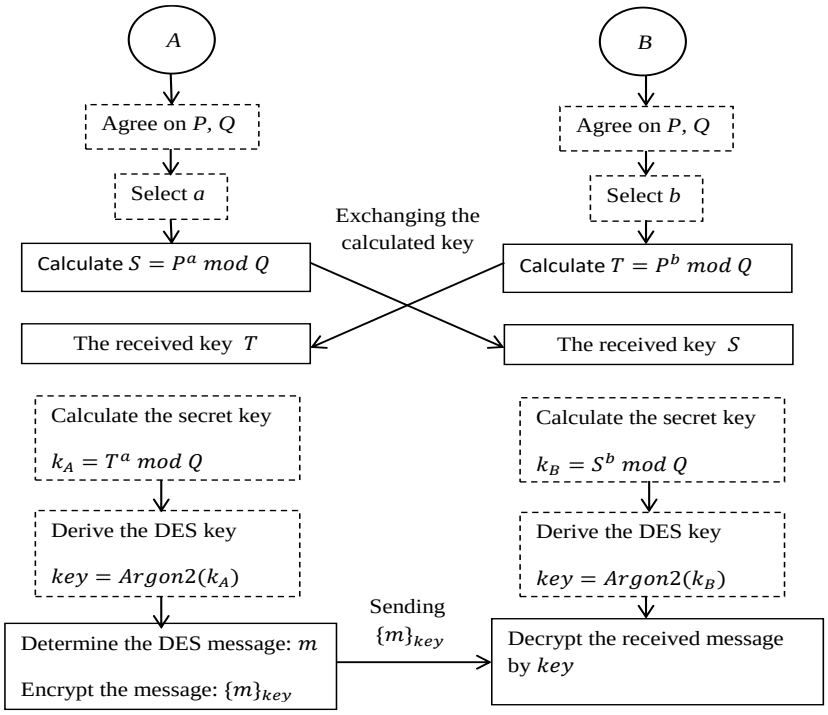


Fig. 4. The Key Distribution Stage

- *THE ENCRYPTION STAGE*

This stage is mainly aimed at strengthening the encryption process of the DES algorithm. Figure 5 diagrammatically illustrates the main steps of this stage, which can be summarized as:

(1) The agreed hexadecimal message (16 letters) is initially permuted using the CTC algorithm.
(2) The produced permuted message is converted into binary format to gain a 64-bit message length.
(3) The obtained message from step 2 is further permuted using the same permuted table dedicated to the classical DES algorithm.
(4) The second permuted output is partitioned into two halves (left and right), 32-bit each.
(5) The two halves and the key are handled in 16 rounds, via carrying out the following:

(5.1) The binary conversion is performed on the agreed key: DES key, and the output is subjected to the first permuted choice that produce 56-bit key.
(5.2) The 56-bit key is circular shifted 16 times to produce 16 different key by using the BS, such that both of the shifting direction and the number of shifted bits are agreed by the parties in advance.
(5.3) The 16 keys are undergone to a compressed permutation, or a second choice permutation to yield 16 keys of 48-bits length.

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

(5.4) The two halves are processed in a way similar to that one in the traditional DES algorithm.

(6) After completing the 16 rounds, the resultant text is subjected to the inverse of the initial permutation to output the encrypted text.

Note that the decryption processing is performed in a reverse order used in the encryption stage.

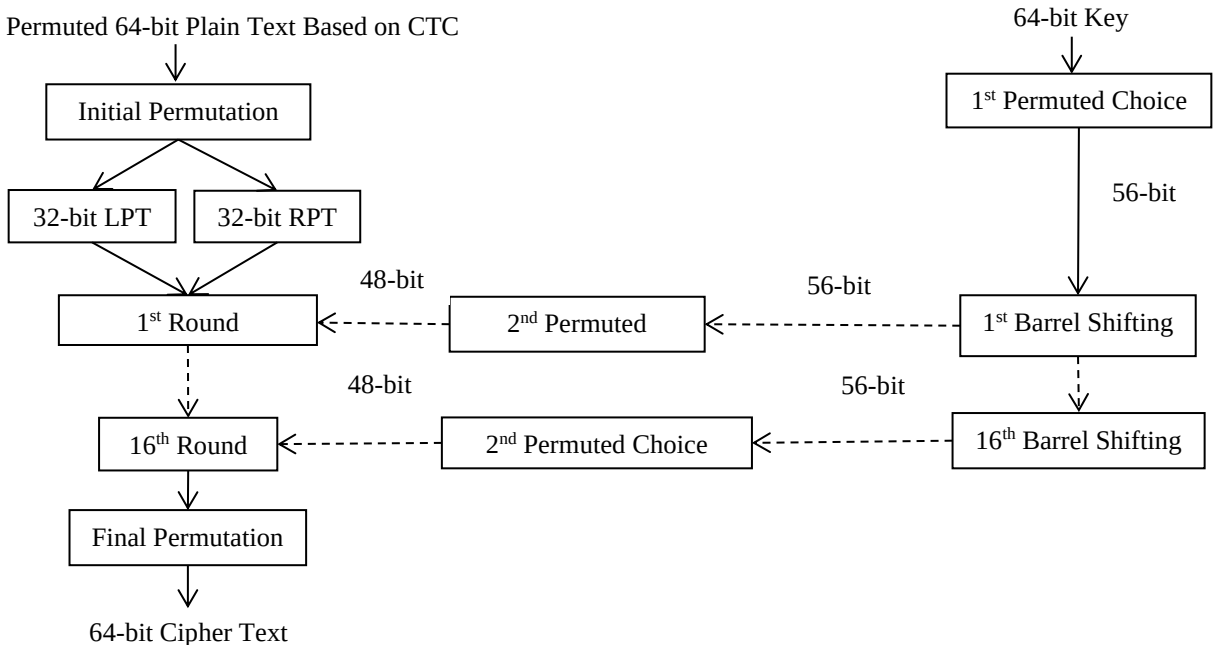


Fig. 5. The Modified DES Encryption Process

4. RESULTS AND DISCUSSION

In order to conduct the experiments of this work, Python Language is used for simulation. The employed computer in the simulation was Intel® Core (TM) i5- 6300U CPU @ 2.40GHz 2.50 GHz with RAM of 8 GB. The performance of the proposed method is evaluated on the basis of parameters such as launching Brute Force attack upon passwords, plaintext and key avalanche. The time has not been measured as the BS operation is more elaborated than the traditional shifting.

- **Brute Force Attack**

For the purpose of analyzing brute force attack upon passwords processed by DES and DES stemming from BS, two different runs of performance tests were conducted. The first run examines the impact of variable password lengths (1-7 letters) on searching time required by passwords handled by DES technique. The next run investigates the effect of the same password lengths used in the previous run on the time needed by passwords handled by DES based on BS. Table 1 tabulates the results of both runs in milliseconds. It can be deduced from this table that the as the length of the password increases, the required time by the attacker to crack it increases as well. In addition, the time for cracking passwords handled by DES based on BS is more than that for traditional DES passwords. This is due to the more probable permutations that a longer password has which make it difficult for cyberattacker to crack.

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

Table 1. Average Time for Brute Force Attacker Against Passwords Used by DES and DES Based on BS

Password Length	Average Time for Cracking Passwords Using DES (Milliseconds)	Average Time for Cracking Passwords Using DES Based on BS (Milliseconds)
1 Digit	0.0	0.0
2 Digits	0.0	0.0
3 Letters	3.66	3.66
4 Letters	53.20	54.19
5 Letters	724.41	785.91
6 Letters	1892.94	36987.43
7 Letters	32822.42	117079.85

- **Avalanche Effect**

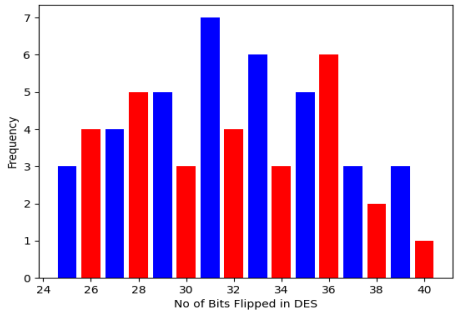
The avalanche effect is used as an indicator of how many bits in the cipher text will be flipped when only single bit is modified in the unencrypted text (or in the key) [24], [25]. It is expressed through the following formula:

$$Avalanche\ Effect = \frac{Hamming\ Distance\ (no.of\ modified\ bits\ in\ encrypted\ text)}{length(encrypted\ text)} \quad \ldots\ldots\ldots(3)$$

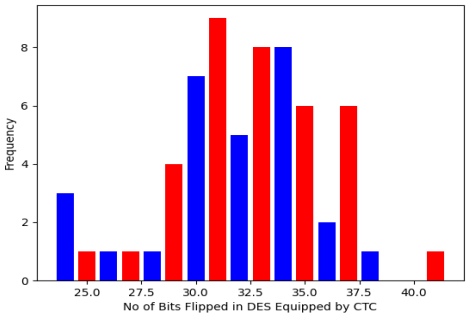
This formula is employed to assess experiments in both the plaintext and the key avalanche effects. These experiments is repeated 64 times as the total length of the DES cipher text is 64 bits. The **plaintext avalanche effect** is firstly carried out. In this experiment, the key remains unchanged and it is equal to "ABCDEF123456789". While the plaintext is initially equal to "computer". Then, a list of 63 different plaintexts is generated from the initial one by altering only one bit. The reason for using variable plaintexts is to inspect how the CTC plaintext permutation will affect the DES encryption result. Figure 6 depicts the number of the modified bits in the x-axis versus frequency of bits that have been changed in the y-axis for subfigures (a), (b), and (c). Figure 6 (a) relates to the classical DES which achieves 40 for max hamming distance, 25 for min hamming distance, and 0.49 for mean plaintext avalanche effect. While figure 6 (b) pertains to the DES that is equipped with permuted plaintext by CTC that gives 41 for max hamming distance, 24 for min hamming distance, and 0.50 for mean avalanche effect. Figure 6 (c) delineates that the DES based on CTC permutation has higher plaintext avalanche effect comparing with the classical DES. As a results, the CTC permutation assists at increasing the plaintext avalanche effect.

With respect to the avalanche effect criterion, the key avalanche effect is also implemented as the key processing in the DES techniques is proceeded with BS operations. Figure 7 shows the histogram representation of Hamming distance for subfigures (a), (b), and (c). Figure 7 (a) is dedicated to the classical DES which achieves 76 for maximum hamming distance, 53 for minimum hamming distance, and 63 for the average of hamming distances. While figure 7 (b) pertains to the DES that is based on the BS operations that gives 79 for maximum hamming distance, 51 for minimum hamming distance, and 64 for average distances. Figure 7 (c) sketches that the DES grounded by BS operation has a slightly higher key avalanche effect comparing with the classical DES. However, if the least significant bit is altered then the produced cipher text remains the same in both techniques. In general, the DES that is grounded by BS operation satisfies the desired key avalanche effect.

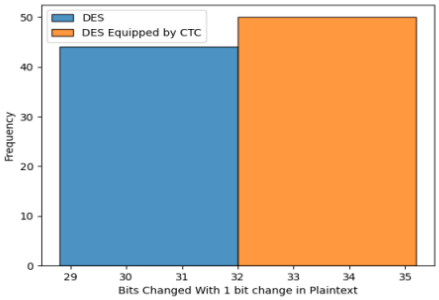
Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985



(a) Plaintext Avalanche Effect of DES

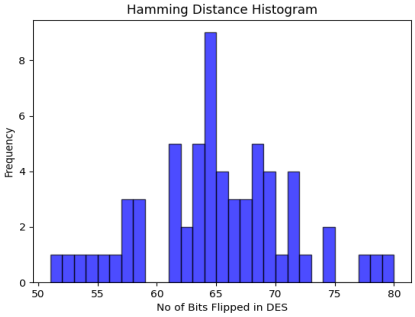


(b) Plaintext Avalanche Effect of DES Equipped by CTC

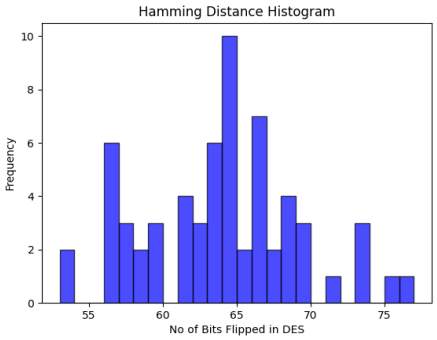


(c) A Plaintext Avalanche Effect Comparison Between DES and DES Equipped by CTC

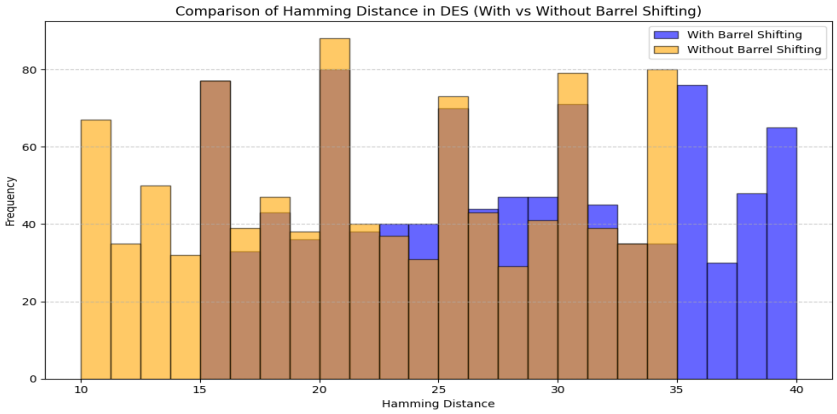
Fig. 6. Distribution of Plaintext Avalanche Effect



(a) Key Avalanche Effect of DES



(b) Key Avalanche Effect of DES Supplied by BF



(c) A Key Avalanche Effect Comparison Between DES and DES Supplied by BS

Fig. 7. Distribution of Key Avalanche Effect

- **Comparison**

To establish the efficiency of the proposed method, a quantitative comparison of the cryptographic key security is performed between this method and that of [14]. In essence, Shannon Entropy [26] is used to measure the unpredictability of a cryptographic key. A higher entropy value refers to more possible key combinations that the attacker should try. Consequently, its task becomes more difficult. It can be computed via the following equation:

$$H = log_2(N) \quad \text{.....(4)}$$

where N is the potential number of key combinations.

As the method of this paper uses Argon2 to derive 56 bit key, then its entropy is $H = log_2(2^{56})$= 56 bits. This means that the attacker has to brute force all 2^{56} combinations. While the method of [14] utilizes DHKE protocol at a certain round of DES method, unlike the method of this paper that initially exchanges key and then derives it via Argon2, As a result, if the DHKE of this method uses 512bit prime number then the entropy is roughly $H = log_2(2^{40})$= 40 bits only, because of the problem of calculating the discrete logarithm. This indicates that the proposed method is less vulnerable to brute force attack comparing with that of [14].

5. CONCLUSIONS

This paper presents a method for adding extra security level to the DES algorithm based on the Diffie-Hellman protocol, CTC technique, and the BS function as well. The Diffie-Hellman can be used to securely distribute a key between participants and demonstrate the authentication of these participants as well. The CTC helps to increase the avalanche effect and the dissimilarity between the original plain text and the obtained encrypted text. The permutation feature of the BS function permits gaining complicate and unique keys, which positively affect the security of the DES. As a future step, changing the BS operation with another more secured permutation.

REFERENCES

[1] Banoth, R., and Regar, R., *Classical and Modern Cryptography for Beginners*, 2023rd ed., Springer International Publishing, Cham, Switzerland, 2024.
[2] Easttom, W., *Modern Cryptography: Applied Mathematics for Encryption and Information Security*, Springer Nature, 2022.
[3] Dooley, J. F., *History of Cryptography and Cryptanalysis*, Springer, 2018.
[4] Sharma, A. K., and Mittal, S. K., "Cryptography & network security hash function applications, attacks and advances: A review," *Proceedings of the 2019 Third International Conference on Inventive Systems and Control (ICISC)*, IEEE, pp. 177–188.2019.
[5] Padhye, S., Sahu, R. A., and Saraswat, V., *Introduction to Cryptography*, CRC Press, London, England, 2021.
[6] Swathi, E., Vivek, G., and Rani, G. S., "Role of hash function in cryptography," *Proceedings of NCCSIGMA-16*, 2016.
[7] Goldreich, O., "Cryptography and cryptographic protocols," *Distributed Computing*, 16 (vol.), 177 (first page), 2003.
[8] Hellman, M. E., "An overview of public key cryptography," *IEEE Communications Magazine*, 40 (vol.), 42 (first page), 2002.

[9] Dritsas, E., Trigka, M., and Mylonas, P., "Performance and security analysis of the Diffie-Hellman key exchange protocol," *Proceedings of the 2024 19th International Workshop on Semantic and Social Media Adaptation & Personalization (SMAP)*, IEEE, pp. 166–171.2024.
[10] Tuchman, W., "A brief history of the data encryption standard," *Internet Besieged: Countering Cyberspace Scofflaws*, 275 (first page), 1997.
[11] Reyad, O., Mansour, H. M., Heshmat, M., and Zanaty, E. A., "Key-based enhancement of data encryption standard for text security," *Proceedings of the 2021 National Computing Colleges Conference (NCCC)*, IEEE, pp. 1–6, 2021.
[12] Amorado, R. V., Sison, A. M., and Medina, R. P., "Enhanced data encryption standard (DES) algorithm based on filtering and striding techniques," *Proceedings of the 2019 2nd International Conference on Information Science and Systems*, pp. 252–256, 2019.
[13] Patel, P., Shah, K., and Shah, K., "Enhancement of DES algorithm with multi-state logic," *International Journal of Research in Computer Science*, 4 (vol.), 13 (first page), 2014.
[14] Habeeb Jafar, S., "Proposal to Complex DES Security Using Diffie-Hellman Injection," *Engineering and Technology Journal*, 29 (vol.), 1216 (first page), 2011.
[15] Verma, J., and Prasad, S., "Security enhancement in data encryption standard," *Proceedings of Information Systems, Technology and Management*, Springer, Berlin, Heidelberg, pp. 325–334.2009.
[16] Fan, J., and Zhu, X., "Data encryption by two keys," *Proceedings of the 2009 First International Conference on Information Science and Engineering*, IEEE, pp. 1683–1686. 2009.
[17] Sharma, M., and Garg, R. B., "DES: The oldest symmetric block key encryption algorithm," *Proceedings of the 2016 International Conference on System Modeling & Advancement in Research Trends (SMART)*, IEEE, pp. 53–58...2016.
[18] Ritu, ., Niram, ., Narwal, E., and Gill, S., "A novel cipher technique using substitution and transposition methods," *Proceedings of Rising Threats in Expert Applications and Solutions: FICR-TEAS 2022*, Springer, Singapore, pp. 123–129. 2022.
[19] Dagadu, J., Armah, A., Aboagye, E. O., and Mansuru, S. A., "Rubik's cube enhanced columnar transposition cipher," *Journal of Computer Science and Applications*, 12 (vol.), 31 (first page), 2024.
[20] Hashmi, I., and Babu, H. M. H., "An efficient design of a reversible barrel shifter," *Proceedings of the 2010 23rd International Conference on VLSI Design*, IEEE, pp. 93–98. 2010.
[21] Srivastava, P., "Case studies: Barrel shifter and binary adders," *Completion Detection in Asynchronous Circuits: Toward Solution of Clock-Related Design Challenges*, Springer, Cham, pp. 45–58. 2022.
[22] Biryukov, A., Dinu, D., and Khovratovich, D., "Argon2: New generation of memory-hard functions for password hashing and other applications," *Proceedings of the 2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, IEEE, pp. 292–302. 2016
[23] Sharma, A., Thapliyal, S., Wazid, M., Mishra, A. K., Kumar, P., and Giri, D., "A secure mechanism for password hash value generator with the security analysis of various hashing algorithms," *Proceedings of the 2024 4th International Conference on Computer, Communication, Control & Information Technology (C3IT)*, IEEE, pp. 1–6. 2024.
[24] Tiwari, N., and Kumar, A., "Security effect on AES in terms of avalanche effect by using alternate S-box," *Proceedings of the International Conference on Intelligent Data Communication Technologies and Internet of Things (ICICI) 2018*, Springer, Cham, pp. 1–14.2019.
[25] Upadhyay, D., Gaikwad, N., Zaman, M., and Sampalli, S., "Investigating the avalanche effect of various cryptographically secure hash functions and hash-based applications," *IEEE Access*, vol. 10, pp. 112472–112486, 2022.
[26] Reeds, J. , Entropy calculations and particular methods of cryptanalysis. *Cryptologia*, *1*(3), 235-254, 1977.

