Available Online at www.ijcsmc.com/journal

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 7.056

IJCSMC, Vol. 11, Issue. 11, November 2022, pg.48 – 67

Text Files Protection using Chaotic Logistic Keys

Prof. Ziad Alqadi
Al Balqa Applied University, Faculty of Engineering Technology, Jordan Amman
DOI: https://doi.org/10.47760/ijcsmc.2022.v11i11.006

Abstract:
Text file some times contains secret or personal data which requires for protection from being hacked by unauthorized data hackers. In this research paper a simple, efficient and highly secure method of text file cryptography will be introduced. The text file will be divided into partitions with small sizes to increase the cryptography process throughput. Each partition will be encrypted-decrypted using its associated secret key. Each secret key will be generated from a chaotic logistic map by running a chaotic logistic map model with selected values of chaotic parameters. The chaotic keys will be converted to indices keys to form the secret keys. The secret keys will be used as a lockup tables to apply text file encryption-decryption. The characters of the text file will be rearranged based on the secret key indexes to form the encrypted text file. The characters of the encrypted file will be retrieved by the sequence of the indexes in the secret key. The contents of the decrypted file will be very sensitive to any changes in the chaotic parameters. The PK will have a complicated structure and it will contain the chaotic parameters values and the partition size. The partition size will be a percentage from the total text file size; this complicated structure will increase the key space, making it capable to resist any attack. The proposed method will be implemented using various in size text files, the obtained results will be analyzed to prove the quality, security and efficiency of the proposed method.

Keywords: Cryptography, PK, SK, CLK, CLMM, MSE, PSNR, CC, NSCR, TP.

Abbreviations

The following abbreviations will be used in this research paper:
PK: private key
CLK: chaotic logistic key
SK: secret key
CLMM: chaotic logistic map model
ET: encryption time
DT: decryption time
TP: throughput
MSE: mean square error
PSNR: peak signal to noise ratio
CC: correlation coefficient
NSCR: number of samples change ratio

Introduction

One of the most popular ways to protect text files or secret messages from being hacked is data cryptography. Data cryptography means encrypting the text file before transmission and decrypting the text file after receiving [48-54]. Encrypting the file means destroying this file and making it unreadable and useless, while decrypting the file means recovering a file which is identical to the source file [36-42]. A good method of data cryptography must provide a low quality in the encryption phase and a high quality in the decryption phase [43-50]. The quality (see figure 1) between two files can be measured by the quality parameters MSE, PSNR, CC and NSCR, the method of cryptography must satisfy the requirements listed in table 1 [22-30]:

Table 1: Quality requirements

Quality parameter	Measured between source and encrypted files	Measured between source and decrypted files
MSE	High	0
PSNR	Low	Infinite
CC	Low	1
NSCR	High	0

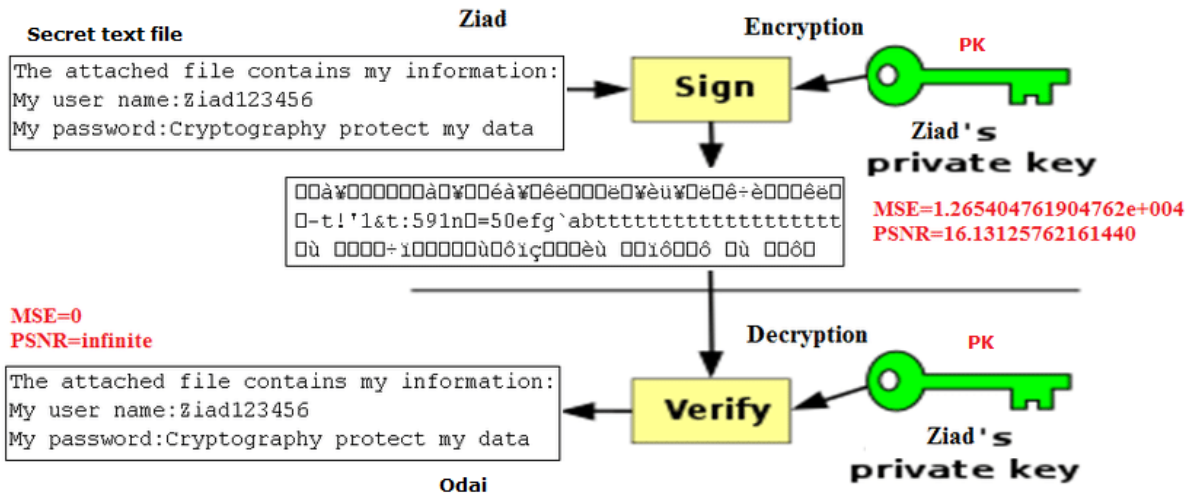

Figure 1: Text file cryptography

The selected PK to be used in encryption and decryption phases must be kept in secret between the data sender and the data receiver. This key is to be used to generate the secret key (keys) needed to apply text file encryption and decryption. In this research paper the CLMM will be used to generate the SKs based on the selected chaotic parameters values, which will form the PK [30-35].

A well known non-linear difference equation that exhibits chaos is the *logistic* map; it is characterized by equation 1 [47].

$$X_{n+1}=a.X_n(1-X_n) \qquad (1)$$

Here, x_n denotes the value of the *state* in the n^{th} iteration and is a real value in the interval [0, 1]. Also, **a** is a real parameter in the interval [0, 4]. The dynamics of the map can be studied by choosing a value for a, assigning an initial value x_0 to the state and iterating repeatedly, to obtain the sequence x_0, x_1, x_2, x_3... Such a sequence of states is called an *orbit*.

For a between 0 and 1, the orbit is found to converge to a fixed point x = 0. For every of a between 1 and 3, the system converges to a different non-zero point. This convergence to fixed points can be easily seen in in figure 2.

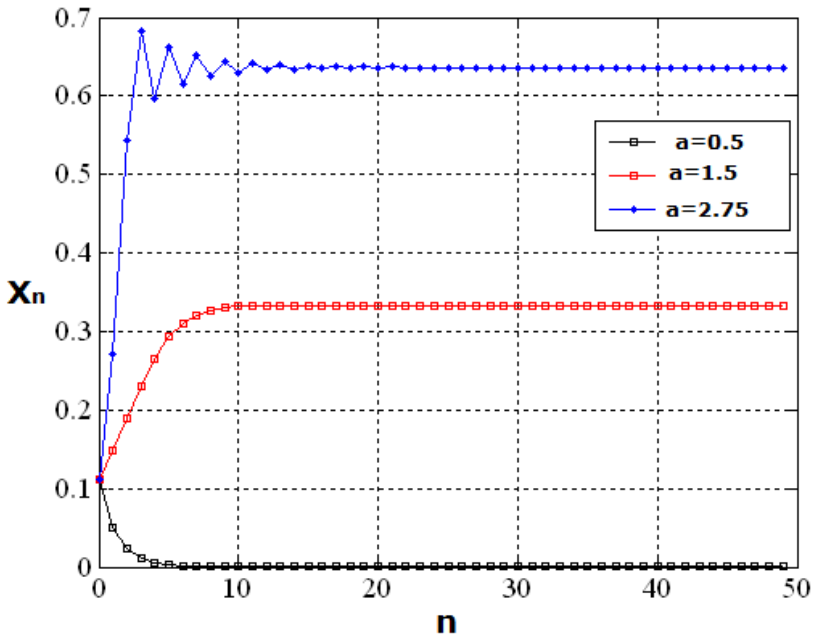

Figure 2: CLM plot for a<3

At a=3, the system undergoes what is known as a period-doubling bifurcation. It means that instead of settling to a fixed point, the system begins to oscillate between two states - a dynamics known as a 2-cycle. Further, if a is progressively increased up to about 3.57, the system undergoes a sequence of period-doubling bifurcations, encountering 4, 8, 16-cycles and so on up to infinity! This usually called a period-doubling cascade. The 2 and 4-cycles can again be seen in figure 3. The blue plot is a 2-cycle and the red plot is a 4-cycle.

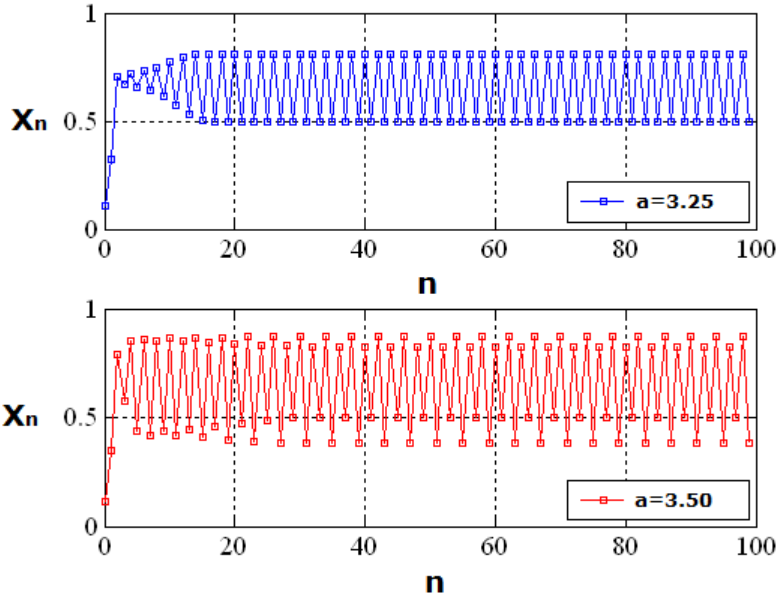

Figure 3: CLM plot for a>3

If the value of a is further increased, it is observed that the orbit neither converges to a fixed point nor **oscillates between a finite number of states. Instead, it appears to behave erratically, without any** apparent pattern over time. This state is called chaos. A chaotic trajectory is shown in figure 4.

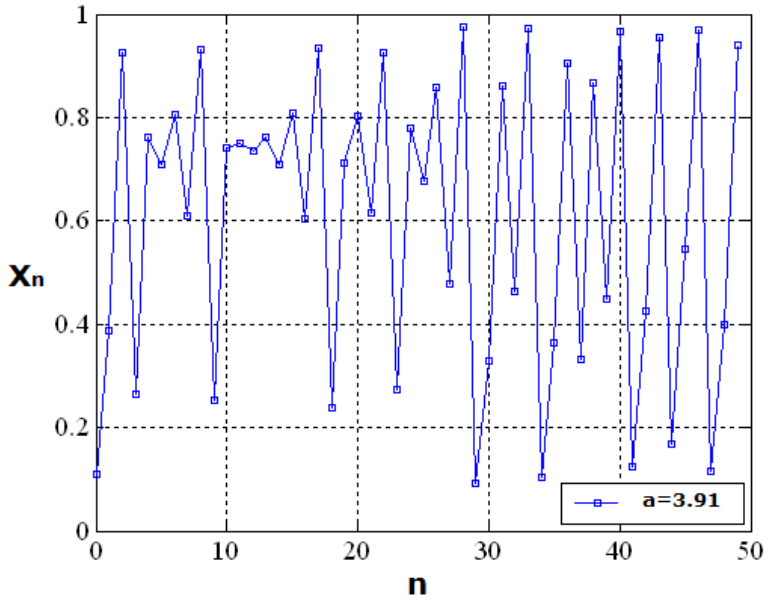

Figure 4: CLM plot for a=3.91

Chaotic systems are sensitive to initial conditions (SIC). It means that if the initial state of a chaotic system is changed by a very small amount, the resulting orbit diverges exponentially from the original orbit. In fact, the two orbits become un-correlated after the passage of a fair amount of time [1-10]. This Sickness of chaotic systems is sometimes called *the butterfly effect*. The divergence of two orbits of the logistic map can be seen in figure 5

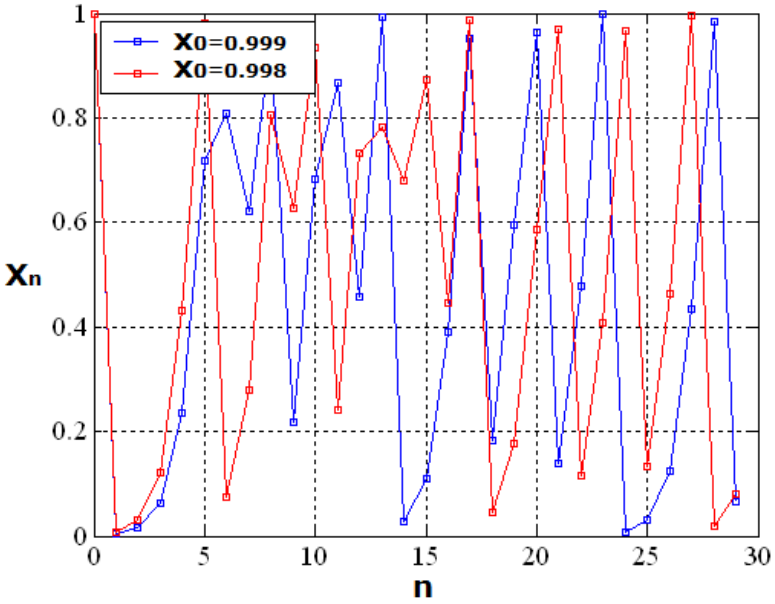

Figure 5: CLM sensitivity

CLM can be easily used to generate various secret keys which can be used in data cryptography to encrypt-decrypt secret important digital data by using chaotic private key to generate a secret key used in algorithm of cryptography. CLK can be generated based on the selected chaotic parameters values, the following sequence of operations can be used to generate CLK:

```
%Chaotic parameters
r1=3.99; x1=0.125;
%Running CLMM
for i=1:20
    x1=r1*x1*(1-x1);
    CLK1(i)=x1;

end
%Changing CLK to index key (SK)
[xx,key1]=sort(CLK1);
```

The length of the CLK must be determine. Here it must be noticed that increasing the length of the CLK will increase the generation time. Using CLK with long CLK will require long time and this will negatively affect the speed of the process of cryptography making the method un efficient, table 2 shows the required times to generate CLK with various lengths:

Table 2: CLKs generation times

Key length (elements)	Generation time (Second)	Key length (elements)	Generation time (Second)
100	0.00042480	**15000**	**0.0757**
200	0.00042920	20000	0.1354
500	0.00065440	25000	0.2169
1000	0.0012	50000	1.9384
2000	0.0040	100000	10.0370
2500	0.0042	200000	57.4469
4000	0.0077	250000	97.3816
5000	0.0108	500000	418.0499
7500	0.0213	1000000	1750.4

From table 2 it is seen that the generation time will rapidly increase when increasing the CLK length over 15000 elements. The generation of the secret key requires time, and the longer the length of the key, the greater the time (see figure 6). When a very long key is used, this will need a very large generation time, which leads us to divide the text file or long secret message into blocks of short length to use short secret keys, and this in turn will reduce the encryption time and decryption time, which will increase the speed of the encryption or decryption process.

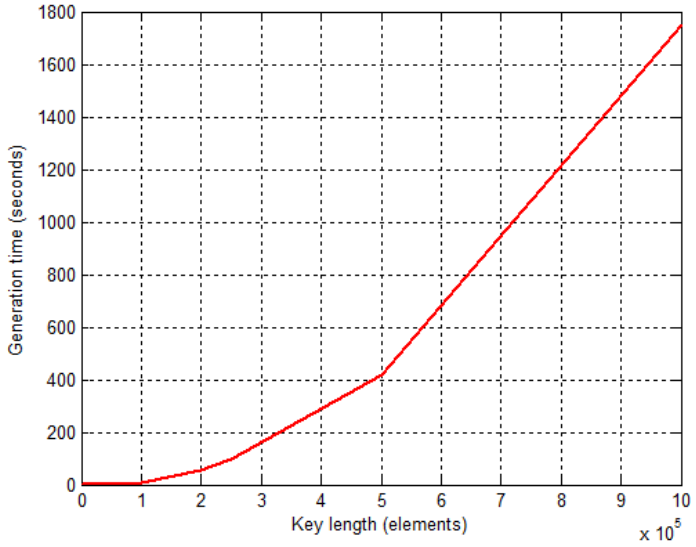

Figure 6: Generation time Vs key length

To overcome this problem, the text file must be divided into partitions, the maximum partition size must be below 15000 characters, and each partition must be treated a lone to increase the cryptography method efficiency.
Another good feature of CLK is its sensitivity to the selected values of the chaotic parameters, any minor changes in the parameters values will lead to change the generated CLK, thus the results of the encryption/decryption phase will be changed. The generated SK is obtained using the CLK by performing the sort function and it will be used as an index key to apply encryption-decryption, SK will be also sensitive to the chaotic parameters values as shown in figures 7 and 8:

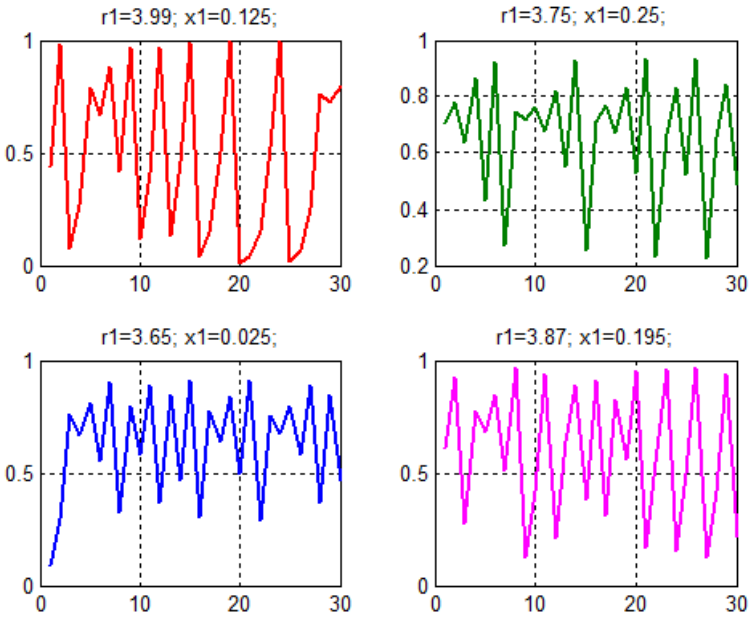

Figure 7: CLK sensitivity

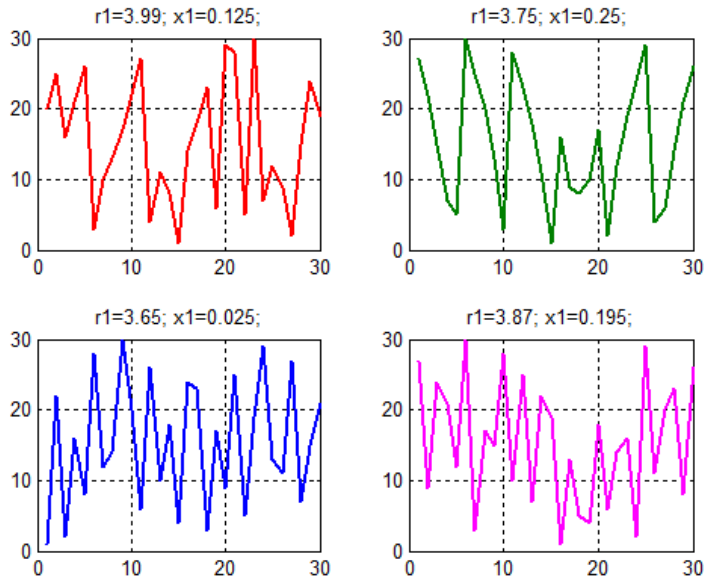

Figure 8: SK sensitivity

Many standards for message cryptography were introduced, here in this research paper we will focus on the most popular standard, the proposed method will be compared with these standard, below a brief description of these standards will be given [1-10]:

DES is a symmetric cryptographic algorithm used for encryption and decryption of data [10]. In DES, only one secret private key is used for both encryption and decryption. The key size of DES is 56-bit. To encrypt-decrypt the data, the data must be divided into blocks with equal size (64 bits blocks), the encryption/decryption process is to be repeated in 16 rounds, each round performs specific logical and arithmetic operations, the key space provided by DES is small and it cannot resist hacking attacks. Another version of DES is 3DES, which increases the PK length to 168 bits making the process of cryptography more secure, but slowly.

AES is the advancement of 3DES algorithm [11]. Basically, AES is based on the Rijndael cipher developed by two cryptographers, Joan Daemon and Vincent Rijmen. AES is different from DES and 3DES due to variables key sizes such as 128,192, and 256 bits [12]. Same like DES and 3DES, AES also performs encryption on blocks which are 128-bit [7]. AES algorithm is used in small devices for encrypting a message to send over a network. Some other applications are monetary transaction [12] and security applications [8] [13].

RSA (Rivest, Shamir and Adleman): RSA stands for Rivest, Shamir and Adleman who introduced the RSA algorithm in 1977 [14]. RSA is an asymmetric cryptographic algorithm [1] which is also used for encryption and decryption of the message. RSA is widely used in transferring of keys over an in secure channel. Due to asymmetric nature, there are two keys used in the algorithm [12-20]. One is public key and second is a private key. The public key is openly accessible to everyone in the cryptosystem and the private key is kept secret by authorized person. RSA provides confidentiality, integrity, authenticity, and non-repudiation of data [15] [11]. RSA is more commonly used in electronic industry for online money transfer [9]. In future, RSA can be used in Java cards [16] ElGamal algorithm was introduced in 1985by Taher ElGamal [16]. ElGamal is an asymmetric key encryption algorithm that is based on the Diffie-Helmankey exchange as an alternative to RSA for public key encryption. ElGamal is also used in digital signature generation algorithm called ElGamal signature scheme [10] [17]. A homomorphic algorithm named paillier used for its semantic security [2].

Blowfish (BF)was designed by Bruce Schneider in 1933, this algorithm uses keys ranging from 32 to 448 bits. Its main purpose was to serve as an alternative to DES [21]. It has the well-known 16 round Feistel Structure and

operates on S-boxes which depend on large keys. It's large key size and range makes the cipher increase its strength and opportunities of use [21].

Other methods were based on chaotic theory, these methods provided a good quality, these methods provided a good quality and throughput [47]:

The Proposed Method

The proposed method uses a complicated PK, this key contains the chaotic parameters to be used to generate 8 SKs. The text file must be divided into partitions with various sizes. For each partition an SK must be generated to apply partition encryption-decryption. Here in the proposed method we select the number of partitions equal 8(this number may be less or more depending on the text file size), thus the PK will contain 23 elements, 16 elements as a chaotic parameters and 7 elements as a partitions percentage, below is an example of one of the selected PKs (see figure 9):

```
PK Example:
r1=3.99; x1=0.125; p1=0.13;
r2=3.99; x2=0.125; p2=0.14;
r3=3.99; x3=0.125; p3=0.10;
r4=3.99; x4=0.125; p4=0.13;
r5=3.99; x5=0.125; p5=0.14;
r6=3.99; x6=0.125; p6=0.10;
r7=3.99; x7=0.125; p7=0.10;
r8=3.99; x8=0.125;
```

Figure 9: PK example

The encryption phase as shown in figure 10 can be implemented applying the following steps:

Step 1: Get the text file, get the PK

Sep2: Use the PK to generate 8 CLKs

Step 3: Convert each CLK to SK

Step 4: Use the PK to divide the text file to partitions

Step 5: Apply encryption of each partition using the associated SK, by rearranging the characters with the indexes in the SK:

Below is the sequence of operation required to apply encryption phase:

```
%Getting each partition size
s1=fix(s*p1);
s2=fix(s*p2);
s3=fix(s*p3);
s4=fix(s*p4);
s5=fix(s*p5);
s6=fix(s*p6);
s7=fix(s*p7);
s8=s-s1-s2-s3-s4-s5-s6-s7;
```


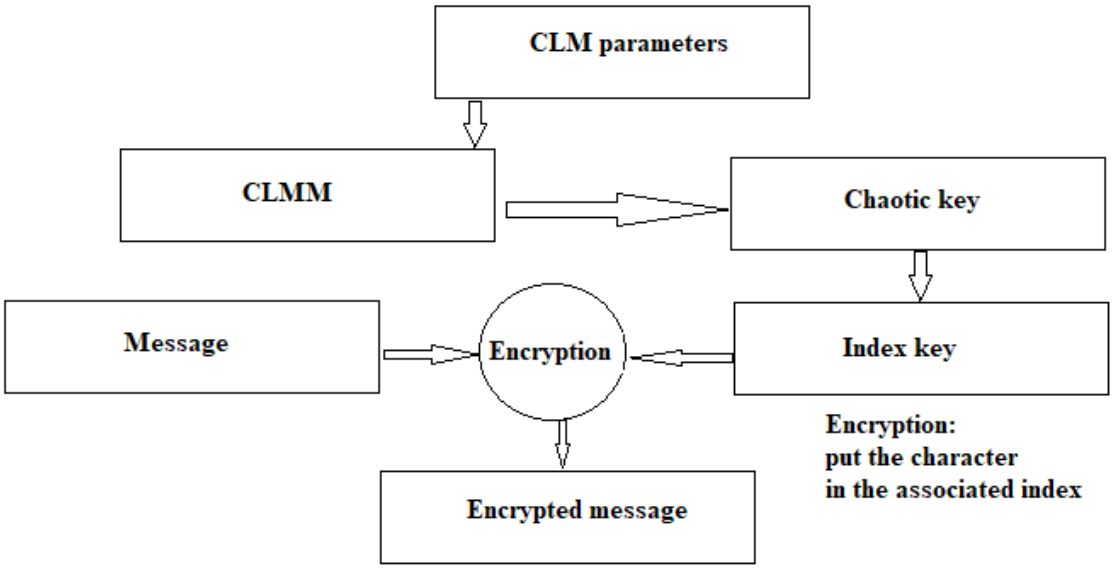

Figure 10: Encryption phase process

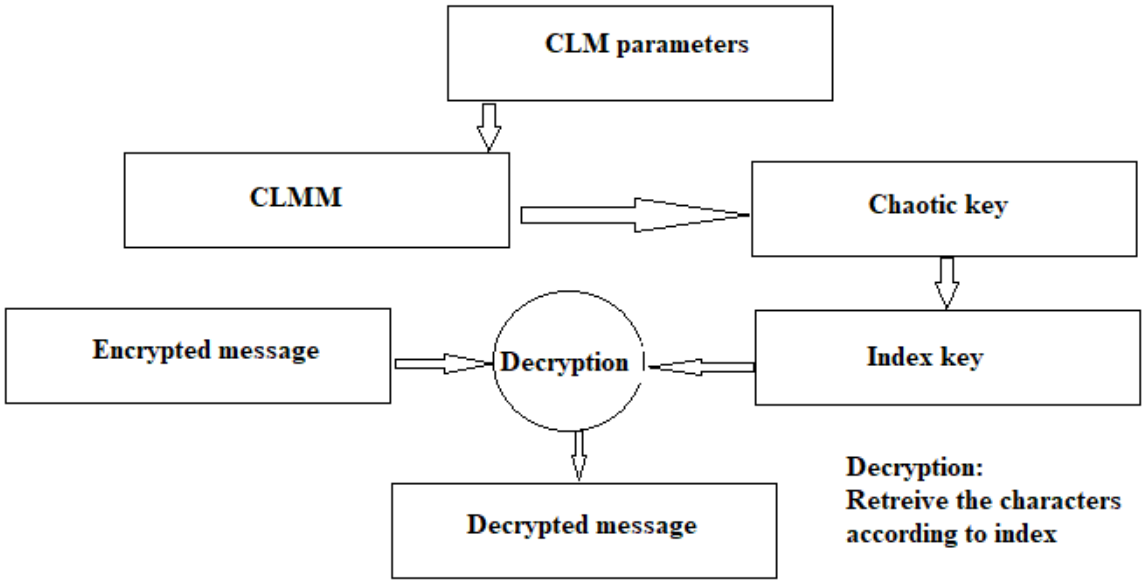

Figure 11: Decryption phase process


```
%Generating SKs
for i=1:s1
    x1=r1*x1*(1-x1);
    CLK1(i)=x1;
 end
[xx,key1]=sort(CLK1);
for i=1:s2
    x2=r2*x2*(1-x2);
    CLK2(i)=x2;
 end
[xx,key2]=sort(CLK2);
for i=1:s3
    x3=r3*x3*(1-x3);
    CLK3(i)=x3;
end
[xx,key3]=sort(CLK3);
for i=1:s4
    x4=r4*x4*(1-x4);
    CLK4(i)=x4;
 end
[xx,key4]=sort(CLK4);
```

```
for i=1:s5
    x5=r5*x5*(1-x5);
    CLK5(i)=x5;
 end
[xx,key5]=sort(CLK5);
for i=1:s6
    x6=r6*x6*(1-x6);
    CLK6(i)=x6;
 end
[xx,key6]=sort(CLK6);
for i=1:s7
    x7=r7*x7*(1-x7);
    CLK7(i)=x7;
end
[xx,key7]=sort(CLK7);
for i=1:s8
    x8=r8*x8*(1-x8);
    CLK8(i)=x8;
end
[xx,key8]=sort(CLK8);
```

```
%Encryption
for i=1:s1
   d=key1(i);
   en1(i)=a1(d);
end
for i=1:s2
   d=key2(i);
   en1(i+s1)=a1(s1+d);
end
for i=1:s3
   d=key3(i);
   en1(i+s1+s2)=a1(s1+s2+d);
end
for i=1:s4
   d=key4(i);
   en1(i+s1+s2+s3)=a1(s1+s2++s3+d);
end
```

```
for i=1:s5
   d=key5(i);
   en1(i+s1+s2+s3+s4)=a1(s1+s2++s3+s4+d);
end
for i=1:s6
   d=key6(i);
   en1(i+s1+s2+s3+s4+s5)=a1(s1+s2++s3+s4+s5+d);
end
for i=1:s7
   d=key7(i);
   en1(i+s1+s2+s3+s4+s5+s6)=a1(s1+s2++s3+s4+s5+s6+d);
end
for i=1:s8
   d=key8(i);
   en1(i+s1+s2+s3+s4+s5+s6+s7)=a1(s1+s2++s3+s4+s5+s6+s7+d);
end
```

The decryption phase as shown in figure 11 can be implemented applying the following steps:

Step 1: Get the encrypted text file, get the PK

Sep2: Use the PK to generate 8 CLKs

Step 3: Convert each CLK to SK

Step 4: Use the PK to divide the text file to partitions

Step 5: Apply encryption of each partition using the associated SK, by getting the characters in a sequence depending on the indexes in SK.

Below is the sequence of operation required to apply encryption phase (the same PK used in the encryption phase must be used in the decryption phase):


```
% Finding partitions sizes
s1=fix(s*p1);
s2=fix(s*p2);
s3=fix(s*p3);
s4=fix(s*p4);
s5=fix(s*p5);
s6=fix(s*p6);
s7=fix(s*p7);
s8=s-s1-s2-s3-s4-s5-s6-s7;
```

```
%Generating SKs
for i=1:s1
    x1=r1*x1*(1-x1);
    CLK1(i)=x1;
end
[xx,key1]=sort(CLK1);
for i=1:s2
    x2=r2*x2*(1-x2);
    CLK2(i)=x2;
 end
[xx,key2]=sort(CLK2);
for i=1:s3
    x3=r3*x3*(1-x3);
    CLK3(i)=x3;
end
[xx,key3]=sort(CLK3);
for i=1:s4
    x4=r4*x4*(1-x4);
    CLK4(i)=x4;
end
[xx,key4]=sort(CLK4);
```

```
for i=1:s5
    x5=r5*x5*(1-x5);
    CLK5(i)=x5;
end
[xx,key5]=sort(CLK5);
for i=1:s6
    x6=r6*x6*(1-x6);
    CLK6(i)=x6;
 end
[xx,key6]=sort(CLK6);
for i=1:s7
    x7=r7*x7*(1-x7);
    CLK7(i)=x7;
 end
[xx,key7]=sort(CLK7);
for i=1:s8
    x8=r8*x8*(1-x8);
    CLK8(i)=x8;
 end
[xx,key8]=sort(CLK8);
```

```
%Decryption
for i=1:s1
   d=key1(i);
   del(d)=dn1(i);
end
for i=1:s2
   d=key2(i);
   del(s1+d)=dn1(i+s1);
end
for i=1:s3
   d=key3(i);
   del(s1+s2+d)=dn1(i+s1+s2);
end
for i=1:s4
   d=key4(i);
   del(s1+s2+s3+d)=dn1(i+s1+s2+s3);
end
```

```
for i=1:s5
   d=key5(i);
   del(s1+s2+s3+s4+d)=dn1(i+s1+s2+s3+s4);
end
for i=1:s6
   d=key6(i);
   del(s1+s2+s3+s4+s5+d)=dn1(i+s1+s2+s3+s4+s5);
end
for i=1:s7
   d=key7(i);
   del(s1+s2+s3+s4+s5+s6+d)=dn1(i+s1+s2+s3+s4+s5+s6);
end
for i=1:s8
   d=key8(i);
   del(s1+s2+s3+s4+s5+s6+s7+d)=dn1(i+s1+s2+s3+s4+s5+s6+s7);
end
```


Figure 12 shows an example that illustrates the encryption and decryption process:

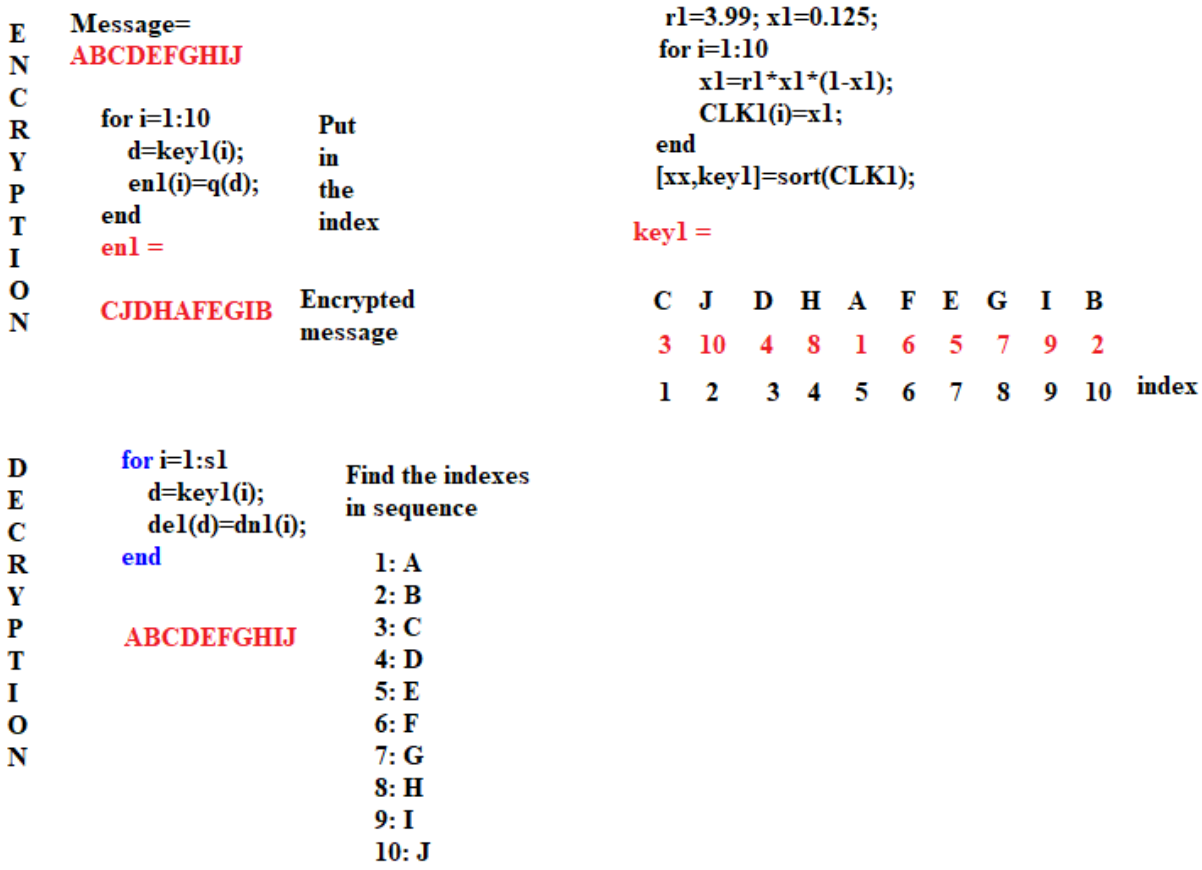

Figure 12: Cryptography example

Implementation and Results Analysis

The proposed method was implemented using various text files and the obtained results were analyzed using various approaches of data analysis methods to prove the achievements, provided by the proposed method, below these analyses will be discussed:

Visual Analysis

A text file shown in figure 13 was encrypted-decrypted using the PK shown in figure 9, figures 14 and 15 show the produced encrypted and decrypted files:

ziad alqadi data steganography data cryptography the house of the rizing sun 123456789 ziad alqadi data steganography data cryptography the house of the rizing sun 123456789 ziad alqadi data steganography data cryptography the house of the rizing sun 123456789ziad alqadi data steganography data cryptography the house of the rizing sun 123456789

Figure 13: Source text file

e oagcgtadd arroddiqzastnyyarpag hal aiptnath4zoe5i9hhsf 6n ya oeae 2r7duti gz sa1hn u p8 i3tgyar aatnaddr alenpp taiaydcstqghoo eufl 5gt s 2r6r hytheut3np8zi7aishge o4ohn stnatanyzqite opipyaa9 adgcrpdorgtaag hadl rdaa n rg u ti t2shhoh4z eushn3 1eoyfieteiag7 q a8aaa5adhsgznn9opirld6dta 5hatr6ondyophi7uga1 pcyghh3 eznoa8ses t u92t f tr ir 4ey

Figure 14: Encrypted text file

ziad alqadi data steganography data cryptography the house of the rizing sun 123456789 ziad alqadi data steganography data cryptography the house of the rizing sun 123456789 ziad alqadi data steganography data cryptography the house of the rizing sun 123456789ziad alqadi data steganography data cryptography the house of the rizing sun 123456789

FIGURE 15: Decrypted text file

The encryption phase destroyed and damaged the source text file by producing a corrupted file, while the decryption phase recovered the source text file by producing a decrypted file identical to the source one, this proves visually that the proposed method satisfies the quality requirements.

Quality Parameters Analysis

The quality between two text files can be measured by the quality parameters MSE, PSNR, CC and NSCR, The MSE and NSCR between the source message and the encrypted one must be high, while the PSNR and CC between them must be low. The MSE and NSCR between the source text file and the decrypted one must be zero, while the PSNR must be infinite, and CC must be equal 1.
MSE and PSNR can be calculated using equations 2 and 3:

$$PSNR - 10\log_{10}\frac{MAX^2}{MSE}\text{dB}, \quad \textbf{(2)}$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2, \quad \textbf{(3)}$$

Where: MAX is the maximum possible value of samples values, N is the total number of samples, xi and yi are the corresponding sample values of the source and encrypted/decrypted SMs.
The value of CC between two text files expresses the dependency between their corresponding sample values. This is another statistical evaluation for testing the quality of the algorithm of data cryptography. Calculating correlation coefficient determines the level of correlation between two text files and the correlation coefficient is always in range [−1, 1]. Values between |1–0.7| is considered as strong correlation (samples from the source files are similar to samples from the encrypted file), correlation between |0.7–0.3| is considered as medium correlation and values between |0.3–0| is considered as weak correlation. Correlation coefficient can be calculated using equation 4:

$$\mathbf{CC}_{xy} = \frac{cov(x,y)}{\sqrt{D(x)}\sqrt{D(y)}}, \quad \textbf{(4)}$$

where

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2,$$

$$D(y) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y})^2,$$

$$cov(x,y) = \sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y}),$$

N is the total number of samples, xi and yi are the sample values of two text files, $\bar{x}$ and $\bar{y}$ are the mean values of samples, and finally cov (x, y) is covariance between both text files.

Number of sample change rate (NSCR) is robustness test for establishing the quality of data cryptography algorithms. The purpose of the test is to compare the corresponding sample values of two text files to show the difference in percent. NSCR can be calculated using equation 5.

$$NSCR = \frac{\sum_{i=1}^{N} D_i}{N} \times 100\%, \quad \textbf{(5)}$$

where

$$D_i = \begin{cases} 1, x_i \neq y_i \\ 0, Otherwise \end{cases}$$

The proposed method was implemented using various text files, the calculated values of MSE between the source text file and the encrypted one was always equal zero, the PSNR was always equal infinite, the CC value was always equal 1, while the NSCR was always equal zero, this prove that the proposed method provided a high quality in the decryption phase, table 3 shows the quality parameters measured between the source text files and the encrypted ones, the results shown in table 7 and 8 prove that the encryption phase provided a low quality SM, thus the provided method satisfied the quality requirements of good cryptography:

Table 3: Quality parameters between the source and encrypted text files

Text file	Size (K bytes)	MSE	PSNR	CC	NSCR
1	1	11231	17.5609	-0.0171	98.6328
2	10	10622	18.1188	0.0174	99.6191
3	50	10927	17.8353	0.00021071	99.5723
4	100	10841	17.9140	0.00017626	99.6230
5	200	10848	17.9077	-0.0014	99.6035
6	300	10776	17.9749	0.0034	99.5977
7	400	10909	17.8523	-0.0056	99.6221
8	500	10848	17.9076	0.00035769	99.6186
9	600	10868	17.8895	-0.0034	99.6069
10	1000	10849	17.9070	-0.00043895	99.5998
Remark		High	Low	Low	High

From table 3 it is seen that the proposed method satisfied the quality requirements by producing a low quality encrypted text files.

Sensitivity Analysis

The produced results of the proposed method are very sensitive to the contents of the PK, any changes in the PK during the decryption phase will be considered as a hacking attack by producing a damaged decrypted text file.

The following two PKs were used, PK1 was to encrypt a text file, while PK2 was used to decrypt the encrypted text file figures 16, 17 and 18 show the produced outputs:

PK1:
r1=3.99; x1=0.125; p1=0.13;
r2=3.99; x2=0.125; p2=0.14;
r3=3.99; x3=0.125; p3=0.10;
r4=3.99; x4=0.125; p4=0.13;
r5=3.99; x5=0.125; p5=0.14;
r6=3.99; x6=0.125; p6=0.10;
r7=3.99; x7=0.125; p7=0.10;
r8=3.99; x8=0.125;

PK2:
r1=3.99; x1=0.125; p1=0.09;
r2=3.99; x2=0.125; p2=0.14;
r3=3.69; x3=0.125; p3=0.10;
r4=3.99; x4=0.125; p4=0.13;
r5=3.99; x5=0.125; p5=0.23;
r6=3.99; x6=0.125; p6=0.10;
r7=3.79; x7=0.025; p7=0.10;
r8=3.99; x8=0.15;

ziad alqadi data stegannography data cryptography the house of the rizing sun 123456789 ziad alqadi data stegannography data cryptography the house of the rizing sun 123456789 ziad alqadi data stegannography data cryptography the house of the rizing sun 123456789ziad alqadi data stegannography data cryptography the house of the rizing sun 123456789 ';

Figure 16: Source text file

e oagcgtadd arroddiqzastnyyarpag hal aiptnath4zoe5i9hhsf 6n ya oeae 2r7duti gz sa1hn u p8 i3tgyar aatnaddr alenpp taiaydcstqghoo eufl 5gt s 2r6r hytheut3np8zi7aishge o4ohn stnatanyzqite opipyaa9 adgcrpdorgtaag hadl rdaa n rg u ti t2shhoh4z eushn3 1eoyfieteiag7 q a8aaa5adhsgznn9opirld6dta 5hatr6ondyophi7uga1 pcyghh3 eznoa8ses t u92t f tr ir 4ey

Figure 17: Encrypted file using PK1

z ad a q di da a s egan ograp y9tize 2iup5dth anh glafgh h aanre sz t ia4oye67oul r8paaya3nhdip1nta end s raatg p 3seh2h1ttfuq5piggi ah 8ador6y euttztyosnc ry t lrnpuni aai tao7oagtgni g o p 9ha4yrt thardroaassoztad2hhq aadpgnc esedg hifau1 a yaso8hiazneah 3e54e 7gqiae td9oihg6do7hlonh tt6rpz5 aardnpisdyn9 u1 f cory ge hs 8 er t h z u a3ya4 et s 2 gpni t

Figure 18: Decrypted file using PK2

From figure 18 it is seen that making any changes in the PK will produce a damaged decrypted file, and the file shown in figure 18 is a destroyed decrypted file.

Security Analysis

The proposed method uses a PK with 23 components; each of them has double data type, so the key space provided by the proposed method will be calculated as shown in equation 6:

$$\text{Key space} = 2^{64(2.L+L-1)} = 2^{64(3.L-1)} \quad (6)$$

L: number of partitions

This key space depends on the number of selected partitions size, and it is a very huge space and it can resist any kind of hacking attacks.

Speed Analysis

Various text files were selected and were processed using the proposed method, the ETs/DTs were measured and the TP was calculated, table 4 shows the obtained results:

Table 4: Speed results

Text file	ET/DT(second)	TP(K bytes per second)
1	0.0060	165.4506
2	0.0396	252.7972
3	0.7986	62.6093
4	3.1215	32.0364
5	11.7753	16.9847
6	29.1212	10.3018
7	55.2042	7.2458
8	89.8848	5.5627
9	135.2702	4.4356
10	395.4130	2.5290

From table 4 we can see that the maximum TP was achieved when the text file size was equal 10 K bytes, increasing the file size will increase the partition size, thus the key generation time will be increased and the TP will be negatively affected decreasing the speed of the method.

So it is recommended to increase the number of partitions when the text file is big, table 5 shows the obtained ET/DTs and TPs using text files with sizes up to 15 k bytes, each text file was divided into 8 partitions using PK1

Table 5: Speed calculations using files up to 15 K byte size

Text file size (K bytes)	ET/DT(second)	TP(K bytes per second)
0.1	0.0050	20.1113
0.2	0.0050	40.1683
0.5	0.0051	98.2357
1	0.0057	176.4384

1.5	0.0062	242.1777
2	0.0073	272.2014
3	0.0100	300.7187
4	**0.0121**	**330.5375**
5	**0.0158**	**317.0275**
10	**0.0392**	**254.8855**
15	0.0791	189.5720

The optimal TPs were achieved when the text file size falls within the range 4 to 10 K bytes, and based on these results we can see that the average throughput equal 300.8168 K bytes per second.

Comparing with other existing methods of data cryptography, the proposed method increased the TP of text file cryptography and it has a significant speedup comparing with most famous methods as shown in table 6

Table 6: speed up of the proposed method

Method	Average encryption TP(K byte per second)	Speedup of proposed method	Average decryption TP(K byte per second)	Speedup of proposed method
DES	148.8889	2.0204	174.0260	1.7286
3DES	86.5633	3.4751	101.1779	2.9731
AES	236.4706	1.2721	255.2381	1.1786
BF	368.5200	0.8163	430.7527	0.6984
RSA	206.6838	1.4554	236.4706	1.2721
ElGamal	124.2658	2.4208	150.2804	2.0017
Non_chaotic ref. [47]	170.3906	1.7655	170.3906	1.7655
Chaotic approach ref. [47]	141.2305	2.1300	141.2305	2.1300
Proposed method	300.8168	1.0000	300.8168	1.0000

Conclusion

A highly secure method to protect text files was introduced. The method used a complicated private key, the number of elements in this key depends on the selected number of partitions, The PK provided a huge key space capable to resist any hacking attack, the produced outputs are very sensitive to the contents of the PK, any changes in the PK during the decryption phase will be considered as a hacking attempt by producing a damaged decrypted text file.

The proposed method was implemented using various text file and it was shown that the proposed method satisfied the quality requirements by producing a low quality encrypted files and producing a high quality decrypted text files based on the calculated quality parameters MSE, PSNR, CC and NSCR between the source files and encrypted/decrypted ones.

The efficiency of the proposed method was tested, the measured ETs/DTs and the calculated TPs values showed that the proposed method is very efficient and it provided a speedup compering with other existing methods of data cryptography.

References

[1] A. Al Hasib and A. A. M. M. Haque, "A comparative study of the performance and security issues of AES and RSA cryptography," Proc. -3rd Int. Conf. Converg. Hybrid Inf. Technol. ICCIT 2008, vol. 2, no. November 2001, pp. 505–510, 2008.

[2] S. Farah, M. Y. Javed, A. Shamim, and T. Nawaz, "An experimental study on Performance Evaluation of Asymmetric Encryption Algorithms," Recent advances Inf. Sci., vol. 8, pp. 121–124, 2012.

[3] G. Singh, "A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security," Int. J. Comput. Appl., vol. 67, no. 19, pp. 975–8887, 2013.

[4] A. Patil and R. Goudar, "A Comparative Survey of Symmetric Encryption Techniques for Wireless Devices," Int. J. Sci. Technol. Res., vol. 2, no. 8, pp. 61–65, 2013.

[5] C. Science and M. Studies, "An Efficient Password Security Mechanism Using Two Server Authentication and Key Exchange," pp. 50–53, 2015.

[6] A. Levi and E. Sava's, "Performance evaluation of public-key cryptosystem operations in WTLS protocol," Proc. - IEEE Symp. Comput. Common., pp. 1245–1250, 2003.

[7] S. S. and K. Annapurna Shetty, "A Review on Asymmetric Cryptography – RSA and ElGamal Algorithm," Int. J. Innov. Res. Comput. Commun. Eng., vol. 2, no. Special issue 5, p. 98, 2014

[8] D. Elminaam, "Performance evaluation of symmetric encryption algorithms," Int. J. Comput. Networks, vol. 8, no. 12, pp. 280–286,2008.

[9] H. Mathur and P. Z. Alam, "Cryptology Algorithm," Int. J. Emerging Trends Technol. Comput. Sci., vol. 4, no. 1, pp. 4–6, 2015.

[10] D. Sukhija, "Performance Evaluation of Cryptographic Algorithms: AES and DES," vol. 3, no. 9, pp. 582–585, 2014.

[11] M. Panda, "Performance Analysis of Encryption Algorithms for Security," pp. 840–844, 2016.

[12] E. Barker, A. Roginsky, G. Locke, and P. Gallagher, "Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths," NIST Spec. Publ., no. January, pp. 800–131, 2011.

[13] H. O. Alanazi, B. B. Zaidan, a. a. Zaidan, H. a. Jalab, M. Shabbir, and Y. Al-Nabhani, "New Comparative Study Between DES, 3DES and AES within Nine Factors," J. Comput., vol. 2, no. 3, pp. 2151–9617,2010.

[14] A. K. Mandal and C. Parakash, "Performance Evaluation of Cryptographic Algorithms: DES and AES," 2012.

[15] A. Sterbenz and P. Lipp, "Performance of the {AES} Candidate Algorithms in {Java}," Third {Advanced Encryption Stand. Candidate Conf. April 13--14, 2000, New York, NY, USA, pp. 161–168, 2000.

[16] R. L. Rivest, A. Shamir, and L. Adleman "A Method for Obtaining Digital Signatures and Public- Key Cryptosystems." Communications of the ACM, vol. 26, no. 1, pp. 96–99, 1983.

[17] M. E. Student, "Algorithms for Secure Cloud," vol. 3, no. 6, pp. 1–9,2014.

[18] G. Bernabé and N. Clarke "Study of RSA Performance in Java Cards,"2013.

[19] P. Nalwaya, V. P. Saxena, and P. Nalwaya, "A cryptographic approach based on integrating running key in feedback mode of ElGamal system," Proc. - 2014 6th Int. Conf. Comput. Intel. Commun. Networks, CICN2014, pp. 719–724, 2014.

[20] X. Li, X. Shen, and H. Chen, "ElGamal digital signature algorithm of adding a random number," J. Networks, vol. 6, no. 5, pp. 774–782, 2011.

[21] S. Sahu, A. Kushwaha, M. Scholar, Performance Analysis of Symmetric Encryption Algorithms for Mobile Ad hoc Network, Published 2014, Computer Science, Corpus ID: 13907600.

[22] Abdullah N. Olimat, Ali F. Al-Shawabkeh, Ziad A. Al-Qadi, Nijad A. Al-Najdawi, Forecasting the influence of the guided flame on the combustibility of timber species using artificial intelligence, Case Studies in Thermal Engineering, Volume 38, 2022, 102379, ISSN 2214-157X, https://doi.org/10.1016/j.csite.2022.102379.

[23] M. Abu-Faraj, and Z. Alqadi, "Image Encryption using Variable Length Blocks and Variable Length Private Key," International Journal of Computer Science and Mobile Computing (IJCSMC), vol. 11, Iss. 3, pp. 138-151, 2022.

[24] M. Abu-Faraj, A. Al-Hyari, and Z. Alqadi, "A Dual Approach for Audio Cryptography," Journal of Southwest Jiaotong University, vol. 57, no. 1, pp. 24-33, 2022.

[25] M. Abu-Faraj, A. Al-Hyari, and Z. Alqadi, "Complex Matrix Private Key to Enhance the Security Level of Image Cryptography," Symmetry, vol. 14, Iss. 4, pp. 664-678, 2022.

[26] M. Abu-Faraj, K. Aldebei, and Z. Alqadi, "Simple, Efficient, Highly Secure, and Multiple Purr- posed Method on Data Cryptography," Traitement du Signal, vol. 39, no. 1, pp. 173-178, 2022.

[27] M. Abu-Faraj, Khaled Aldebe, and Z. Alqadi, "Deep Machine Learning to Enhance ANN Performance: Fingerprint Classifier Case Study," Journal of Southwest Jiaotong University, vol. 56, no. 6, pp. 685-694, 2021.

[28] M. Abu-Faraj, and Z. Alqadi, "Improving the Efficiency and Scalability of Standard Meth- odds for Data Cryptography," International Journal of Computer Science and Network Security (IJCSNS), vol. 21, no.12, pp. 451-458, 2021.

[29] J. Vilkamo and T. Bäckström, "Time-Frequency Processing: Methods and Tools," in Parametric Time-Frequency Domain Spatial Audio, V. Pulkki, S. Delikaris-Manias, and A. Politis, Eds. Wiley, 2017, pp. 3–24.

[30] K Matrouk, A Al-Hasanat, H Alasha'ary, Ziad Al-Qadi, H Al-Shalabi, Speech fingerprint to identify isolated word person, World Applied Sciences Journal, 31 (10), 1767-1771, 2014.

[31] Ziad alqadi, Analysis of stream cipher security algorithm, Journal of Information and Computing Science, vol. 2, issue 4, pp. 288-298, 2007.

[32] 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, vol. 3, issue 1, pp. 86-93, 2019.

[33] Musbah J Aqel, Ziad ALQadi, Ammar Ahmed Abdullah, RGB Color Image Encryption-Decryption Using Image Segmentation and Matrix Multiplication, International Journal of Engineering and Technology, vol. 7. Issue 3.13, pp. 104-107. 2018.

[34] Jihad Nadir, Ashraf Abu Ein, Ziad Alqadi, A Technique to Encrypt-decrypt Stereo Wave File, International Journal of Computer and Information Technology, vol. 5, issue 5, pp. 465-470, 2016.

[35] Saleh Khawatreh, Belal Ayyoub, Ashraf Abu-Ein, Ziad Alqadi, A Novel Methodology to Extract Voice Signal Features, International Journal of Computer Applications, vol. 975, pp. 8887, 2018.

[36] Majed O. Al-Dwairi, Amjad Y. Hendi, Mohamed S. Soliman, Ziad A.A. Alqadi, A new method for voice signal features creation, International Journal of Electrical and Computer Engineering (IJECE), vol. 9. Issue 9, pp. 4092-4098, 2019.

[37] Aws Al-Qaisi, Saleh A Khawatreh, Ahmad A Sharadqah, Ziad A Alqadi, Wave File Features Extraction Using Reduced LBP, International Journal of Electrical and Computer Engineering, vol. 8. Issue 5, pp. 2780-2787, 2018.

[38] Ayman Al-Rawashdeh, Ziad Al-Qadi, using wave equation to extract digital signal features, Engineering, Technology & Applied Science Research, vol. 8, issue 4, pp. 1356-1359, 2018.

[39] Ashraf Abu-Ein, Ziad AA Alqadi, Jihad Nader, A TECHNIQUE OF HIDING SECRETE TEXT IN WAVE FILE, International Journal of Computer Applications, 2016.

[40] Ismail Shayeb, Ziad Alqadi, Jihad Nader, Analysis of digital voice features extraction methods, International Journal of Educational Research and Development, vol. 1, issue 4, pp. 49-55, 2019.

[41] Jihad Nader Ahmad Sharadqh, Ziad Al-Qadi, Bilal Zahran, Experimental Investigation of Wave File Compression-Decompression, International Journal of Computer Science and Information Security, vol. 14m issue 10, pp. 774-780, 2016.

[42] Ziad A AlQadi Amjad Y Hindi, O Dwairi Majed, PROCEDURES FOR SPEECH RECOGNITION USING LPC AND ANN, International Journal of Engineering Technology Research & Management, vol. 4, issue 2, pp. 48-55, 2020.

[43] Majed O Al-Dwairi, A Hendi, Z AlQadi, an efficient and highly secure technique to encrypt-decrypt color images, Engineering, Technology &Applied Science Research, vol. 9, issue 3, pp. 4165-4168, 2019.

[44] Amjad Y Hendi, Majed O Dwairi, Ziad A Al-Qadi, Mohamed S Soliman, a novel simple and highly secure method for data encryption-decryption, International Journal of Communication Networks and Information Security, vol. 11, issue 1, pp, 232-238, 2019

[45] Prof. Ziad Alqadi, Dr. Mohammad S. Khrisat, Dr. Amjad Hindi, Dr. Majed Omar Dwairi, USING SPEECH SIGNAL HISTOGRAM TOCREATE SIGNAL FEATURES, International Journal of Engineering Technology Research & Management, vol. 4, issue 3, pp. 144-153, 2020.

[46] M. Abu-Faraj, Z. Alqadi, and K. Aldebei, "Comparative Analysis of Fingerprint Features Ex- Traction Methods," Journal of Hunan University Natural Sciences, vol. 48, iss. 12, pp. 177-182, 2021.

[47] M. Bala Kumara, P. Karthikkab, N. Dhivyac, T. Gopala Krishnan, A Performance Comparison of Encryption Algorithms for Digital Images, International Journal of Engineering Research & Technology (IJERT), Vol. 3 Issue 2, February – 2014.

[48] Dr. Amjad Hindi, Dr. Majed Omar Dwairi, Prof. Ziad Alqadi, Analysis of Procedures used to build an Optimal Fingerprint Recognition System, International Journal of Computer Science and Mobile Computing, vol. 9, issue 2, pp. 21 – 37, 2020.

[49] Aws AlQaisi, Mokhled AlTarawneh, Ziad A. Alqadi, Ahmad A. Sharadqah, Analysis of Color Image Features Extraction using Texture Methods, TELKOMNIKA, vol. 17, issue 3, pp. 1220-1225, 2019.

[50] Ziad AA Alqadi, Musbah Aqel, Ibrahiem MM El Emary, Multiple Skip Multiple Pattern Matching Algorithm (MSMPMA), IAENG International Journal of Computer Science, vol. 34, issue 2, 2007.

[51] ziad alqadi, Analysis of program methods used in optimizing matrix multiplication, journal of engineering, vol. 15, issue 1, 2005.

[52] Muaad Abu-Faraj, Abeer Al-Hyari, Khaled Aldebei, Ziad Alqadi, Bilal Al-Ahmad, Rotation Left Digits to Enhance the Security Level of Message Blocks Cryptography, IEEE Access, VOLUME 10, pp. 69388-69397, 2022.

[53] Prof. Ziad Alqadi, Improving Standard Methods of Message Cryptography, International Journal of Computer Science and Mobile Computing, vol. 11, issue 11, pp. 13-30, 2022.

[54] Mohamad T Barakat, Ziad A Alqadi, Securing Digital Image using Modified Chaotic Logistic Key, International Journal of Computer Science and Mobile Computing, vol. 11, issue 10, pp. 24-47, 2022.

