Journal of King Saud University – Computer and Information Sciences 35 (2023) 101716

HOSTED BY

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

A new modified DES algorithm based on the development of binary encryption functions

Hasan Kadhim A. Alsuwaiedi [a,*], Abdul Monem S. Rahma [b]

[a] Iraqi Commission for Computers & Informatics, Informatics Institute for Postgraduate Studies, Baghdad, Iraq
[b] Computer Science Department, Al-Maarif University College, Anbar, Iraq

ARTICLE INFO

Article history:
Received 21 May 2023
Revised 14 July 2023
Accepted 16 August 2023
Available online 22 August 2023

Keywords:
Feistel structure
Information security encryption
Des
Multi-level keys
Symmetric block cipher

ABSTRACT

One of the most commonly used symmetric encryption methods is the Data Encryption Standard (DES), but it has a number of flaws. For instance, the encryption method only employs two functions (XOR), acts on one bit, and only allows for a finite number of combinations of data. To solve these issues, a new modified DES algorithm is presented in this study. The typical encryption DES algorithm of 16 rounds has been extended in the key space and the plaintext (using multiple keys) by presenting a new operation (**Xo**) replace the present XOR operation to increase complexity. In contrast to DES, which only employs one key, the research approach uses three keys other than the standard DES one. The input key for encryption and decryption processes is the first key. while other three keys are used to determine the table numbers (2, 4, 6, 8, 10, and 12), which have values ranging from 0 to 4096. When compared to the famous DES and other modified algorithms, the complexity analysis of the suggested method reveals it is the most difficult. As a result, the attacker must make at least $(4.75947385446610445448940208947 48E + 173)$ attempts to break the key and get the clear message. This indicates that the well-known DES security level will be raised by the suggested DES algorithm, depending on the new **Xo** function.

1. Introduction

The ongoing advancements in digital information technology made by the evolution of digital and electronic contents as well as the growth of communication networks have made the digital world more complex and made it necessary for the transmission of sensitive and important information to be kept secure, allowing only authorized individuals to have access to it (Qusay Zuhair Abdulla, 2023). Text is one of the basics of information, along with other types such as images, audio, video, etc. This research focuses on text encryption but can be developed for other types. The DES algorithm is the topic of this research, which is one of the techniques that was approved for its efficiency from 1975 until 1995, but it was replaced by other technologies due to penetration and breaking (Kumar, 2020). Improving the DES algorithm and raising its complexity will lead to obtaining a new modified DES algorithm based on new technology.

There are two varieties of encryption techniques: symmetric key encryption, in which the same key is used in the encryption and decryption processes, and asymmetric key encryption, in which different keys are used in the encryption and decryption processes. Block ciphers and stream ciphers are the two forms of symmetric algorithms. In stream ciphers, each bit is independently encrypted by stream ciphers by appending a bit from a key stream to a bit of plaintext. Block ciphers use the same key to encrypt a whole block of plaintext bits at once. Block ciphers may be based on a non-Feistel network, for instance, the advance encryption standard (AES), or a Feistel network, such as the data encryption standard (DES) (Vincenzo Agate, 2023). Horst Feistel created the Feistel network, which has been included in several block cipher schemes. It may be characterized as a typical technique for turning a function typically referred to as an **F** function into a permutation. Feistel networks' main task is to double the rounds needed to repeat the identical operations, for instance, bit-swapping (using P-boxes for permutations), straightforward nonlinear functions (S-boxes for substitution), and a logical operation that is specified as XOR using two states 0 or 1 (Vincenzo Agate, 2023) (Dai, 2020).

The Feistel structure used in the DES algorithm is the topic of this research. IBM developed this technique for symmetric encryption in

* Corresponding author.
E-mail addresses: phd202030563@iips.icci.edu.iq (H.K.A. Alsuwaiedi), monem.rahma@uoa.edu.iq (A.M.S. Rahma).
Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2023.101716
1319-1578/

1972, and the National Bureau of Standards adopted it as the Federal Information Processing Standard in 1977 (Yongzhen, 2021). Ideally, the DES algorithm is suited for hardware execution. In contrast, it tends to exhibit sluggish software implementation. The technique supports 64-bit keys, while it originally only accepted 56-bit keys since the other eight bits are utilized to identify errors (Chen, 2021). For encryption and decryption, most current cryptography procedures operate with two states (0, 1). One of the block algorithms, DES, employs the traditional logical operation XOR, which is dependent on two states: just (0, 1), which has various flaws that lead to broken DES; for instance, it can be simple to comprehend and hence vulnerable to attacks (Chen, 2021) (Hassan, 2010). As a result, in the following sections, researchers propose to replace the two states with sex (2, 4, 6, 8, 10, 12) to increase key space. concentrating on the shortcomings of XOR in this study by changing it with the novel **Xo** operation with configurable block-bit sizes (n): (2, 4, 6, 8, 10, 12) instead of a single block size. The new **Xo** operation works on a wide and multi-size data volume, which increases efficiency depending on the addition in GF(2^n). Each block will create various state tables. Two additional keys are used to oversee the whole new **Xo** procedure. This process is done within every DES round or phase to raise the algorithm's security. The findings suggest that this modified DES algorithm will raise the degree to which cryptography is secured by maximizing the complexity in each phase or round, ensuring the safety of information.

2. Related work

For modern encryption and decryption, most cryptographic algorithms rely on functions having two states (0, 1) (Stallings, 2023). As a result, most algorithms employ the traditional logical operation (XOR), which is based on two states: simply (0, 1) and has various drawbacks, including being simple since attackers can readily understand it. Therefore, researchers sought to replace the two states with four (0, 1, 2, 3). This section provides a summary of the associated literature on possible improvements to key distribution using state tables in the DES algorithm.

- **In 2010** (Hassan, 2010)**;** the researcher revealed a novel way for improving the efficiency of the DES-algorithm by altering the predetermined XOR operation used during the sixteen rounds of the traditional Feistel algorithm. The (#) action is mostly dependent on the use of two keys. Each key is made up of a combination of four states (0, 1, 2, and 3) rather than the traditional two states (0 and 1). This update increases the protective strength and resilience of breaking techniques against attackers.
- **In 2019** (Rahma, 2019), this paper offered a new approach for the Modified DES (MODDES) algorithm, which replaces the XOR function with a new operation based on multiplication in a Galois-Field GF (2^n) based on irreducible polynomials. It got through each round by utilizing four keys. The primary key is used to produce two keys, and the other two keys are generated internally. The new method works on a bit rather than a byte. The suggested approach has been evaluated and found to be more secure and resistant to assault.
- **In 2020** (Kareem, 2020)**;** this work presents a new modification referred to as "extended function" (E#) by altering the classic XOR operation used during the sixteen rounds of the Feistel structure in the DES algorithm with the E#, which is based on implementing multi-state tables created by using addition in GF(2^n) with n-dynamic sections of (1, 2, 4, 8 blocks), used in the DES algorithm as a symmetric cryptographic method. By enhancing its complexity and unpredictability, it offers a new level of security against breaking approaches.

3. Theoretical background of binary functions and Feistel cipher

Modern encryption is built on the finite field GF(2^n) (Kareem, 2020) and the binary function XOR used in encryption and decryption, such as the Vernam Cipher in Fig. 1, to produce the cipher text by mixing the key stream (***Ki***) with the plaintext (***Pi***), producing a ciphertext (***Ci***), and vice versa. Inside the finite field, the arithmetic operations are performed on polynomials (addition and multiplication) using the ordinary rules of algebra based on modular polynomial arithmetic. By far, the most common algebraic structure utilized in the development of cryptographic methods is finite fields (Stallings, 2023). Based on this foundation, several new cryptographic systems have been developed, as in (Bahjat, 2009).

Using state tables based on AND and XOR functions in encryption and decryption will add more security to the system due to the high randomness of the produced key or plaintext. Algorithms for cryptography depend on functions with four states (0, 1, 2, and 3), as in Fig. 2 (Wiemers, 2021).

The (#) function, which uses dynamic blocks (1, 2, 4, 8-blocks) and addition in GF(2^n) to create multi-state tables, is added to the encryption algorithms for strengthening security, increasing the complexity of the ciphertext or key, and relatively minimizing the encryption time (Chen, 2021) (Haneen Alabdulrazzaq, 2022).

A structure known as a Feistel block cipher underpins symmetric block encryption methods. A block cipher transforms an n-bit block of clear text into an n-bit block of encrypted text. For the encryption to be reversible (i.e., for decryption to be feasible), each of the 2^n alternative plaintext blocks that are possible must result in a different ciphertext block. A change like that is referred to as "reversible". Fig. 3 illustrates reversible and irreversible transformations for $n = 2$ (Bahjat, 2009) (velampalli, 2018).

In general, one round of DES, as in Fig. 4, The Feistel structure contained two steps:

- *First step*: It is based on mixing the **key (kn)** with the plaintext or ciphertext in **F**.
- *Second step*: the result of **F**, which used to be XORed with the continued encryption and decryption processes to perform the second function (Stallings, 2023)**.**

4. The proposed development

For many uses, the DES is not considered safe for a number of reasons. Mostly, it depends on just one bit (0 or 1). In the same way, it only uses one function (XOR), which means it is not random enough and can be attacked. So, in this portion of the article, it's a novel way to change the DES in order to fix these problems and make the encryption work better and harder to break.

This can be done by removing the binary function XOR and using a new **Xo** function for each phase or round of the standard DES algorithm. As described in Fig. 5, the first-key **k1** is the standard DES input key that is employed in the encryption and decryption processes. The second keys **k2** and **k2`** are dynamic bit-block sizes extracted from plaintext (**Ri**) and **ki** consecutively. The keys **k2** and **k2`** are considered preprocessing of the function **Xo**.

Inside the function **Xo,** the key **k2** is segmented in size into three parts from plaintext (**Ri**) as (**k3**, **k4**, **k5**). Also, the key **k2`** is segmented into three parts: **k3`**, **k4`**, and **k5`**. The whole parts' sizes in the function Xo are described as follows:

- The first segment represents the **first main state table**, denoted as keys **k3** and **k3`**.
- The second segment represents the **second main state table**, denoted as keys **k4** and **k4`**.

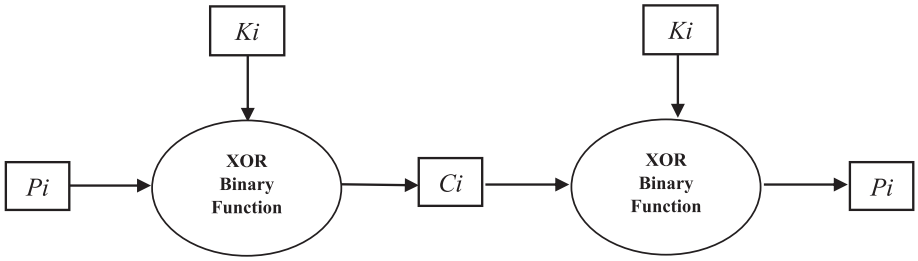

Fig. 1. Binary Function XOR in Vernam Cipher (Stallings, 2023).

#0	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

#1	0	1	2	3
1	1	0	3	2
0	0	1	2	3
3	3	2	1	0
2	2	3	0	1

#2	0	1	2	3
3	3	2	1	0
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1

#3	0	1	2	3
0	0	1	2	3
3	3	2	1	0
2	2	3	0	1
1	1	0	3	2

Fig. 2. Four state tables for (#) operator (Hassan, 2010) (Bahjat, 2009).

Mapping with a Reversal	
Plaintext	**Ciphertext**
00	11
01	10
10	00
11	01

Mapping with an Irreversible	
Plaintext	**Ciphertext**
00	11
01	10
10	01
11	11

Fig. 3. Reversible and Irreversible transformations for n = 2 in Feistel (Stallings, 2023).

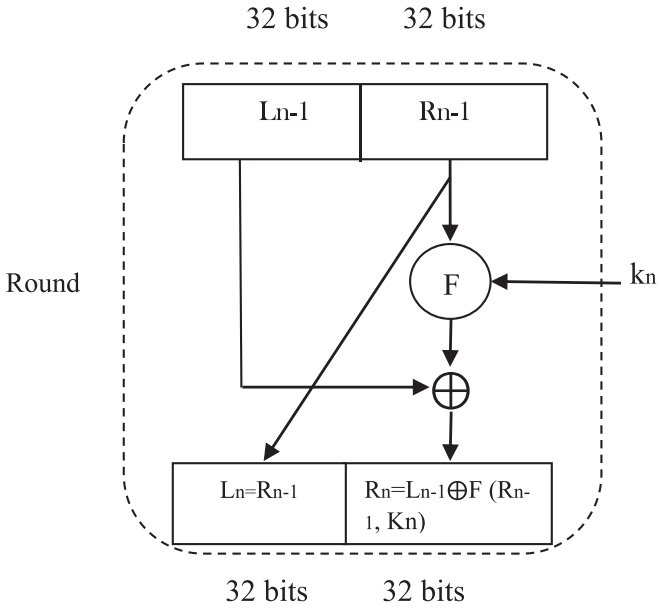

Fig. 4. One round of Des (Haneen Alabdulrazzaq, 2022).

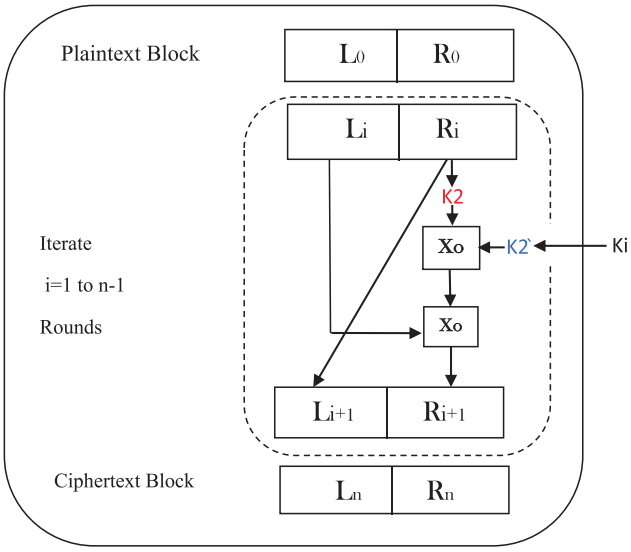

Fig. 5. Basic Operation of the proposed DES algorithm.

- The third segment represents the **third main state table**, denoted as keys **k5** and **k5`**.

In this study, a variety of tables containing more states are employed to increase the unpredictability of the process. Also, a new bit process manipulation is presented since the famous DES algorithm depends on the XOR function, which acts on [0, 1], but the new suggested approach employs the function (**Xo**), which works on six variable block bit sizes: **2**, **4**, **6**, **8**, **10**, and **12** state tables.

The values of **(k3, k4, k5)** and **(k3`, k4`, k5`)** are selected from **2, 4, 6, 8, 10, 12,** or (**0** for only one part of the three as optional)**,** which represent the six state tables used in this research. Note that the sizes of **k2** and **k2`** are equal to the total of the preceding three keys. An illustration of a specification is shown in Table 1.

Moreover, work was done in this study to create extra substate tables from the three main state tables (**k3, k4, k5**) and (**k3`, k4`, k5`**), based on the six state tables (**2, 4, 6, 8, 10, or 12**). Therefore, there are:

- 4 state tables (0, 1, 2, and 3) for the main state, or GF (2^2).
- 16 state tables (0, 1,…., 15) for the main state, or GF (2^4).
- 64 state tables (0, 1,…., 63) for the main state, or GF (2^6).
- 256 state tables (0, 1,…., 255) for the main state, or GF (2^8),
- 1024 state tables (0, 1,…., 1023) for the main state table 10, or GF (2^{10}).
- 4096 state tables (0, 1,…., 4095) for the main state table 12, or GF (2^{12}).

A number of instances of these tables are presented in Tables 2–5 below. After specifying the three main state tables for both (**k3, k4, k5**) and (**k3`, k4`, k5`**), inside **Xo** the function, new processing

Table 1
The keys specification.

	First main state table = k3 or k3` /bits	**Second main state table = k4 or k4`** /bits	**Third main state table = k5 or k5`** /bits	**k3 + k4 + k5 = k2**/bits **k3`+k4`+k5`=k2`** /bits
1	8	8	0	16
2	8	4	4	=
3	4	4	8	=
4	6	10	0	=
5	2	4	10	=
6	12	4	0	=
7	2	2	12	=
:	:	:	:	:
:	:	:	:	:

Table 2
State (**Xo**0) addition in GF (2^{10}).

Xo0	**0**	**1**	**2**	**3**	…	…	**60**	**61**	**62**	**63**	…	…	**885**	**886**	**887**	…	**1021**	**1022**	**1023**
0	0	1	2	3	…	…	60	61	62	60	…	…	885	886	887	…	1021	1022	1023
1	1	0	3	2	…	…	61	60	63	61	…	…	884	887	886	…	1020	1023	1022
2	2	3	0	1	…	…	62	63	60	62	…	…	887	884	885	…	1023	1020	1021
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
1021	1021	1020	1023	1021	…	…	961	960	963	961	…	…	170	165	164	…	1	2	3
1022	1022	1023	1020	1022	…	…	962	963	960	962	…	…	169	166	167	…	0	3	2
1023	1023	1022	1021	1023	…	…	963	962	961	963	…	…	168	167	166	…	3	0	1

Table 3
State (**Xo**1023) addition in GF (2^{10}).

Xo1023	**0**	**1**	**2**	**3**	…	…	**60**	**61**	**62**	**63**	…	…	**885**	**886**	**887**	…	**1021**	**1022**	**1023**
1	197	196	199	198	…	…	249	248	251	250	…	…	954	837	836	…	824	827	826
2	102	103	100	101	…	…	90	91	88	89	…	…	793	998	999	…	923	920	921
3	89	88	91	90	…	…	101	100	103	102	…	…	806	985	984	…	932	935	934
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
1021	149	148	151	150	…	…	169	168	171	170	…	…	1002	789	788	…	872	875	874
1022	166	167	164	165	…	…	154	155	152	153	…	…	985	806	807	…	859	856	857
1023	1023	1022	1021	1020	…	…	963	962	961	960	…	…	128	127	126	…	2	1	0

Table 4
State (**Xo**7) addition in GF (2^{12}).

Xo7	**0**	**1**	**2**	**3**	…	…	**2572**	**2573**	…	…	**3006**	**3007**	**3008**	…	**4093**	**4094**	**4095**
0	0	1	2	3	…	…	2572	2573	…	…	3006	3007	3008	…	4093	4094	4095
1	5	4	7	6	…	…	2569	2568	…	…	3003	3002	3013	…	4088	4091	4090
2	3	2	1	0	…	…	2575	2574	…	…	3001	3000	3015	…	4094	4093	4092
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
4093	4093	4092	4095	4093	…	…	1521	1520	…	…	1091	1090	1085	…	0	3	2
4094	4094	4095	4092	4094	…	…	1522	1523	…	…	1088	1089	1086	…	3	0	1
4095	4095	4094	4093	4095	…	…	1523	1522	…	…	1089	1088	1087	…	2	1	0

Table 5
State (**Xo**4096) addition in GF (2^{12}).

Xo4096	**0**	**1**	**2**	**3**	…	…	**2017**	**2018**	…	…	**3974**	**3975**	**3976**	…	**4093**	**4094**	**4095**
0	19	18	17	16	…	…	2034	2033	…	…	3989	3988	3995	…	4078	4077	4076
1	6	7	4	5	…	…	2023	2020	…	…	3968	3969	3982	…	4091	4088	4089
2	3	2	1	0	…	…	2017	2018	…	…	3973	3972	3979	…	4094	4093	4092
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
4093	12	13	14	15	…	…	2030	2029	…	…	3978	3979	3972	…	4081	4082	4083
4094	17	16	19	18	…	…	2029	2030	…	…	3991	3990	3993	…	4076	4079	4078
4095	14	15	12	13	…	…	2032	2035	…	…	3976	3977	3974	…	4083	4080	4081

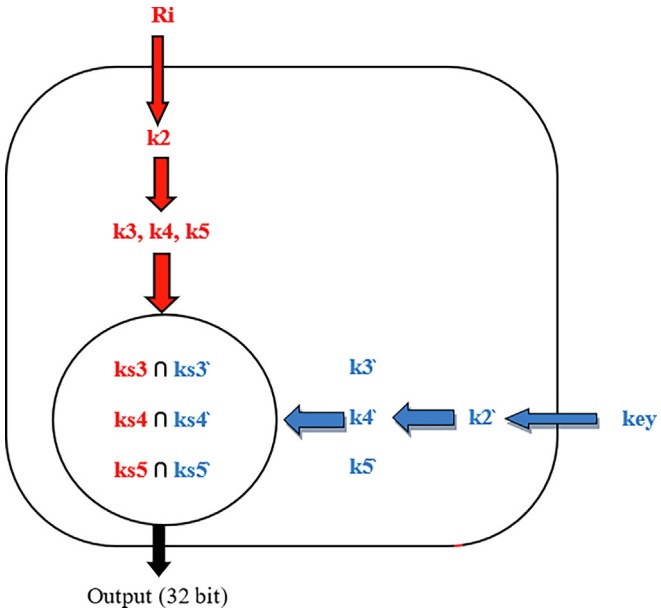

Fig. 6. The **Xo** operation in the proposed DES round.

values are generated in random base (**ks3, ks4, ks5**) and (**ks3`, ks4`, ks5`**), to specify the needed sequence table for each of the (**k3, k4, k5**) and (**k3`, k4`, k5`**). The following equation explains the values of the **Xo** function: The intersection of the plaintext with the key inside the state table produces the ciphertext value.

$$Ciphrtext = Xo(plaintextRi(ks3, ks4, ks5) \cap key(ks3`, ks4`, ks5`)) \quad (1)$$

Fig. 6 depicts the Xo functionality for each cycle of the suggested DES algorithm. The output result is based on the intersection between row and column in tables **ks3, ks4, ks5** and **ks3`, ks4`, ks5`** and the same process repeated in encryption is also repeated in decryption. It is important to note that only seven probabilities were used to segment the keys **k2, k2`** as samples in this study, as depicted in the sequence of Table 1.

Algorithm 1: The suggested DES

Input: – 64 bits original message and key (K).
Output: – 64 bits Cipher block
1. Key is decoded to generate sixteen (48-bit) miner keys Ki, as follows:
 1.1. By using the initial permutation table, the 8 parity bits are removed from the key.
 1.2. Divide K into two equal parts: Ci and Di.
 1.3. Depending on the round, each half of the key is shifted by one or two bits.
 1.4. The two halves are reunited and compressed.
 1.5. The key is reduced from 56 bits to 48 bits via permutation.
2. *__Generate six keys with n bits size each__.
3. Permute the bits of the original text block (64 bits) using the IP table.
4. The result of 2 is divided into two 32-bit halves (L_0, R_0) on the left and right.
For the next 16 rounds, calculate L_i and R_i as follows:
 1.1. $L_i = R_{i-1}$
 1.2. $*\boldsymbol{R_{i-1} = L_{i-1} Xo P` where P` = f(R_{i-1}, K_i) = P(S(E(R_{i-1}) Xo K_i}$
 which is calculated as follows:
 1.2.1. Widen $R_{i-1} = r1, r2 \cdots ..r3$ *from* 32 *to* $48, T = E(R_{i-1})$
 1.2.2. **Apply the Xo operation** $\boldsymbol{T, T` \leftarrow (TXoK_i)}$
 1.2.3. *Output of Step*1.2.2 *is fed into an SBox, Which substitutes key bits and reduces* 48 *bits blockdown to* 32 *bits* $T(\hat{\cdot}\hat{\cdot}) \leftarrow S(T\hat{\cdot})$
 1.2.4. Output of Step 1.2.3 is subject to a P-box to *permute*, $P` \leftarrow P(T``)$
* ***Calculation Xo requires to do the in Ri-1 = Li-1 Xo P in Algorithm 3.1***.
End for
5. Final blocks L_{16} and R_{16} should be exchanged.
6. Inverse permutation (IP^{-1}) is used to transpose the findings.
End.

4.1. The status tables building phase

This section contains examples of the built tables using the mathematical addition operation in Galois Field GF(2^n). The built state-tables were GF (2^2), GF (2^4), GF (2^6), GF (2^8), GF (2^{10}), and GF (2^{12}). As an example, the Tables 2–5 represent samples of the addition tables in GF (2^{10}) and GF (2^{12}) consecutively.

4.2. Phased approach to DES

The suggested DES algorithm is laid out in detail in Algorithm 1 below. Changed procedures are marked in bold.

5. Simulated and evaluated the proposed improvement

The science of cryptography enables secure communication over insecure channels. By employing the key and mathematical processes to create the ciphertext, the message is encrypted. As a result, an attacker acting as a third party and lacking the key cannot decipher the message from the ciphertext. Moreover, the quantity of brute force attacks made by the attackers to estimate the key might be included. The DES algorithm is one of the algorithms that is susceptible to this kind of attack (Stallings, 2023) (Wiemers, 2021). Thus, this work introduces a fresh update to the DES algorithm.

This will raise the degree of security to its maximum by expanding the key space and employing several control keys to estimate the number of bits from the block that will be used to encrypt using the provided table. As a result, estimating the key will be quite difficult. The proposed DES algorithm is evaluated in the present section using three indicators (**complexity analysis, encryption time, throughput, and NIST tests**), as shown below. The algorithm is simulated and evaluated using Microsoft Visual Studio C++ 2019 on an Intel Core I9-12900H 2.90 GHz processor and 32.0 GB of RAM.

A. Security complexity analysis

By calculating the number of possible keys an attacker would need to use three keys with six blocks of either 2, 4, 6, 8, 10, or 12 bits in size and different state tables to decipher the encrypted message with 64 bits, the paper can determine how hard the proposed technique is. The number of potential keys utilized for encryption and decryption is estimated by first determining the complexity of the famous DES algorithm using the binary operation XOR (0, 1) (Haneen Alabdulrazzaq, 2022).

$$2 \times 2\hat{8} \times 32 \times 2 = 2\hat{1} \times 2\hat{8} \times 2\hat{5} \times 2\hat{1} = 2\hat{1}5 \quad (2)$$

Second, the number of potential keys utilized in encryption and decryption is calculated as follows when using the # operation in the DES algorithm with four states (1, 2, 4, or 8) as in (Kareem, 2020).

$$\begin{aligned}&((2^1)^{32} \times (2^2)^{16} \times (2^4)^8 \times (2^8)^4) \times ((2^1)^{32} \times (2^2)^{16} \times (2^4)^8 \times (2^8)^4))^8 \\ &\times (2 \times 2^2 \times 2^4 \times 2^8) \times 32 \times 2 \\ &= 2^{128} \times 2^{1024} \times 2^{15} \times 2^5 \times 2^1 = 2^{1173}\end{aligned} \quad (3)$$

Determine the method's complexity using the proposed DES algorithm. Using just a single key (4, 4, 8) from previous Table 1 of sequence 3, do the following calculation:

$$\begin{aligned}&= ((2^4 \times 16\,(\text{state tables}) \times (2^4 \times 16\,(\text{state tables}) \\ &\times (2^8 \times 256\,(\text{state tables}) \times 3\,(\text{for 48 bits of plaintext of} \\ &(\text{Xo})\,\text{function})) \times (2^4 \times 16\,(\text{state tables}) \\ &\times ((2^4 \times 16\,(\text{state tables}) \times (2^8 \times 256\,(\text{state tables}) \\ &\times 3\,(\text{for 48 bits of key size}))^{16\,\text{DES round}} \times 2^5 \times 2^1\end{aligned} \quad (4)$$

$$= (256 \times 256 \times 65,536 \times 3) \times (256 \times 256 \times 65,536 \times 3)^{16} \times 2^{6}$$
$$= (12,884,901,888) \times (12,884,901,888)^{16} \times 2^{6}$$

Consequently, the difficulty of the recommended DES of 16 rounds for only one key is 4.75947385446610445448940208947 48e + 173.

Comparing the recommended approach with the famous DES method and the updated algorithm from (Kareem, 2020) in terms of computational complexity, Table 6 summarizes the results.

These results demonstrate that the suggested DES algorithm is more computationally difficult than the alternatives. Fig. 7 illustrates the security complexity of the proposed 16-round DES using two algorithms (the famous DES and the enhanced DES with the 4-states given in (Kareem, 2020). These outcomes in Table 6 and Fig. 7 demonstrate how much more difficult the suggested approach was compared to others.

In this study, only seven key-size possibilities have been worked on to divide the keys **k2** and **k2`** into three sections as in the previous Table 1. Therefore, if the total possibilities for this division are calculated according to Table 7, then a huge number of possibilities required from the intruder will appear for the purpose of breaking the proposed DES algorithm. As a result, the suggested research is more resistant to brute-force attacks.

B. Encryption time and throughput

The encryption time is calculated using the amount of time needed to transform the plaintext into an unknown form (ciphertext), which is another metric for evaluating the algorithm's performance. In this situation, the throughput metric is calculated as follows: (velampalli, 2018)

$$\text{Throughput} = \text{plaintext size (in kilobytes)} / \text{total encryption time (ms)} \quad (5)$$

According to Table 8 and Fig. 8, the computation times for the original, modified, and suggested DES algorithms are all identical. However, the results of the suggested method are more effective in terms of complexity evaluation against attacks, making it more challenging for an attacker to recover the clear message from the proposed DES algorithm.

C. Nist tests analysis

The encryption algorithm's output is required to be more chaotic and unpredictable. In this paper, we check the binary pattern inconsistency using 15 statistical tests from NIST (National Institute of Standards and Technology), as shown in Table 9. Calculations are made using numerous ciphertexts created using the original, famous DES and the suggested modified DES. To ensure the output is unpredictable, the probability value (p-value) is set to a value of 0.01 (Sreeja Rajesh, 2019) (Zhang, 2022) Table 9 lists the average tests after they've been calculated. According to Table 8, the majority of tests using the suggested DES algorithm have p-values higher than those using the famous DES. As a result, the majority of tests showed that the suggested DES performed better than the standard DES.

Table 6
Findings from analysis of complexity.

Algorithm	complexity
Famous DES	2^{15} = 32,768
Enhanced DES (HYPERLINK "SPS: refid::bib6" Kareem, 2020)	2^{1173}
Proposed DES	4.75947385446610445448940208947 48e + 173

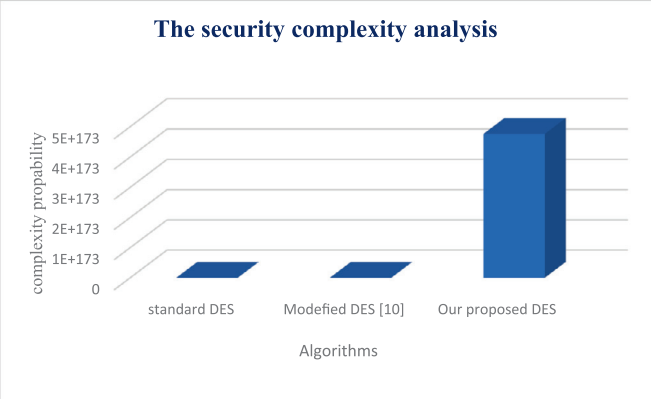

Fig. 7. The security complexity analysis.

Table 7
The total possibilities complexity analysis.

First main state table = k3 or k3` /bits	Second main state table = k4 or k4` /bits	Third main state table = k5 or k5` /bits
2^6	2^6	2^6
$2^6 \times 2^6 \times 2^6 = 2^{18}$ The total possibilities of keys size division		

Table 8
The computation time of the three types (in milliseconds).

File sizes (kb)	famous DES algorithm	Enhanced DES algorithm in (Kareem, 2020)	The proposed DES algorithm
10	5	5	5
50	13	14	14
100	29	30	31
450	113	117	118
Average time	40	41.5	42
throughput	3.815	3.674	3.633

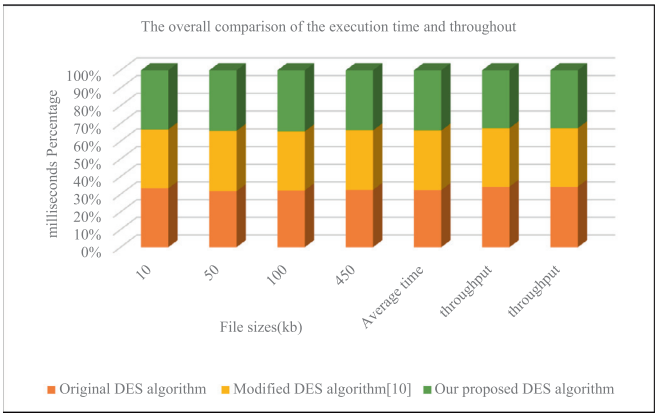

Fig. 8. The overall comparison of the execution time and throughout.

Producing very unpredictable and chaotic results from an encryption algorithm (Moatsum Alawida, 2022).

6. Discussions and results

The proposed DES employs three keys with each of the 16 DES rounds: the first key for encryption or decryption. The second key

Table 9
Nist Tests computation between the standard DES and suggested DES.

Name of Statistical Test	Standard DES		Suggested DES	
	P-Value	Status	P-Value	Status
Approximate Entropy	0.624	pass	0.810	Pass
Block Frequency	0.639	pass	0.749	pass
Cumulative Sums	0.068	pass	0.330	pass
FFT	0.082	pass	0.662	pass
Frequency	0.116	pass	0.410	pass
Linear complexity	0.884	pass	0.623	pass
Longest Run	0.25	pass	0.847	pass
Non-Overlapping Template	0.527	pass	0.517	pass
Overlapping Template	0.480	pass	0.787	pass
Random Excursions	0.591	pass	0.757	pass
Random Excursions Variant	0.761	pass	0.630	pass
Rank	0.432	pass	0.434	pass
Runs	0.001	pass	0.388	pass
Serial	0.649	pass	0.670	pass
Universal	0.326	pass	0.973	pass

specifies the block size of 2, 4, 6, 8, 10, or 12-bits, then generates the three more keys to represent the state tables, as well as a sub-key as an index to select one sub-state table for encryption and decryption. Each DES round will use these component parameters (three keys with changeable block sizes) to improve the difficulty of the suggested DES against attack, according to this method. In contrast to a triple DES, which operates by repeating DES three times, the proposed DES employs three keys during the course of 48 rounds. Additionally, although the triple DES only calculates bits with two attitudes (0 and 1), the suggested DES manipulates bits with a variety of attitudes or states. Therefore, the triple DES has a lower level of complexity than the suggested DES.

7. Conclusion

One of the first and most well-known encryption algorithms used until very recently was DES. Due to its numerous flaws, the DES is regarded as insecure for many applications. These disadvantages include things like the length of the key, the use of two (XOR) functions, the lack of randomness, etc.

As a result, it is required to add new degrees of protection to this algorithm to make it safer. The previous XOR function is changed with a novel operation called the **Xo** function that has more state tables (2, 4, 6, 8, 10, or 12) bit sizes and uses additional keys based on employing GF (2^6), GF (2^{10}) and GF (2^{12}) for the first time in encryption and decryption processes.

This modification is proposed in this research to strengthen the DES algorithm. This will strengthen its defenses against all types of deciphering. The effectiveness of the key is increased when many keys are used rather than just one. The technique becomes more secure when each round uses a configurable block bit size holding a new value. However, this will improve the DES's effectiveness and resistance to brute-force assaults. As was seen in the previous section, the suggested DES algorithm increases the complexity of calculating the keys, but with a slight increase in the time needed to perform the calculations, it is considered acceptable with the desired results for the revival of the DES algorithm.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Bahjat, A.S.R.H., 2009. Proposed new quantum cryptography system using quantum description techniques for generated curves. In: The 2009 International Conference on Security and Management, SAM2009. USA, SAM, LasVegas, p. 2009.

Chen, M., 2021. Accounting data encryption processing based on data encryption standard algorithm. Hindawi Complexity 2021 https://www.hindawi.com/journals/complexity/2021/7212688/.

Y. W. a. X. Dai, "Encryption of accounting data using DES algorithm in Computing Environment," *Journal of Intelligent & Fuzzy Systems,* vol. 39, p. 5085–5095, 2020, DOI:10.3233/JIFS-179994.

M. N. A. Haneen Alabdulrazzaq, "Performance Evaluation of Cryptographic Algorithms: DES, 3DES, Blowfish, Twofish, and Threefish," *International Journal of Communication Networks and Information Security (IJCNIS),* vol. 14, no. 1, April 2022, Available: DOI: 10.17762/ijcnis.v14i1.5262.

Hassan, R.F., 2010. New approach for modifying DES algorithm using 4-States multi keys. Eng. Tech. J. 28 (20) https://etj.uotechnology.edu.iq/article_40805.html.

Kareem, S.M., 2020. New modification on feistel DES algorithm based on multi-level keys. Int. J. Electric. Comput. Eng. (IJECE) 10 (3), 3125–3135. https://doi.org/10.11591/ijece.v10i3.pp3125-3135.

A. H. a. B. Kumar, "A Review Paper on DES, AES, RSA Encryption Standards," in *2020 9th International Conference System Modeling and Advancement in Research Trends (SMART)*, Moradabad, India, 2020, Available: https://ieeexplore.ieee.org/document/9336800.

Moatsum Alawida, J.S.T., 2022. A chaos-based block cipher based on an enhanced logistic map and simultaneous confusion-diffusion operations. J. King Saud Univ. Comput. Inform. Sci. 34 (10), 8136–8151. https://doi.org/10.1016/j.jksuci.2022.07.025.

M. D. A.-H. Qusay Zuhair Abdulla, "Robust Password Encryption Technique with an Extra Security Layer," *Iraqi Journal of Science,* vol. 64, no. 3, pp. 1477-1486, 2023, Available: https://ijs.uobaghdad.edu.iq/index.php/eijs/article/view/6788/3306.

S. D. M. a. A. M. S. Rahma, "Modifying DES Algorithm by Using Diagonal Matrix Based on Irreducible Polynomial," *Journal of Theoretical and Applied Information Technology,* vol. 97, no. 5, 2019.

Sreeja Rajesh, V.P., 2019. A secure and efficient lightweight symmetric encryption scheme for transfer of text files between embedded IoT devices. Symmetry. https://doi.org/10.3390/sym11020293.

W. Stallings, Cryptography And Network Security Principles And Practice, Eight Edition, Pearson Education Limited, 2023.

S. S. a. S. velampalli, "Performance Evaluation for DES and AES Algorithms- An Comprehensive Overview," in *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT-2018), MAY 18th & 19th 2018*, Bangalore, India, 2018, DOI: 10.1109/RTEICT42901.2018.9012536.

F. C. ,. A. D. P. Vincenzo Agate, "Bayesian Modeling for Differential Cryptanalysis of Block Ciphers: A DES Instance," *IEEE Access,* vol. 11, pp. 4809-4820, 2023, Available: https://ieeexplore.ieee.org/ielx7/6287639/10005208/10015012.pdf.

M. j. Wiemers A., "Improving recent side-channel attacks against the DES key schedule," *Journal of Cryptographic Engineering,* 2021, DOI: 10.1007/s13389-021-00279-2.

W. Y. a. L. Yongzhen, "Improved Design of DES Algorithm Based on Symmetric Encryption Algorithm," in *2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA)*, Shenyang, China, 2021.

Zhang, W., 2022. p-value based statistical significance tests: Concepts, misuses, critiques, solutions and beyond. Comput. Ecol. Softw. 12 (3), 80–122.