International Journal on Advanced Science Engineering Information Technology

Vol.10 (2020) No. 3
ISSN: 2088-5334

A New Method of Data Encryption based on One to One Functions

Osama R. Shahin[a,b,1], Anis Ben Aissa[a,2], Yasser Fouad[c], Hassan Al-Mahdi[d], Mansi Alsmarah[a]

[a] *Department of Computer Science & Information, Jouf University, Gurayat, Saudi Arabia*
E-mail: [1]orshahin@ju.edu.sa; [2]aabenaissi @ju.edu.sa

[b] *Physics and Mathematics Department, Faculty of Engineering, Helwan University, Egypt*

[c] *Department of Computer Science, Faculty of Computers & Information, Suez University, Suez, Egypt*

[d] *Department of Computer Science, Faculty of Computers & Informatics, Suez Canal University, Ismailia 41522, Egypt*

***Abstract*— Due to the rapid growth of computer networks, critical highly confidential information shared across these networks. Accordingly, securing such information from unauthorized intruders has become a vital issue in the field of information technology. In this paper, we present a new algorithm for encrypting and decrypting English plain text based on the well-known Caesar's algorithm and a special type of functions called One to One function. The proposed algorithm is referred to as the One to One function algorithm (OtO). The OtO belongs to a symmetric key concept where the same key is used in both encryption and decryption processes. In the OtO algorithm, the triplet (*K, a, b*) represent the private keys. To speed up the proposed OtO computation, the value of *K* is calculated based on Fibonacci sequence, on eigenvalues, Leslie matrices and Markov chain. This private key *K* only knew to the transmitter and receiver and considered one of the private keys used in the encryption process. When the message arrives at the receiver, it uses the inverse function of the proposed one to one function that used at the transmitter. The proposed OtO algorithm is conducted using MATLAB and its efficiency is checked in terms of encrypted time, decrypted time, and Avalanche Effect. We think that the obtained results are acceptable compared to famous algorithms DES, 3DES, AES and RSA.**

***Keywords*— cryptography; symmetric encryption; fibonacci sequence; eigenvalues; one to one function.**

I. INTRODUCTION

Information is currently a treasure and wealth of peoples and countries, especially with the expansion of communication between the various parts of the globe. The extensive use of technology is due to the development of new technologies such as the Internet, mobile phones, and computers. Therefore the subject of information security has gained particular importance in different areas of daily life. Information security can be defined as a set of techniques, standards, and practices that are applied to information to maintain its integrity [1, 2]. Encryption is a method of converting plain text data into something that looks random and meaningless (ciphertext) to protect information from being accessed by unauthorized people. It was developed by mathematicians such as Francois Vite 1540-1603, John Willias 1616-1703, William F. Friedman 1920, and Lester S. Hill 1929. Ronald Reeves (1977), also known as a science that uses mathematical methods to encrypt and decrypt data. [3]. The exchange of hidden information and encryption is an important area of information security that involves different methods. [4] Encryption provides an essential tool for securing and moving messages from one location to another. There are many encryption algorithms, such as public and private keys and digital signature. Encryption has four objectives: Confidentiality, Integrity, Authentication, and Non-repudiation. Encryption algorithms are divided into two main types: symmetric (or private key) encryption and asymmetric (or public key) encryption [5].

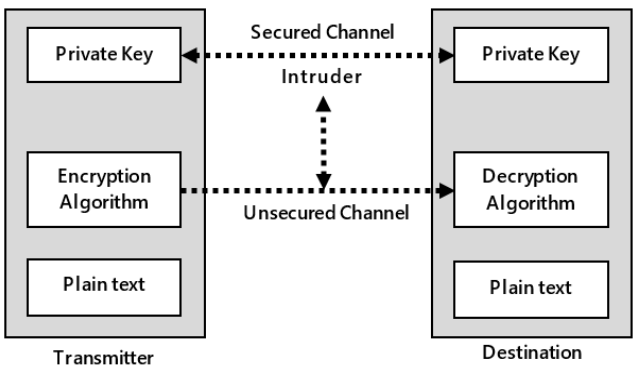

Fig. 1 General Construction of Symmetric Encryption Algorithms.

Traditional encryption is also called symmetric cryptography (Cryptography Symmetric) and uses a single key for data encryption and decryption. The use of this type of encryption began with Julius Caesar, one of the Roman czars (51-58 BC). This type of encryption is still used in military and commercial fields. This type of encryption remains the most used compared to other encryption types [6]. Figure 1 shows an illustration of the symmetric key encryption method.

The Caesar code is one of the symmetric encryption methods in which each alphabet character is encoded by replacing it with another alphabet character. For example, in the case of a three-digit offset, the order of the alphabet is as shown in Table I.

TABLE I
REARRANGE THE ALPHABET USED IN CAESAR'S ALGORITHM

Alphabet	→	**after the offset**	**Alphabet**	→	**after the offset**
A	→	D	N	→	Q
B	→	E	O	→	R
C	→	F	P	→	S
D	→	G	Q	→	T
E	→	H	R	→	U
F	→	I	S	→	V
G	→	J	T	→	W
H	→	K	U	→	X
I	→	L	V	→	Y
J	→	M	W	→	Z
K	→	N	X	→	A
L	→	O	Y	→	B
M	→	P	Z	→	C

Accordingly, to encrypt the word HELLO using the Julius Caesar method, the character H will replace the character K and the character E will be replaced by the character H and so on. It is worth noting that the amount of displacement in this method is the encryption key that only the transmitter and receiver know [7]. Public key encryption is also known as asymmetric cryptography. Figure 2 demonstrates the asymmetric essential encryption process.

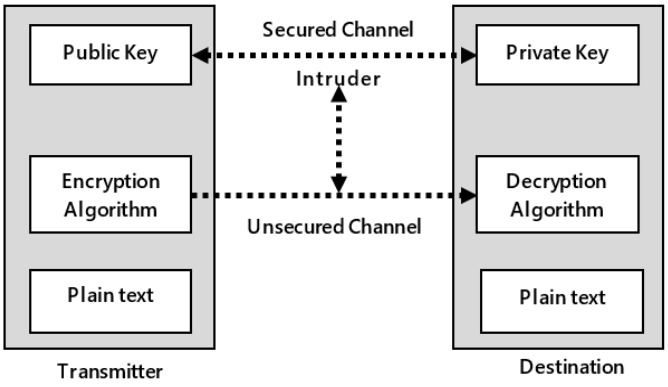

Fig. 2 General Construction of Asymmetric Encryption Algorithms

There are several encryption methods that researchers have developed and compared to choose the best and most suitable for the field of encryption such as DES (Data Encryption Standard) and RSA (Rivest, Shamir, and Adleman) [8] [9]. The DES uses a shared secret key used for 56-bit encryption and decryption. This algorithm converts the plain text to 64 bits ciphertext through a series of operations. This algorithm was also developed in 1978 as a Triple Data Encryption Standard (3DES). The 3DES divide a length of 112 ~ 192 bits into three partial keys. The DES algorithm uses each partial key to encrypt the plain text. RSA involves a public key and a private key. The public key is used to encrypt messages, but encrypted messages can only be decrypted using the private key. These keys are formed for the RSA algorithm in various ways [10]. Blowfish is one of the world's most widely used cryptographic algorithms, introduced in 1993 by Bruce Schneier, one of the world's leading Egyptologists, and president of Counterpane Systems. This algorithm encrypts explicit 64-bit text encryption with a 32 ~ 448-bit two-part encryption key. Many researchers have, in the past, attempted to discover the best encryption and decryption algorithm [11].

The work was then carried out by researchers Joan Daemen and Vincent Rijmen to create the AES: Advanced Encryption Standard (1998). A shared secret key is used for 256-bit encryption and decryption. Converting to 128-bit encrypted text, Singh's work [12] is an excellent example of this because it is a comparison of different symmetrical algorithms, including DES, 3DES, and AES. Similarly, Cornwell's effort [13] was that the Blowfish algorithm could keep the information confidential, according to the researcher. Many Blowfish researchers have found an ideal method of encryption and decryption, including Nadim [14], who has proven that Blowfish algorithms can compete with other algorithms. Besides, Nadim's work concluded that AES is much more sophisticated than DES and 3DES. Similarly, Seth [15] compared three algorithms: DES, AES, and RSA. They concluded that RSA lacks longer coding time and higher memory than the other two algorithms. The rest of this paper will be organized as follows. In section II, we present the methods of the proposed algorithm. Section III, the results, and discussion are represented. Finally, the last section covers the conclusion of this paper.

II. MATERIAL AND METHODS

The proposed One to One function (OtO) algorithm follows the symmetric encryption system for a private key (or symmetric key) that uses the same key for transmission and reception. The OtO algorithm consists of the following five phases. The private keys selection, the conversion table (CT) construction, the one to one function determination and finally the encryption and decryption processes.

A. Select Private Key

One of the private keys that used in the OtO algorithm is the number K, this number is generated from the Fibonacci sequence as follows. The recurrence relation for $n \geq 2$ in Fibonacci sequence can be written as follows:

$$x_n = x_{n-1} + x_{n-2} \quad (1)$$

The value of the private key K is set to x_n. To speed up the OtO algorithm, the direct solution for (1) can be deduced as follows. Firstly, the recurrence relation (1) will be written in the matrix form as follows:

$$x_n = \begin{bmatrix} x_n \\ x_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_{n-1} \\ x_{n-2} \end{bmatrix}, \quad n \geq 2 \tag{2}$$

The characteristic equation of the coefficient matrix, i.e.

$A = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$ will be:

$$\lambda^2 - \lambda - 1 = 0 \tag{3}$$

The solutions $\lambda_1 = \frac{1-\sqrt{5}}{2}$ and $\lambda_2 = \frac{1+\sqrt{5}}{2}$ of equation (3) are the eigenvalues of the matrix *A*. The eigen spaces are given as:

$$E_1 = span\left(\begin{bmatrix} \lambda_1 \\ 1 \end{bmatrix}\right), \; E_1 = span\left(\begin{bmatrix} \lambda_2 \\ 1 \end{bmatrix}\right) \tag{4}$$

Once, $\lambda_1 \neq \lambda_2$, the matrix *A* can be written as follows:

$$A = PDP^{-1} \tag{5}$$

Where $P = \begin{bmatrix} \lambda_1 & \lambda_2 \\ 1 & 1 \end{bmatrix}$, and $P = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$. So, the matrix *A* will equal to:

$$\begin{aligned} A &= \begin{bmatrix} \lambda_1 & \lambda_2 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} \lambda_1 & \lambda_2 \\ 1 & 1 \end{bmatrix}^{-1} \\ &= \frac{1}{L} \begin{bmatrix} \lambda_1 & \lambda_2 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} 1 & -\lambda_2 \\ -1 & \lambda_1 \end{bmatrix} \end{aligned} \tag{6}$$

Where L is the determination of matrix P. The matrix A^j can be written as:

$$A^j = \frac{1}{L} \begin{bmatrix} \lambda_1 & \lambda_2 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \lambda_1^j & 0 \\ 0 & \lambda_2^j \end{bmatrix} \begin{bmatrix} 1 & -\lambda_2 \\ -1 & \lambda_1 \end{bmatrix} \tag{7}$$

From Leslie matrices and Markov chain, we can write x_n as follows:

$$\begin{aligned} x_n &= A^{n-1} x_1 \\ &= \frac{1}{L} \begin{bmatrix} \lambda_1 & \lambda_2 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \lambda_1^{n-1} & 0 \\ 0 & \lambda_2^{n-1} \end{bmatrix} \begin{bmatrix} 1 & -\lambda_2 \\ -1 & \lambda_1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ &= \frac{1}{L} \begin{bmatrix} \lambda_1^n - \lambda_2^n \\ \lambda_1^{n-1} - \lambda_2^{n-1} \end{bmatrix} \end{aligned} \tag{8}$$

Then,

$$x_n = C_1 \left(\frac{1-\sqrt{5}}{2}\right)^n + C_2 \left(\frac{1+\sqrt{5}}{2}\right)^n \tag{9}$$

Where C_1 and C_2 can be evaluated from the first and second terms of Fibonacci sequence. If the first term $f_0 = 0$ then the corresponding equation will be:

$$C_1 + C_2 = 0 \tag{10}$$

If the second term $f_1 = 1$ then the corresponding equation will be:

$$1 = C_1 \left(\frac{1-\sqrt{5}}{2}\right)^n + C_2 \left(\frac{1+\sqrt{5}}{2}\right)^n \tag{11}$$

The values of the constants C_1 and C_2 can be obtained by solving (11) and (12) simultaneously to get:

$$C_1 = -\frac{1}{\sqrt{5}}, \; C_2 = \frac{1}{\sqrt{5}} \tag{12}$$

By substituting for the values of the constants C_1 and C_2 into (9), the required direct formula that required to calculate the private key K is given as :

$$K = \frac{1}{\sqrt{5}} \left[\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n \right] \tag{13}$$

The following algorithm describes the process of generating the private key K, where the symbol $\otimes$ denotes the binary XOR operation.

Algorithm 1: Generating the private key K

1: Input: n_1, n_2, n_3, n_4
2 $n \leftarrow n_i$
2: **for** $i \leftarrow 2$ to 4
3: $n = n \otimes n_i$
4: **end for**
5: $K \leftarrow \frac{1}{\sqrt{5}} \left[\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n \right]$
6: **End**

B. Conversion Table

The conversion table (CT) is created for all characters used in the encoding process based on the value of the private key K. Thus process will be executed at both the sender and receiver sides. Each character is given an integer number as an identifier (ID). The process starts with substituting all characters with their decimal ASCII code values. The ASCII code corresponding to each character is then converted to eight binary formats. XORing every eight binary formats to the private key K. The resulting eight binary formats is converted to decimal number and then to Coded Character (CC). To clarify the generation process of the CT, let the four numbers that responsible for generating the private key K are given as $n_1 = 184, n_2 = 235, n_3 = 224$ and $n_4 = 180$. By using algorithm 1, the value of K is calculated to 13 with binary value 00001101. As shown in Table 2, column five represents the XORing of the binary value for the ASCII codes with $K = 00001101$. The resulting values are converted into decimal and then to their corresponding CC according to the ASCII code table. Now, the last column, CC, is the replacement of the second column. The resulting CT is a modified version of that is used in Caesar's algorithm, where the XORing operation with the value of K replaces the offset used Caesar's algorithm.

C. One to One Function

The one-to-one function has an important mathematical property, which is the only function that can have an inverse function. A function is one-to-one if for any output we have only one input whose image gives that output. In other words, the function $f(\cdot)$ is one-to-one on domain D if $f(x_1) \neq f(x_2)$, whenever $x_1 \neq x_2$ in D. The relationship between a single function and an inverse function is that the field of one is considered a field opposite to the other and vice versa, as shown in Figure 4. Therefore, in this paper, we assume a single-to-one coupling function given as follows:

$$f(x) = a^x + b \tag{14}$$

Where a is positive real number except for $a = 0, 1$, x and b are any real numbers given that $b < x$. Using the inverse function ensures that we have one code for each encoded character, even if this character is repeated it will have a different code depending on its place within the sentence. The variable x expresses the ID of the character to be encrypted. The value of x is obtained from the CT as in Table II.

TABLE II
THE MODIFIED CT OF THE PROPOSED SCHEME

ID	Characters	ASCII	Binary	After XOR with K=13	ASCII	CC
1	A	65	01000001	01001100	76	t
2	B	66	01000010	01001111	79	O
3	C	67	01000011	01001110	78	N
4	D	68	01000100	01001001	73	I
5	E	69	01000101	01001000	72	H
6	F	70	01000110	01001011	75	K
7	G	71	01000111	01001010	74	J
8	H	72	01001000	01000101	69	E
9	I	73	01001001	01000100	68	D
10	J	74	01001010	01000111	71	G
11	K	75	01001011	01000110	70	F
12	L	76	01001100	01000001	65	A
13	M	77	01001101	01000000	64	@
14	N	78	01001110	01000011	67	C
15	O	79	01001111	01000010	66	B
16	P	80	01010000	01011101	93	]
17	Q	81	01010001	01011100	92	\
18	R	82	01010010	01011111	95	_
19	S	83	01010011	01011110	94	^
20	T	84	01010100	01011001	89	Y
21	U	85	01010101	01011000	88	X
22	V	86	01010110	01011011	91	[
23	W	87	01010111	01011010	90	Z
24	X	88	01011000	01010101	85	U
25	Y	89	01011001	01010100	84	T
26	Z	90	01011010	01010111	87	W
27	[	91	01011011	01010110	86	V
28	\	92	01011100	01010001	81	Q
29	]	93	01011101	01010000	80	P
30	^	94	01011110	01010011	83	S
31	_	95	01011111	01010010	82	R
32	@	64	01000000	01001101	77	M

The constants a, b and K collectively represent the private key known only to the transmitter and receiver. The inverse function $f^{-1}(x)$ for the proposed function in (14) must satisfy the following conditions:

1. The function $f(x)$ is a one-to-one function.
2. $f(f^{-1}(x)) = f^{-1}(f(x))$, for all x in $f(x)$ the domain.

The inverse function $f^{-1}(x)$ can be calculated using (14) as follows.

$$x = a^{f(x)} + b$$

which gives

$$f(x)^{-1} = \log_a(x - b) \tag{15}$$

The equation (15) represents the inverse function that the receiver will use it to return the encoded character to the original one.

D. Encryption Process.

First, the transmitter generates the *Priv* = (K, a, b) and sends it to the receiver side over a secure channel. To clarify the encryption process, we try to encrypt the simple plain text Ptxt="DATA_ENCRYPTION_ALGORITHM" with length L. Let $n_1 = 184$, $n_2 = 235$, $n_3 = 224$ and $n_4 = 180$. In addition, the initial values of a and b are set to 1.1 and 0 respectively. The following steps will occur to encrypt this plain text statement:

- Calculate the value of n and the private number K using algorithm 1.
- Generate the CT as depicted in Table 2.
- Extract the characters of the plain text from the CC column in the CT.
- For each extracted character, determine its ID from the first column in the CT as shown in Table III. The ID value the replacement of the x value in (14).

TABLE III
x VALUES FOR CHARACTERS TO BE ENCODED

Letter	ID = x	Letter	ID = x	Letter	ID = x
D	9	Y	20	O	2
A	12	P	29	R	31
T	25	T	25	I	4
A	12	I	4	T	25
_	18	O	2	H	5
E	8	N	3	M	32

E	8	_	18		
N	3	A	12		
C	14	L	1		
R	31	G	10		

The values of the pairing function are calculated one by one for each value of the variable $x_i, i = 1,2,3,\ldots,L$ separately, the initial value of a_1 is set to 1.1 and then incremented gradually by 0.1. Where, the initial value of b_1 is set to 0 and the subsequent values of b_i will be obtained from the following equation:

$$b_i = \mathrm{mod}(bits_{i-1}, x_i),\ i = 2,3,4,\ldots,L \qquad (16)$$

Where $bits_{i-1}$ denoting the number of bits of the binary value of $f(x_i),\ i = 1,2,3,\ldots,L$. Table IV represents the values of the one to one function defined by equation (14), based on the location of each character of the Ptxt.

TABLE IV
X VALUES FOR CHARACTERS TO BE ENCODED.

i	x_i	a_i	b_i	$a_i^{f(x_i)} + b_i$	$bits_{i-1}$
1	9	1.1	0	2.357947434	22
2	12	1.2	10	18.91608392	24
3	25	1.3	24	729.6410256	28
4	12	1.4	4	60.69387755	26
5	18	1.5	8	1485.891892	31
6	8	1.6	7	49.94968553	26
7	3	1.7	2	6.913	22
8	14	1.8	8	3756.133333	31
9	31	1.9	0	437886574	32
10	20	2	12	1048588	21
11	29	2.1	21	2209833492	32
12	25	2.2	7	363552411	29
13	4	2.3	1	28.98412698	23
14	2	2.4	1	6.76	22
15	3	2.5	1	16.625	8
16	18	2.6	8	29479518	25
17	12	2.7	1	150095.6667	37
18	1	2.8	0	2.8	22
19	10	2.9	2	42072.75	18
20	2	3	0	9	4
21	31	3.1	4	1706917413072329	51
22	4	3.2	3	107.8576052	22
23	25	3.3	22	9180122932519	44
24	5	3.4	4	458.3541667	29
25	32	3.5	29	257144606263358340	58

The following vector of the Hexadecimal Codes denotes the ciphertext (Ctxt) of the plain text.

Ctxt = [2.5BA2 716A B795 135E 3CDD 12.EA84 79CC CA9E 7449 D3DF 2D9.A41A 40F3 E616 49EE 609B 3C.B1A1 F588 ADB9 0B4E E89B 5CD.E453 08BB 9064 66B1 E5C1 31.F31E 9744 D59D 0000 80D9 6.E9BA 5E35 3F7C ED91 6873 EAC.2222 1C8A 7A41 E57D 9DBB 1A19 9E6E 10 000C 83B7 6214 1C.FBEF BEEA 3614 391D 0219 6.C28F 5C28 F5C2 8F5C 28F6 10.A 1C1 D25E 2 4A4F.AAAC D9E8 3E42 5AEE 632 2.CCCC CCCC CCCC CCCC CCCD A458.C 9 6B.DB8C 03AE E129 7B23 8CCD 859 69F7 6527 1CA.5AAA AB39 D50D E3EE 5182 391 8FA8 303C 3384].

E. Decryption Process.

The decryption process represents the encryption process but in reverse order. When receiving the Ctxt vector from the receiver side, the Hexadecimal Codes vector is converted into a decimal vector whose elements represent the $x_i, i = 1,2,3,\ldots,L$ values. Using the initial value of the private keys a and b , the inverse function values for each number in the decimal vector is calculated and done as shown in Table V. The inverse function values represent the character places in CT Table II, which is also present at the receiver, so the final decryption form of the receiving Ctxt will be given as Ptxt="DATA_ENCRYPTION_ALGORITHM".

TABLE V
x_i VALUES FOR CHARACTERS TO BE ENCODED

x_i	a_i	b_i	$f(x_i)^{-1}$
2.357947434	1.1	0	9
18.91608392	1.2	10	12
729.6410256	1.3	24	25
60.69387755	1.4	4	12
1485.891892	1.5	8	18
49.94968553	1.6	7	8
6.913	1.7	2	3
3756.133333	1.8	8	14
437886574	1.9	0	31
1048588	2	12	20
2209833492	2.1	21	29
363552411	2.2	7	25
28.98412698	2.3	1	4
6.76	2.4	1	2
16.625	2.5	1	3
29479518	2.6	8	18
150095.6667	2.7	1	12
2.8	2.8	0	1
42072.75	2.9	2	10
9	3	0	2
1706917413072329	3.1	4	31
107.8576052	3.2	3	4
9180122932519	3.3	22	25
458.3541667	3.4	4	5
257144606263358340	3.5	29	32

III. Results and Discussions

Each cryptographic algorithm has its own strong and weak points. Several performance metrics are used to compare algorithms. For our experiment, we will compare the proposed algorithm named "OtO" with the most encryption methods used like DES, 3DES, AES, and RSA. We evaluate the proposed OtO in terms of encrypting time, decrypting time, memory used, and the avalanche effect. We carry out the algorithms using Matlab IDE, personnel computer with 2.4 GHz CPU and 6 GB of memory. In addition, we use plain text with different size such as 256 KB, 512 KB, 1 MB, 2 MB and 3 MB.

A. Encrypting Time

Figure 3 shows that RSA takes the highest encryption time for any file size compared to DES that takes the least time encryption undependable of file size. For the OtO encryption method, we notice that it is faster than RSA and 3DES for the larger file up to 2 MB.

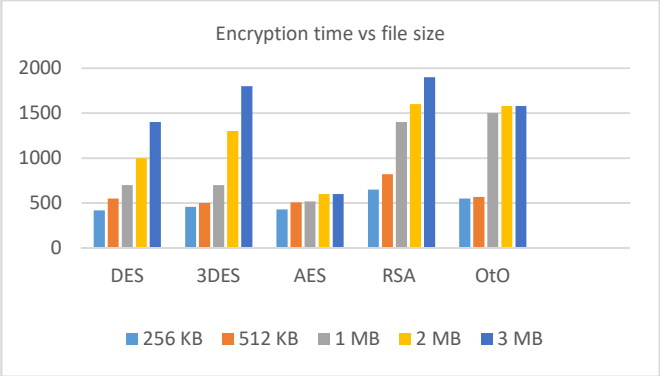

Fig. 3 Encryption time vs file size for DES, 3DES AES, RSA and OtO

B. Decrypting time

Figure 4 shows that the proposed method OtO is a stable time for size file larger than 1 MB and its better than RSA. On the other hand, the proposed OtO takes less time for decryption compared to 3DES and AES when for 512 KB.

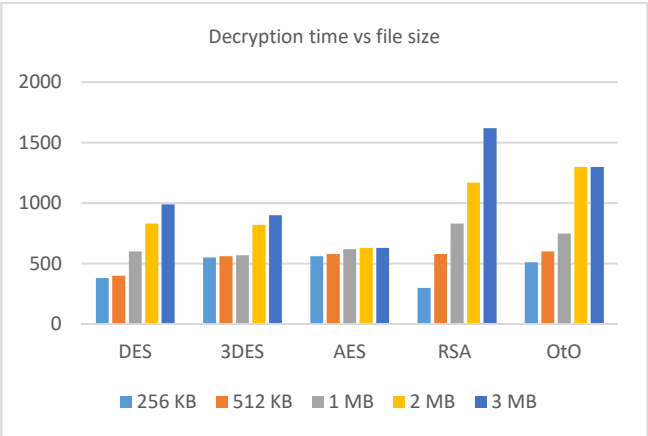

Fig. 4 Decryption time vs. file size for DES, 3DES AES, RSA and OtO

C. Memory Used

Comparing memory used for different algorithms depending on the complexity of code used, type of operation (encryption or decryption), and the size of the file. In our case, we make a test for a file with a size of 3 MB in the encryption test.

TABLE VI
Comparison of Used Memory

Algorithm	Memory used in KB
DES	20.3
3DES	23.6
AES	19.2
RSA	36.9
OtO	30.3

D. Avalanche Effect

The Avalanche Effect measures the correlation between bits of ciphertext and body text bits. In other words, the breakdown test measures the change in ciphertext when a bit in the plain text or master key is changed [16]. To this end, more than one value has been proposed for the encryption keys (a, b), which have the most significant effect in changing the shape of the encrypted text in case of similarity of characters or even increase or decrease the character of a word as shown in Table 7 and 8. It should be noted that the quality of the Avalanche Effect scale is high if the change of a single character, whether increased or decreased in the ciphertext, affects more than fifty percent of the coded text bits from the ciphertext before changing this cell. [16]. The equation of the avalanche effect can be written as follows:

$$Avalanche\ Effect = \frac{bit_{changed}}{bit_{total}} \quad (16)$$

where, $bit_{changed}$ is refers to change only one bit in ciphertext, and bit_{total} is a total number of bits in the cipher text. Tables VII and VIII showed that the proposed algorithm has a very good Avalanche Effect property with an average of 45%.

TABLE VII
Avalanche Values of Plaintext

Plaintext	Ciphertext	Bits	Bits changed	Avalanche test
ENCRYPTION	2.24C2 4C26 8561 0E9E E56F 2.BA5E 353F 7CED 9168 72B 2F.5FAD 4095 716B D30C D7D5 8471 D0D.41A4 15F4 5E0B 4E11 DBCB C AD32 8 CE3B A.7F63 3731 F068 077B 4977 3.9C28 F5C2 8F5C 28F5 C28F 9	205	–	–
EMCRYPTION	2.24C2 4C26 8561 0E9E E56F 16B.D269 3477 0011 9DB7 358C 32.5FAD 4095 716B D311 D7D5 8472 D05.41A4 15F4 5E0B 4E11 DBCB C AD32 8 C112 A.7F63 3731 F068 077B 4977 3.2231 E231 8F5C 28F5 C28F 9	201	96	0.4682926

ENCLYPTION	2.24C2 4C26 8561 0E9E E56F 2.BA5E 353F 7CED 9168 72B 2F.5FAD 4095 716B D30C D7D5 1.6666 6666 6666 6666 6666 CFD.41A4 15F4 5E0B 4E11 DFFD C EED2 8 CE3B A.7F63 3731 3345 7789 8667 3.9C28 F5C2 8F5C 28F5 C28F 9	209	70	0.3414634

TABLE VIII
AVALANCHE VALUES OF PLAINTEXT

Plaintext	Ciphertext	Bits	Bits changed	Avalanche test
DATA	2.5BA2 716A B795 135E 3CDD 12.EA84 79CC CA9E 7449 D3DF 2D7.A41A 40F3 E616 49EE 609B 3C.B1A1 F588 ADB9 0B4E E89B	98	-	-
PATA	F.DCF3 CF45 221E 0D06 4DCD 8.EA85 83C4 E874 3F8F 5971 2D6.A41A 40F3 E616 49EE 609B 3C.B1A1 F588 ADB9 0B4E E89B	100	46	0.4693877
D**E**TA	2.5BA2 716A B795 135E 3CDF C.4CC0 D196 E102 8ED0 92D5 2D9.A41A 40F3 DCDC DCD3 FCAB 3C.B1A1 F588 ADB9 0B4E E89B	99	30	0.4461224
DTA	2.5BA2 716A B795 135E 3CDD 75.656F 17E5 CFD3 118A 6BFF 1A.4C4E C4DF 17A8 1258 473F	72	50	0.51020408

Figure 5 shows the AES exhibits the highest Avalanche effect. However, RSA exhibits the lowest Avalanche.

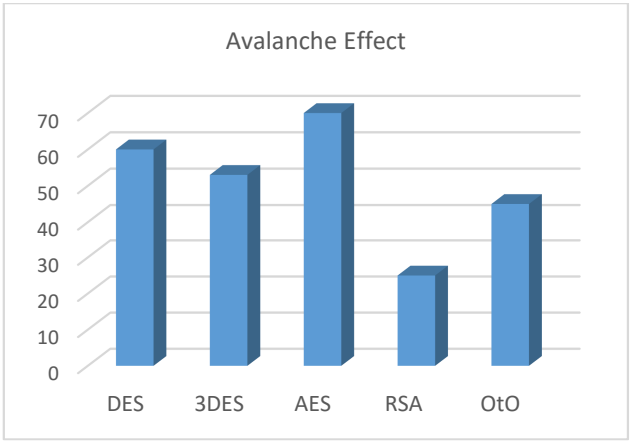

Fig. 5 Avalanche effects of the different algorithms

IV. CONCLUSION

In this paper, we introduced a new algorithm of data encryption based on Caesar's Algorithm and One to One function. The proposed algorithm has four steps easy to apply. We used serial tests like encryption time, decryption time, memory used, and the avalanche effect of evaluating the OtO algorithm. We think that the obtained results are acceptable compared to famous algorithms DES, 3DES, AES, and RSA. In future work, we will try to optimize the algorithm, and we will test brute force attack vulnerability.

REFERENCES

[1] S. Jain and V. Bhatnagar, "Analogy of various DNA based security algorithms using cryptography and steganography," *(ICICT), 2014 International Conference on*, pp 285–291, 2014.

[2] J. Yang, J. Ma, S. Liu, and C. Zhang, "A molecular cryptography model based on structures of DNA self-assembly," *Chinese science bulletin*, vol 59, no. 11, pp. 1192–1198, 2014.

[3] T.Larrieux, Aurelia. *Technical Tools and Designs for Data Protection*. In: Designing for Privacy and its Legal Framework. Springer, Cham, 2018. p. 101-148.

[4] J. L. Philjon and N. Venkateshvara, "Metamorphic cryptography—a paradox between cryptography and steganography using dynamic encryption," *(ICRTIT), 2011 International Conference on*, pp. 217–222. IEEE, 2011.

[5] M. E. Saleh, A. Aly, and F. Omara, "Data security using cryptography and steganography techniques," *(IJACSA) International Journal of Advanced Computer Science and Applications*, vol 7.no. 6, 2016.

[6] H. Al-Mahdi, M. Alruily, O. Shahin, & K. Alkhaldi, "Design and Analysis of DNA Encryption and Decryption Technique based on Asymmetric Cryptography System. computing," vol 10, no. 2, 2019.

[7] K. Chen, "Cryptography. School of Informatics," *University of Manchester*. pp. 32, 2005.

[8] B. Silva, J. Rodrigues, F. Canelo, M. Lopes, J. Lloret, "Towards a cooperative security system for mobile-health applications,". *Electronic Commerce Research*, vol 19, no. 3, pp. 629-654, 2019.

[9] R. Davis, "The Data Encryption Standard in Perspective," *Proceeding of Communication Society magazine*, IEEE, vol. 16, Nov 1978.

[10] R.L.Rivest, A.Shamir, L.Adleman, "A Method for obtaining Digital Signatures and Public-Key Cryptosystem," *Communication of the ACM*, vol 21, Feb 1978.

[11] P. Chandra Mandal "Superiority of Blowfish Algorithm," *International Journal of Advanced Research in Computers Science and Software Engineering*, Vol 2, no. 9, 2012.

[12] G. Singh, A. Kumar, K. S. Sandha, "A Study of New Trends in Blowfish Algorithm," *International Journal of Engineering Research and Application*. vol. 1, no. 2, pp.321-326, 2011.

[13] J. W. Cornwell, G. A. Columbus, "Blowfish Survey," *Department of Computer Science. Columbus: GA Columbus State University*, 2012.

[14] A. Nadeem, M. Y. Javed "A Performance Comparison of Data Encryption Algorithms. In Information and communication technologies," *(ICICT), First international conference on IEEE*, 2005.

[15] S. M. Seth, R. Mishra, "*Comparative Analysis of Encryption Algorithms for Data Communication*," 2011.

[16] P. B. Jayant, N. C. Prashant, "Avalanche Effect of AES Algorithm." *IJCSIT) International Journal of Computer Science and Information Technologies*, vol. 5, no. 3, pp. 3101-3103, 2014.

[17] H. Al-Mahdi O. R. Shahin, Y. Fouad, K. Alkhaldi, "Design and analysis of DNA Binary Cryptography Algorithm for Plaintext," International Journal of Engineering and Technology (IJET), vol. 10, no. 3 , pp. 699 -706, 2018 .