ISSN 2278-3091
Volume 8, No.6, November – December 2019
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse90862019.pdf
https://doi.org/10.30534/ijatcse/2019/90862019

Developing new Multilevel security algorithm for data encryption-decryption (MLS_ED)

Rashad J. Rasras [1], Ziad Alqadi[2], Mutaz Rasmi Abu Sara[3], Belal Zahran[4]
[1]Department of Computer Engineering, Al-Balqa' Applied University, Jordan, rashad.rasras@bau.edu.jo
[2]Department of Computer Engineering, Al-Balqa' Applied University, Jordan, Natalia_maw@yahoo.com
[3]Department of Computer Science, Taibah University, Al-Medina, Saudi Arabia, mabusara@taibahu.edu.sa
[4] Department of Computer Engineering, Al-Balqa' Applied University, Amman, Jordan, zahrab@bau.edu.jo

ABSTRACT

The desire of secrecy has led people to use data cryptography for effective and secure data transmission by applying advanced methods of cryptography. At the same time rival companies have invested a lot of resources to break into these encryptions, steal secrets and gain an upper hand. This research paper will introduce MLS_ED method of data encryption-decryption; this method will be implemented and compared with other existing standard methods of data cryptography. The aim of this research paper is to provide the users with flexible, efficient and secure method of data encryption-decryption**.**

Key words: Cryptography, PK, round, secret number, encryption time, decryption time, speedup.

1. INTRODUCTION

Data cryptography means encrypting and decrypting data using a selected tool or method, encrypting the data means destroying the data to make it impossible to be understood or used by any other unauthorized party [1], [2].

Standard data encryption-decryption methods perform the cryptography process using secret private key (see figure 1) and applying some mathematical and logical operations, and here the availability of strong, trustworthily, efficient, flexible method of data encryption-decryption is an important building block of people and companies that are more than ever depending on internet data transmission.

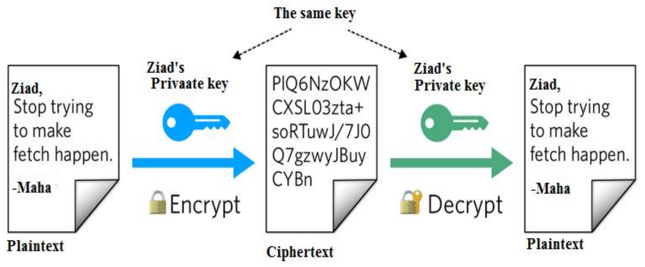
The same key
Ziad's Privaate key
Ziad's Private key
Ziad, Stop trying to make fetch happen. -Maha
PlQ6NzOKW CXSL03zta+ soRTuwJ/7J0 Q7gzwyJBuy CYBn
Encrypt
Decrypt
Plaintext
Ciphertext
Plaintext

Figure 1: Data encryption-decryption process

Flexibility means the ability of the data encryption-decryption method to use various length of the private key, various length of plaintext block, using simple hash functions for key generation [3], ability to change the hash functions.

Method strength means making the hacking process impossible or very difficult in order to trusted by any other user, while method efficiency means minimizing the encryption, decryption times as much as possible.

Encrypted data, also known as ciphertext, appears scrambled or unreadable to a third party person or entity accessing without permission and without knowing the PK and the operations used for encryption (see figure 1).

Data encryption is a very necessary process [3] for any type of secret and confidential data such as plaintext [3], [4] digital image [5], [6], digital voices [7], [8] and any other type of data.

Data encryption standard (DES) is one of the simplest methods of data encryption-decryption. Here the data to be encrypted is to be divided into 64 data blocks as shown in figure 2, encryption of a block of the message takes place in 16 states or rounds as shown in figure 3. From the input key, sixteen 48 bit keys are generated, one for each round. In each round, eight so-called S-boxes are used. These S-boxes are fixed in the specification of the standard [9], [10]. Using the S-boxes, groups of six bits are mapped to groups of four bits. The contents of these S-boxes have been determined by the U.S. National Security Agency (NSA). The S-boxes appear to be randomly filled, but this is not the case. Recently it has been discovered that these S-boxes, determined in the 1970s, are resistant against an attack called differential cryptanalysis which was first known in the 1990s.

The block of the message is divided into two halves. The right half is expanded from 32 to 48 bits using another fixed table. The result is combined with the sub-key for that round using the XOR operation. Using the S-boxes the 48 resulting bits are then transformed again to 32 bits, which are subsequently permutated again using yet another fixed table. This by now thoroughly shuffled right half is now combined with the left half using the XOR operation. In the next round, this combination is used as the new left half.

DES requires a small amount of time for encryption and decryption, but it is not secure and it can be easily hacked by a third party person, because the length of the used PK is small [11], [13].

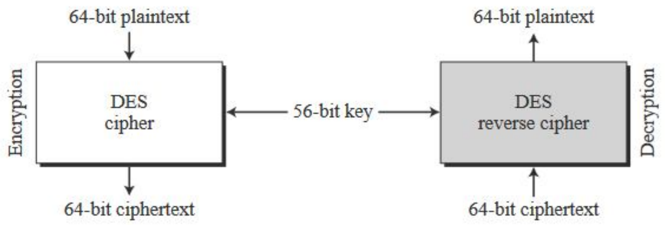

Figure 2: DES encryption-decryption

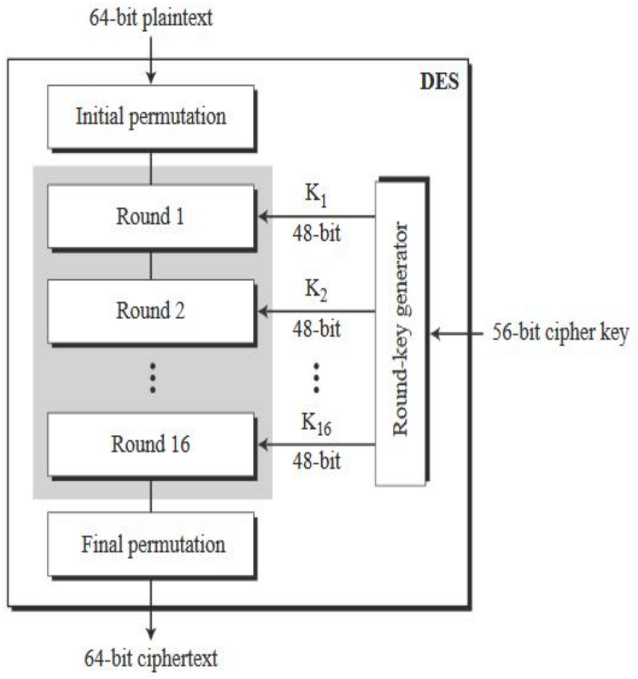

Figure 3: DES structure

To improve the security level of data encryption-decryption, advanced encryption standard (AES) was proposed and used [10], [11], [12].

AES is a symmetric block cipher chosen by the U.S. government to protect classified information and is implemented in software and hardware throughout the world to encrypt sensitive data (see figure 4).

The National Institute of Standards and Technology (NIST) started development of AES in 1997 when it announced the need for a successor algorithm for the DES, which was starting to become vulnerable to brute-force attacks.

This new, advanced encryption algorithm would be unclassified and had to be "capable of protecting sensitive government information well into the next century," according to the NIST announcement of the process for development of an advanced encryption standard algorithm. It was intended to be easy to implement in hardware and software, as well as in restricted environments, and offer good defenses against various attack techniques.

AES comprises three block ciphers: AES-128, AES-192 and AES-256. Each cipher encrypts and decrypts data in blocks of 128 bits using cryptographic keys of 128-, 192- and 256-bits, respectively as shown in figure 5.

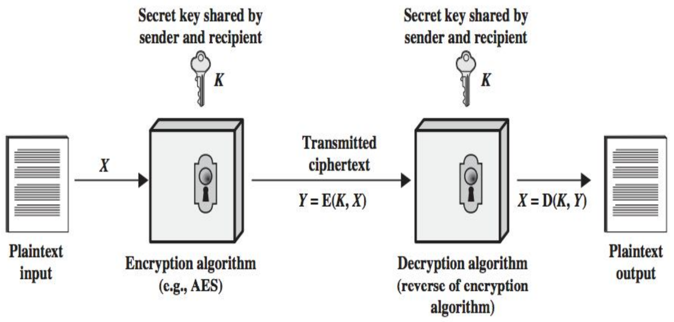
Secret key shared by sender and recipient
K
X
Transmitted ciphertext
Y = E(K, X)
X = D(K, Y)
Plaintext input
Encryption algorithm (e.g., AES)
Decryption algorithm (reverse of encryption algorithm)
Plaintext output

Figure 4: AES encryption-decryption

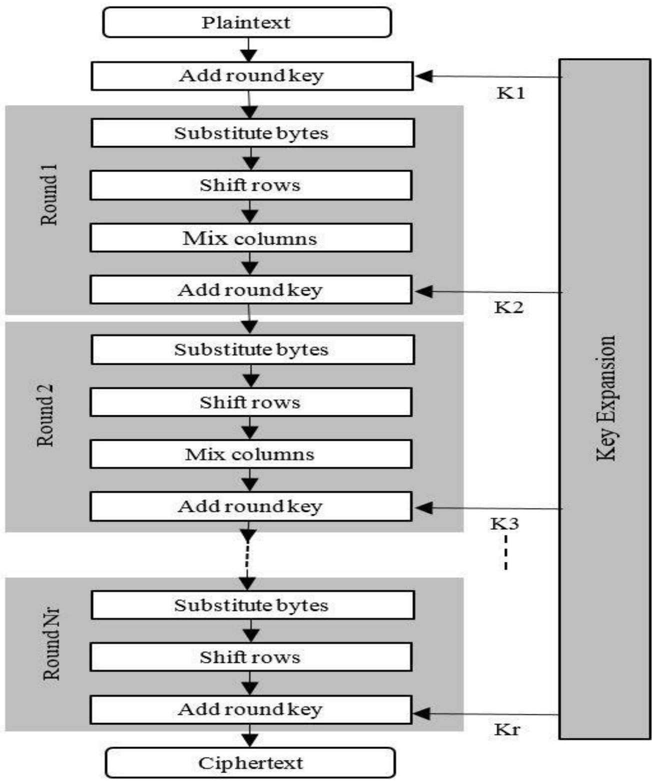

Figure 5: AES structure

The AES encryption algorithm defines a number of transformations that are to be performed on data stored in an array. The first step of the cipher is to put the data into an array; after which the cipher transformations are repeated over a number of encryption rounds. The number of rounds is determined by the key length, with 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys.

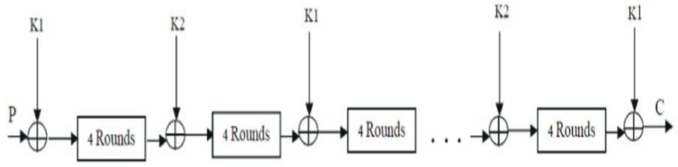

Figure 6: LED structure

Light encryption device (LED) is an SPN(Substitution Permutation Network) [14] type Light weight block cipher was first introduced by Guo et al. in 2011.The step function performed 8 times for the 64 bit key and12 times for the 128-bit keys. The keys used in LED block cipher may vary from 64 bits to 128 bits [15]. The LED algorithm block diagram is shown in Figure 6. The steps during the encryption and decryption process depend on the keys and the S-boxes.

2. ALGORITHM DESCRIPTION

The proposed method of data cryptography has the diagram and structure as shown in figures 7 and 8.

For data encryption-decryption MLS_ED method uses the following:

- Variable size of plaintext blocks (64, or 128, or 256 bits).
- Variable size of PK (64, 128, 256, 512 bits).
- Variable number of rounds (from zero to number related to the secret number).
- Variable secret number used to generate sub-keys.
- Simple and changeable hash functions.

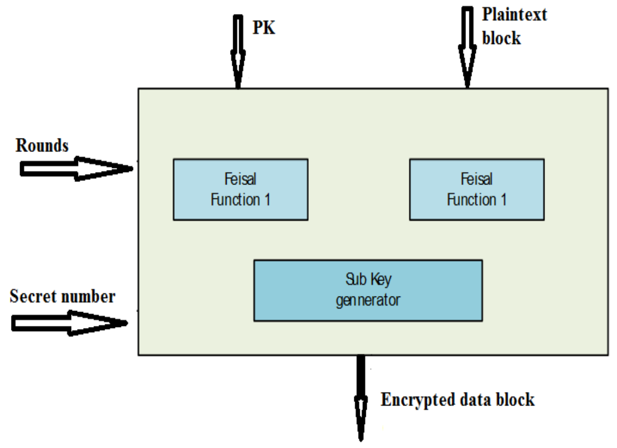

Figure 7:MLS_ED diagram

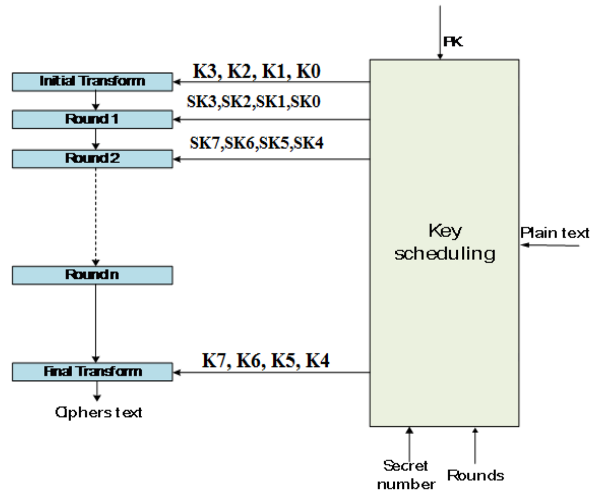

Figure 8: MLS_ED structure

The main features (advantages) of MLS_ED algorithm can be summarized in the following:

- The receiver and the sender must agree on the used private key, this key has a variable length (64, 128, 256 bits or bigger), this key is to be secret and to be used by both the sender and the receiver to generate a work keys (WK1, WK2, …, WK8) by mean of applying a selected hash function, for a 64 bits a plaintext block, we need 4 work keys in the initial transformation, and 4 work keys in the final transformation.
- The receiver and the sender must agree on the used secret number, this number is used to generate sub-keys (SK) (4 for each round) by mean of applying another selected hash function.
- The receiver and the sender must agree on the number of used rounds to encrypt-decrypt data, here each round requires 4 SK.
- The data to be encrypted is to be divided into equal blocks, and here the block size may be variable (64, 128, 256 bits and may be bigger).
- Each data block must be divided into 8 bits partitions; the number of petitions is equal to the number of generated WK divided by 2.
- The algorithm uses two Feisal functions F0 and F1 to implement each round operations, these functions are not fixed, and they are subjected to changes, but the receiver and the sender must use the same functions, such as shown in the equations 1:

$$\boxed{\begin{aligned} F_0(x_i) &= x_i \ll 3 \oplus x_i \ll 5 \oplus x_i \ll 7 \\ F_1(x_i) &= x_i \ll 2 \oplus x_i \ll 4 \oplus x_i \ll 6 \end{aligned}} \quad (1)$$

Encryption phase

This phase can be implemented applying the following tasks:

- Initialization:

In this task the following operation must be performed:

✓ Dividing the data into equal blocks.
✓ Dividing each block into equal partitions.
✓ Selecting PK.
✓ Generating WKs from PK.
✓ Selecting the secret number.
✓ Generating SKs from the secret number.
✓ Selecting the number of rounds (from 1 to number of SKs divided by 4).

- Initial transformation:

From the partitions and work keys generate new partitions X as shown in equation 2:

$$\boxed{\begin{array}{l} X_{i,1} = mod(P_1 + WK_1, 256)\ ,\ X_2 = P_2 \\ X_3 = P_3 \oplus WK_2\ ,\ X_4 = P_4 \\ X_5 = mod(P_5 + WK_3, 256)\ ,\ X_6 = P_6 \\ X_7 = P_7 \oplus WK_4 \quad X_8 = P_8 \end{array}} \qquad (2)$$

- Round Operations:

For each round use Feisal functions to generate new partitions as shown in equation 3:

$$\boxed{\begin{array}{l} X_{(i+1,1)} = X_{(i,0)}; X_{(i+1,3)} = X_{(i,2)}; X_{(i+1,5)} = X_{(i,4)}; X_{(i+1,7)} = X_{(i,6)} \\ X_{(i+1,0)} = X_{(i,7)} \oplus ((F_0(X_{(i,6)}) mod SK_{(4i+3)}) \\ X_{(i+1,2)} = X_{(i,1)} \oplus ((F_0(X_{(i,0)}) mod SK_{(4i+2)}) \\ X_{(i+1,4)} = X_{(i,3)} \oplus ((F_0(X_{(i,2)}) mod SK_{(4i+1)}) \\ X_{(i+1,6)} = X_{(i,5)} \oplus ((F_0(X_{(i,4)}) mod SK_{(4i)}) \end{array}} \qquad (3)$$

- Final transformation:

Generate the cipher block applying equations 4:

$$\boxed{\begin{array}{l} C_1 = mod(X_2 + WK_5, 256)\ ,\ C_2 = X_3 \\ C_3 = X_4 \oplus WK_6\ ,\ C_4 = X_5 \\ C_5 = mod(X_6 + WK_7, 256)\ ,\ C_6 = X_7 \\ C_7 = X_8 \oplus WK_8 \quad C_8 = X_1 \end{array}} \qquad (4)$$

Decryption phase

This phase can be implemented applying the following tasks:

- Initialization:

In this task the following operation must be performed:

- ✓ Dividing the cipher data into equal blocks.
- ✓ Dividing each block into equal partitions.
- ✓ Getting PK.
- ✓ Generating WKs from PK.
- ✓ Getting the secret number.
- ✓ Generating SKs from the secret number.
- ✓ Getting the number of rounds (from 1 to number of SKs divided by 4).

- Initial transformation:

From the partitions and work keys generate new partitions X as shown in equation 5:

$$\boxed{\begin{array}{l} X_2 = mod(P_1 - WK_5, 256)\ ,\ X_3 = P_2 \\ X_4 = P_3 \oplus WK_6\ ,\ X_5 = P_4 \\ X_6 = mod(P_5 - WK_7, 256)\ ,\ X_7 = P_6 \\ X_8 = P_7 \oplus WK_8\ ,\ X_1 = P_8 \end{array}} \qquad (5)$$

- Round Operations:

For each round use Feisal functions to generate new partitions as shown in equation 3.

- Final transformation:

Getting the decrypted block by applying equation 6:

$$\boxed{\begin{array}{l} C_1 = mod(X_1 - WK_1, 256)\ ,\ C_2 = X_2 \\ C_3 = X_3 \oplus WK_2\ ,\ C_4 = X_4 \\ C_5 = mod(X_5 - WK_3, 256)\ ,\ C_6 = X_6 \\ C_7 = X_7 \oplus WK_4\ ,\ C_8 = X_8 \end{array}} \qquad (6)$$

2.1 Algorithm implementation and Experimental Results

DES, AES, LED and the proposed MLS_ED methods were implemented in several experiments, below are the results of each experiment:

Experiment 1: Efficiency calculation

The four mentioned above methods were implemented using selected plaintext block and PK as shown in table 1:

Table 1: Used blocks and PKs

Method	Plaintext block size	PK size
DES	64	56
AES	64	128
LED	64	128
MLS_ED	64	64

Table 2: Encryption and decryption times

Test	DES		AES		LED		MLS_ED	
	E. time	D. time	E. time	D. time	E. time	D. time	E. time	D. time
1	0.0640	0.0574	0.7256	0.9822	0.1271	0.1907	0.0960	0.0545
2	0.0632	0.0573	0.7258	0.9826	0.1274	0.1902	0.0967	0.0540
3	0.0636	0.0579	0.7259	0.9823	0.1278	0.1908	0.0964	0.0540

4	0.0635	0.0570	0.7257	0.9830	0.1270	0.1906	0.0969	0.0543
5	0.0639	0.0578	0.7252	0.9827	0.1271	0.1901	0.0965	0.0540
6	0.0638	0.0580	0.7254	0.9824	0.1272	0.1902	0.0964	0.0544
7	0.0635	0.0580	0.7259	0.9827	0.1272	0.1906	0.0968	0.0547
8	0.0630	0.0578	0.7259	0.9823	0.1276	0.1906	0.0965	0.0541
9	0.0638	0.0574	0.7254	0.9824	0.1273	0.1904	0.0962	0.0540
10	0.0634	0.0575	0.7259	0.9829	0.1272	0.1906	0.0967	0.0546

From table 2 we can see that the efficiency factors for the proposed method are closed to DES efficiency factors and more better than the efficiency factors of other method (see figure 9)

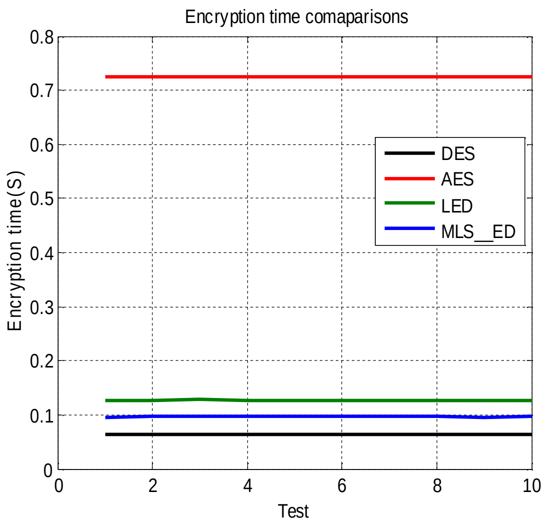

Figure 9: Time comparisons

The averages of encryption and decryption times were calculated and the calculation results are shown in table 3:

Table 3: Average times

Algorithm	Encryption time(s)	Decryption time(s)
DES	0.0630	0.0570
AES	0.7250	0.9820
LED	0.1270	0.1900
MLS_ED (one round)	0.0960	0.0540

Using formula 7 we can calculate the speedup of the proposed method

$$Speedup = \frac{timeofothermethod}{timeoftheproposedmethod} \quad (7)$$

The speedup calculations are shown in table 4:

Table 4: Speedup calculation

Algorithm	Encryption speedup	Decryption speedup
DES	0.6563	1.0556
AES	7.5521	18.1852
LED	1.3229	3.5185

From table 4 we can see that the proposed method has a significant speedup comparing with AES, and LED methods.

Experiment 2: Varying plaintext block and PK sizes

The proposed method can use variable data block and PK sizes, keeping the proposed method efficient, table 5 shows the encryption and decryption times using various block and PK sizes(number of rounds=4).

Table 5: Encryption-decryption times using variable size block and PK

Plaintext size(Bits)	PK size(Bits)	Encryption time(S)	Decryption time(S)
64	64	0.1620	0.0590
64	128	0.1620	0.0590
128	128	0.1631	0.0641
64	256	0.1620	0.0590
128	256	0.1631	0.0641
256	256	0.1701	0.0693
64	512	0.1620	0.0590
128	512	0.1631	0.0641
256	512	0.1701	0.0693
512	512	0.1796	0.0736

From the results shown in table 5 we can see that we can increase the security level of the proposed method with minor losses in the method efficiency (increasing the PK length will increase the security level of the used method of encryption).

Experiment 3: Varying the number of rounds

The proposed method has a good feature, we can use a variable number of rounds to accomplish the encryption and decryption process this number is vary from zero to number of sub-keys divided by 4, or 8, or 16 for a block size of 64 bits, or 128 bits, or 256 bits.

Table 6 shows the encryption-decryption times for various numbers of rounds (block size=64, PK size=128).

From the obtained results shown in table 6 we can see that increasing the number of rounds will increase the encryption and decryption times and this will affect the method efficiency negatively.

From table 6 we can see that even if we use 30 rounds, the proposed method efficiency will better than AES method efficiency and the relationship between the number of rounds and encryption-decryption time is close to linear as shown in figure 10; here we can conclude that keeping the number of rounds secret will increase the proposed method security.

Table 6: Encryption-decryption time for different number of rounds

Rounds	Encryption time(s)	Decryption time(s)
1	0.0960	0.0540
2	0.1210	0.0510
4	0.1620	0.0590
8	0.1670	0.0860
10	0.1940	0.0970
16	0.2340	0.1300
20	0.2600	0.1390
24	0.2760	0.1500
28	0.3060	0.1670
30	0.3340	0.1870

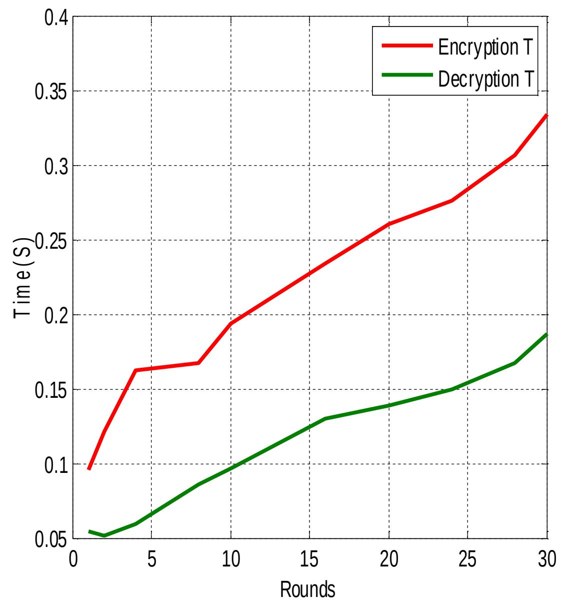

Figure 10: Times as functions of number of rounds

Experiment 4: Measurement MLS_ED algorithm security strength

The security strength of the MLS_ED algorithm was measured using an avalanche effect equation:

$$Avalancheeffect = \frac{Numberofflippedbits}{blocksize} \cdot 100$$

In cryptography, the avalanche effect is the desirable property of cryptographic algorithms, typically block ciphers and cryptographic hash functions, wherein if an input is changed slightly (for example, flipping a single bit), the output changes significantly (e.g., half the output bits flip).

Tables 7 thru 10 show the results of Avalanche effect calculation using various ways:

1) Conditions1

Data block= 'ZIADQADI'

Decimal: 90 73 65 68 81 65 68 73

Hexadecimal: 5A 49 41 44 51 41 44 49

Private Key="ffeeddccbbaa99887766554433221101"

Secret number='0101101'

Rounds=8

Table 7: Conditions 1 results

Plaintext	**Cipher text**	**Number of changed bits**	**Avalanche effect %**
5A49414451414449	711D40C9A85E3ECC		
5A4941445141444**8**	05CA8D60B0D21939	34	53.125
5**2**49414451414449	D807C5C9695ED219	23	35.93
5A**0**9414451414449	38AAA9F2744BC459	36	56.25
5A4**8**404451414449	7F201A335C3A6F34	34	53.125
5A49**5**14451414449	4C0CC7599CCD418C	28	43.75
5A494**0**4451414449	C1D5C4F012C9D214	31	48.43
5A4941**0**451414449	C0B4F13711D8ADFB	36	56.25
5A49414**0**51414449	7792348B1BDDAE3F	29	45.31
5A494144**2**1414449	71CE6276AFC6072E	28	43.75
5A4941445**0**414449	7E4E09FD5718E5C5	33	51.56
Average		**31.2000** **48.75**	

2) Conditions 2

Data block= 'ZIADQADI'

Decimal: 90 73 65 68 81 65 68 73

Hexadecimal: 5A 49 41 44 51 41 44 49

Private Key="ffeeddccbbaa99887766554433221101"

Secret number='0101101'

Rounds=8

Table 8: Conditions 2 results

Secret number	**Cipher text**	**Number of changed bits**	**Avalanche effect %**
0101101	711D40C9A85E3ECC		
1101101	BB681B165CB2E585	40	62.5
0**0**01101	5820B4ADD49B3268	30	46.87
01**1**1101	DE998117C21C95AF	32	50.00
010**0**101	C0A368E878B7955C	29	45.31
0101**0**01	E79AFF1BAAEBF0A5	35	54.68
01011**1**1	4DD9CE37A221DB80	35	54.68
010110**0**	677B4D679EC16BF3	35	54.68
Average		**33.7143**	**52.67**

3) Conditions 3

Data block= 'ZIADQADI'
Decimal: 90 73 65 68 81 65 68 73
Hexadecimal: 5A 49 41 44 51 41 44 49
Private Key="ffeeddccbbaa99887766554433221101"
Secret number='0101101'
Rounds=8

Table 9: Conditions 3 results

PK	Cipher text	Number of changed bits	Avalanche effect %
ffeeddccbbaa99887766554433221101	711D40C9A85E3ECC		
efeeddccbbaa99887766554433221101	8FBE42A397018F00	36	56.2500
ffeeddccbbaa998**0**7766554433221101	43430E7FE070650E	31	48.4375
Ffeeddccbb**8**a99887766554433221101	F7DFD67F9C5BC4F9	30	46.8750
ffeeddccbbaa9988776655443322110**0**	3AB6B0E1C8A46EDF	33	51.5625
ffeeddccbbaa998877665544332211**1**1	AB6B360EC421DB0C	31	48.4375
ffeeddccbbaa99887766554433221**0**01	7F4E3CD2D55FE4C5	30	46.8750
ffeeddccbbaa998**9**7766554433221101	14C5A507A7ECF09D	34	53.1250
ffeeddccbbaa99887766554**0**33221101	47EB019528810F39	33	51.5625
ffeeddccbbaa9988776655443**3**321101	5E4CBFDD6954231C	30	46.8750
ffeeddccbbaa9988**f**766554433221101	91FC7EDA9C7FFF26	29	45.3125
ffeeddccbbaa99887**3**66554433221101	DCCCDB1FCC562108	31	48.4375
Ffeeddc**8**bbaa99887766554433221101	5F72CFD9EB46B391	30	46.8750
ffeed**c**ccbbaa99887766554433221101	BF14652B961FC6FD	30	46.8750
Average			**49.0385**

4) Conditions 4

Data block= 'ZIADQADI'
Decimal: 90 73 65 68 81 65 68 73
Hexadecimal: 5A 49 41 44 51 41 44 49
Private Key="ffeeddccbbaa99887766554433221101"
Secret number='0101101'
Rounds=8

Table 10: Conditions 4 results

Rounds	Cipher text		
8	711D40C9A85E3ECC		
7	CB4D2EA9A7114F6E	30	46.8750
6	6DA77E67879C003F	37	57.81
5	3E0A947D454C8D31	37	57.81
4	3081397F5B435D78	34	53.1250
3	7761B20D5DBA5282	34	53.1250
2	81AB5278EB70AB0D	29	45.3125
1	0C9498F4563A6161	39	60.93
9	687327CA5F2FB272	35	54.68
Average		**34.3750**	**53.71**

From the results shown in tables 7-10, we can see that the Avalanche effect is acceptable and it is always closed to 50%, which means that the security strength of the proposed method is acceptable.

3. CONCLUSION

A proposed MLS_ED of data cryptography was introduced, implemented and tests, the obtained experimental results were compared with experimental results obtained for DES, AES and LED methods. The obtained results showed that the proposed MLS_ED method is very efficient, and strength of MLS_ED method security is acceptable.

The proposed MLS_ED has a high level of security by maintaining various versions of the method through adjusting the data block size, private key size, the secret number value, the number of rounds and by updating the hash functions when needed.

REFERENCES

[1] Jamil Al-Azzeh, Ziad Alqadi, Qazem Jaber, A Simple, Accurate and Highly Secure Method to Encrypt-Decrypt Digital Images, DOI: http://dx.doi.org/10.30630/joiv.3.3.230, 2019.

[2] Rashad J. Rasras; Mohammed Abuzalata; Ziad Alqadi; Jamil Al-Azzeh, Qazem Jaber, Comparative Analysis of Color Image Encryption-Decryption Methods Based on Matrix Manipulation. IJCSMC, Vol. 8, Issue. 3, March 2019, PP 14–26.

[3] Musbah J. Aqel, Ziad A. Alqadi, Ibraheim M. El Emary, Analysis of Stream Cipher Security Algorithm, Journal of Information and Computing Science Vol. 2, No. 4, pp. 288-298, 2007.

[4] Jamil Al-Azzeh, Bilal Zahran, Ziad Alqadi, Belal Ayyoub, Muhammed Mesleh, A Novel Based on Image Blocking Method to Encrypt-Decrypt Color, International Journal on Informatics Visualization, v. 3, issue 1, pp 86-93, 2019. https://doi.org/10.30630/joiv.3.1.210

[5] Jihad Nadir, Ziad Alqadi, Ashraf Abu Ein, Classification of Matrix Multiplication Methods Used to Encrypt-decrypt Color Image, International Journal of Computer and Information Technology, v 5, issue 5, pp 459-464, 2016.

[6] Musbah J. Aqel, Ziad ALQadi, Ammar Ahmed Abdullah, RGB Color Image Encryption-Decryption Using Image Segmentation and Matrix Multiplication, International Journal of Engineering & Technology, 7 (3.13) (2018) 104-107. https://doi.org/10.14419/ijet.v7i3.13.16334

[7] Jihad Nadir, Ashraf Abu Ein , Ziad Alqadi, A Technique to Encrypt-decrypt Stereo Wave File, International Journal of Computer and Information Technology (ISSN: 2279 – 0764) Volume 05 – Issue 05, September 2016.

[8] Majed O. Al-Dwairi, Amjad Y. Hendi, Ziad A. AlQadi, An Efficient and Highly Secure Technique to Encrypt and Decrypt Color Images , Engineering, Technology & Applied Science Research Vol. 9, No. 3, 2019, 4165-4168.

[9] Tingyuan Nie, Chuanwang Song and Xulong Zhi, "Performance Evaluation of DES and Blowfish Algorithms", IEEE International Conference on Biomedical Engineering and Computer Science (ICBECS-2010), pp. 1-4, 23-25 Apr 2010.

[10] E. Thambiraja, G. Ramesh and Dr. R. Umarani, "A Survey on Various Most Common Encryption Techniques", International Journal of Advanced Research

in Computer Science and Software Engineering, Volume 2, Issue 7, pp. 226-233, July 2012.

[11] Diaa Salama Abdul. Elminaam, Hatem Mohamed Abdul Kader and Mohie Mohamed Hadhoud, "Performance Evaluation of Symmetric Encryption Algorithms", IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.12, pp. 280-286, December 2008.

[12] Uma Somani, Kanika Lakhani and Manish Mundra, "Implementing Digital Signatures with RSA Encryption Algorithm to Enhance the Data Security of Cloud in Cloud Computing", 1st International Conference on Parallel, Distributed and Grid Computing (PDGC), pp. 211-216, 2010.
https://doi.org/10.1109/PDGC.2010.5679895

[13] Aman Kumar, Dr. Sudesh Jakhar and Mr. Sunil Makkar, "Comparative Analysis between DES and RSA Algorithm's", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 7, pp. 386-391, July 2012.

[14] Jian Guo, Thomas Peyrin, Axel Poschmann and MattRobshaw, 2011. "The LED Block Cipher", Cryptographic Hardware and Embedded Systems, Springer-Verilog.
https://doi.org/10.1007/978-3-642-23951-9_22

[15] Raja, Raja R. and D. Pavithra, 2013. "Implementation of Hardware Efficient Light Weight Encryption Method", International conference on Communication and Signal Processing, April 3-5, 2013, India, ©2013 IEEE.
https://doi.org/10.1109/iccsp.2013.6577041