Contents lists available at GrowingScience

International Journal of Data and Network Science

homepage: www.GrowingScience.com/ijds

Increasing the security of transmitted text messages using chaotic key and image key cryptography

Mua'ad Abu-Faraj[a*], Abeer Al-Hyari[b], Ismail Altaharwa[a], Zaid Alqadi[c] and Basel J. A. Ali[d]

[a]*Department of Computer Information Systems, The University of Jordan, Aqaba 77110, Jordan*
[b]*Electrical Engineering Department, Al-Balqa Applied University, As Salt 19117, Jordan*
[c]*Computers and Networks Engineering Department, Al-Balqa Applied University, Amman 15008, Jordan*
[d]*Accounting and Finance Department, Applied Science University, Kingdom of Bahrain*

CHRONICLE

Article history:
Received: October 20, 2022
Received in revised format: October 28, 2022
Accepted: January 24, 2023
Available online: January 24, 2023

Keywords:
Cryptography
Image key
Chaotic logistic key
Quality
Sensitivity
Security
MSE
CC
Throughput

ABSTRACT

It is critical to safeguard confidential data, especially secret and private messages. This study introduces a novel data cryptography approach. The new approach will be capable of encrypting and decrypting any communication size. The suggested approach will use a sophisticated private key with a convoluted structure. The private key will have 5 components with a double data type to prevent guessing or hacking. The confidential data will produce two secret keys, the first of which will be taken from the image key. These keys will be vulnerable to slight changes in private key information. To maximize the approach's efficiency, the suggested method will deal with lengthy messages by splitting them into chunks. On the other hand, the chaotic logistic map model will be used to create the second key. The suggested technique will be implemented, and several sorts of analysis (sensitivity, quality, security, and speed analysis) will be undertaken to demonstrate the benefits of the proposed method. The quality metrics MSE, PSNR, and CC will be computed to validate the suggested method's quality. To illustrate the efficiency of the proposed technique, encryption and decryption times will be measured, and cryptography throughputs will be determined. Various PKs will be tried throughout the decryption process to demonstrate how sensitive the produced outputs are to changes in the private key. The suggested approach will be tested, and the results will be compared to the results of existing methods to demonstrate the improvement offered by the proposed method.

1. Introduction

Encryption is a cyber security mechanism that secures personal data by encrypting it using unique codes that obscure data and make it hard for hackers to read. Encryption guarantees the security of an organization's sensitive data, even if attackers get over the firewall. They are increasingly collecting a large amount of personal user data. To prevent this data from falling into the hands of unwanted third parties, companies must guarantee that any data in their control is encrypted. The data encryption procedure is simple. An encryption key is used with an encryption method to convert plain text data into unreadable data. The data that has been encrypted is known as cipher text (Fig. 1). Because encrypted data can only be decoded with the associated encryption key, hackers cannot access it even if they circumvent system security measures (Abu-Faraj, Al-Hyari, & Alqadi, 2022; Abu-Faraj & Alqadi, 2021b; Idbeaa, Abdul Samad, & Husain, 2016; Wang, Li, Xia, & Zheng, 2014).

* Corresponding author.
E-mail address: m.abufaraj@ju.edu.jo (M. Abu-Faraj)

ISSN 2561-8156 (Online) - ISSN 2561-8148 (Print)

doi: 10.5267/j.ijdns.2023.1.008

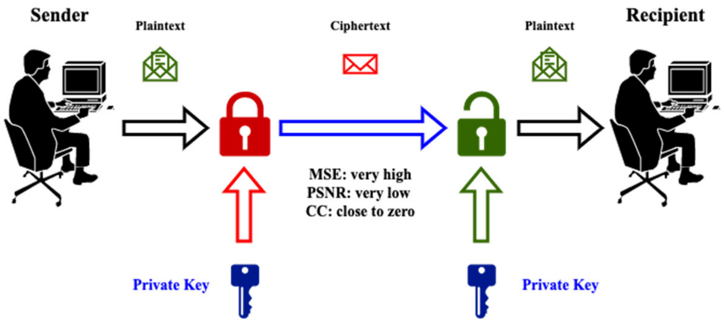


Fig. 1. Data cryptography process

For the following reasons (Al-Hyari, Aldebei, Alqadi, & Al-Ahmad, 2022; Idbeaa et al., 2016; Mua'ad, Aldebei, & Alqadi, 2022; Pavithra & Ramadevi, 2012; Wang et al., 2014; Xue, Zhou, & Zhou, 2020), businesses and people must make efforts to invest in data encryption for their online transactions and operations:

- The transmitted data is in jeopardy: Hackers can swiftly target communication links and intercept data sent across the network. Data encryption ensures that sensitive information cannot be stolen during transmission via data mining techniques such as brute-force assaults.
- Security threats are always developing. Attackers continually find new ways to circumvent the most sophisticated cyber security measures, so businesses must guarantee that attackers cannot access their data or breach their firewalls. Data encryption prevents intruders from accessing important information.
- Installing and utilizing illegal internet resources and apps may result in invasions and security breaches. When this occurs, your sensitive data may be exposed to malevolent agents online; data encryption guarantees that your private data is encrypted and difficult for unauthorized users to access.
- Hacking is a business: Because of competition and rivalry, businesses always attempt to obtain their rivals' data to determine their next commercial actions. As a result, the hacking business has expanded, with tech-savvy individuals compromising firms' web resources for a charge. Data encryption ensures that no one who has been granted access to your data can read, comprehend, or use your personal information, even if that person circumvents your security procedures.

The following is how data cryptography works:

First, the text to be encrypted is interpreted as a text message or an e-mail. Complex encryption techniques are applied to that text and turned into an unreadable format; the converted material is called "cipher text." This contributes to the security of digital data, whether it is kept on computer systems or sent via a network such as the Internet.

Second, when the encrypted content reaches the intended recipient, it is decrypted; the text is restored to what it was before encryption procedures were applied.

To unlock the communication, both the sender and the recipient must employ a "secret" encryption key, a set of algorithms that defends and decrypts the data into a readable format.

A decent data cryptography technique must meet the following requirements:

- High level of security: Provide high data protection by employing a complex private key (PK) that cannot be hacked.

- The method must destroy the original data by producing unreadable encrypted data; this can be measured by measuring the mean square error (MSE), peak signal-to-noise ratio (PSNR), and correlation coefficient (CC) between the original and encrypted data; the MSE value must be very high, the PSNR value must be very low, and the CC value must be close to zero.

- The method must recover the original data by producing decrypted data that is identical to the original data; this can be measured by mean square error (MSE), peak signal-to-noise ratio (PSNR), and correlation coefficient (CC) values measured between the original and encrypted data; MSE must be zero, PSNR must be infinite, and CC must be one.

MSE, PSNR, and CC can be calculated using equations 1, 2, and 3 (Abu-Faraj & Alqadi, 2021a; Al-Hyari, Aldebei, et al., 2022; Mua'ad et al., 2022; Usama & Zakaria, 2017; Xue et al., 2020; Zhang, Zhang, & Harn, 2019):

$$MSE = \frac{1}{N}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[S(i,j) - R(i,j)]^2, N = m \times n \tag{1}$$

$$cc = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}} \tag{2}$$

$$PSNR = 10 \times \log_{10}\frac{(MAX_I)^2}{MSE_t} \tag{3}$$

where cc is the correlation coefficient, x_i represents the first message's values, is the mean of x, y_i represents the second message's values, and y is the mean of y.

The digital color picture serves as a massive data incubator. Typically, it has a large number of pixels with values ranging from 0 to 255; this range corresponds to the range of ASCII letters (Abu-Faraj, Alqadi, Al-Ahmad, Aldebei, & Ali, 2022; Al-Hyari, Al-Taharwa, Al-Ahmad, & Alqadi, 2022; Kang, Jung, Lee, Kim, & Won, 2017; Khan, Masood, Alghafis, Amin, & Batool Naqvi, 2019; Khan & Waseem, 2018; Mua'ad & Zubi, 2020). Fig. 2, depicts the picture pixels grouped in a 3D matrix.

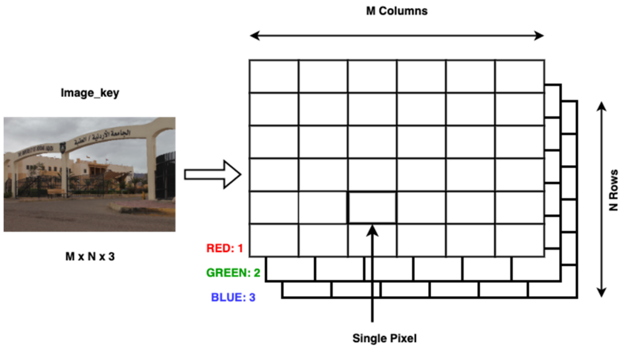


Fig. 2. Color image example

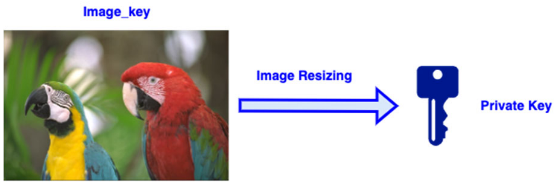


Fig. 3. Resize formation of an image for PK

The color image is a 3D matrix that can be resized to get any matrix with a selected dimension; this matrix can be used to form a secret key which can be easily used in the data cryptography process, as shown in Fig. 3. In this paper research, we will introduce a high-security method of message cryptography; the high level of data protection will be achieved by:

- Using a secret color image as an image_key.
- Extracting a private image key (IPK) from the image by applying image resizing, the size of IPK will depend on the message length or the message block length here; R points to the row number, C points to the column number, and BN points to the selected block number.
- Using the chaotic logistic map model to generate a chaotic private key (CPK) by setting the parameters of the chaotic map model (r and x are the chaotic parameters)
- IPK and CPK will form a complicated PK that cannot be hacked because it contains a complex structure, as shown in Figure 4; this structure will have five elements with a double data type, thus increasing the number of combinations, which will make the hacking process impossible.

Private Key (PK)				
Image Private Key (IPK)			Chaotic Private Key (CPK)	
R	C	BN	r	x

Fig. 4. PK structure

2. Related works

The goal of this paper research is to offer a way of message cryptography that improves on the efficiency of existing data encryption methods by accomplishing the following:

- Reducing encryption and decryption times while increasing message cryptographic throughput.

- Increasing security by employing a complex private key.

- Achieving high-quality encryption and decoding.

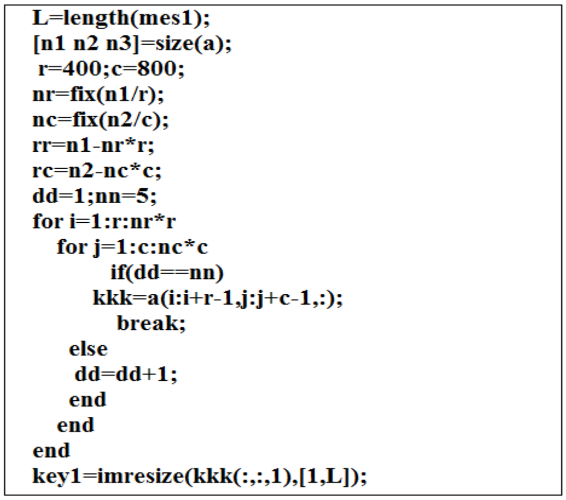
- The private key will be sensitive, which means that the decryption results will be dependent on the private key used; any changes in the utilized private key during the decryption phase will be deemed a hacking attempt, resulting in a damaged decrypted message. Many methods for data cryptography were established, some of which were based on standards such as data encryption standard (DES), advanced encryption standard (AES), 3DES, and blowfish (BF). These approaches have certain similarities, such as a fixed data block size, a defined length of PK, a fixed number of rounds, and the use of PK to produce additional required sub keys (Idbeaa et al., 2016). Many message cryptography approaches have been introduced. (Elmanfaloty & Abou-Bakr, 2020) proposed a technique based on the chaotic map model, and the throughput was increased to 0.1691 M bytes per second, but Vijayalakshmi et al. (2016) presented a method with a throughput of 0.71 M bytes per second. The authors of (Sohal & Sharma, 2022) developed a BDNA-A DNA-inspired symmetric key cryptography technology to secure cloud computing, performed performance comparisons, and demonstrated that the method outperformed existing methods (Abu-Faraj et al., 2022; Biryukov & De Cannière, 2011; Furht, 2008; Rachh, Mohan, & Anami, 2012; Thiyagarajan & Kamalakannan, 2014; J. Wang & Kissel, 2015), as shown in Table 1.

Table 1
Throughputs of various methods

Method	Throughput (K byte per second)
BDNA	180.4
BF	159.6
DES	33.32
AES	126.8
DNA	20.92

3. The proposed technique

To avoid hacking, the suggested approach employs a sophisticated PK; this key is composed of two parts: the first part is extracted from the selected picture block by identifying the values of R, C, and BN; and the second component is generated (see Fig. 4). Fig. 5 depicts the processes necessary to obtain this component of PK (IPK):



```
L=length(mes1);
[n1 n2 n3]=size(a);
 r=400;c=800;
nr=fix(n1/r);
nc=fix(n2/c);
rr=n1-nr*r;
rc=n2-nc*c;
dd=1;nn=5;
for i=1:r:nr*r
   for j=1:c:nc*c
          if(dd==nn)
        kkk=a(i:i+r-1,j:j+c-1,:);
           break;
      else
       dd=dd+1;
      end
   end
end
key1=imresize(kkk(:,:,1),[1,L]);
```

Fig. 5. Sequence of operations to get IPK

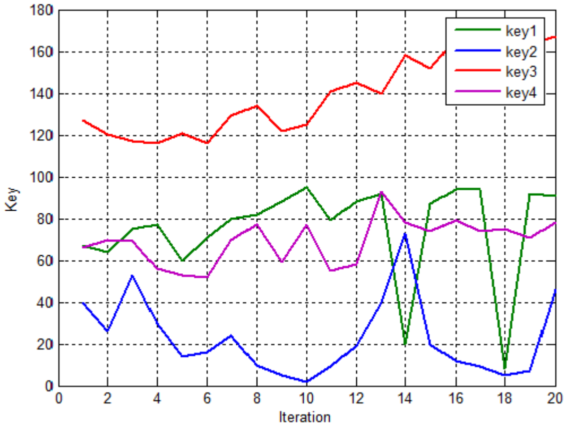


Fig. 6. Sensitivity of IPK

IPK is sensitive to selected image_key, selected R, selected C, picked BN, and message or message block size; any changes in one or more parameters will lead to a change in IPK; Fig. 6 shows an extracted IPKs for a message with a length equal to 20 characters using different images and fixing the values of R, C, and BN:

The chaotic logistic map key (CPK) is the second component of the PK; this key may be created by running the chaotic logistic map model and changing the variables of *L* (message length), *r* and *x* (Karule & Nagrale, 2016; Kharel, Busawon, & Ghassemlooy, 2011; Pavithra & Ramadevi, 2012; Vijayalakshmi et al., 2016; Wu, Li, & Kurths, 2015). The chaotic logistic map model will be used to construct the chaotic logistic map key (CMK). A chaotic logistic map is a one-dimensional function with the following key characteristics (Vijayalakshmi et al., 2016):

- The function generates deterministic values, and the likelihood of repetition relies on the length of the sequence.

- The generated numbers are quite sensitive to the original values used.

The chaotic logistic map (CLM) function is calculated by Eq. (4) (Biryukov & De Cannière, 2011; Karule & Nagrale, 2016; Rachh et al., 2012; Sohal & Sharma, 2022; Vijayalakshmi et al., 2016):

$$X_{n+1} = r \cdot X_n \cdot (X_n - 1) \tag{4}$$

where X_n is the first chaotic logistic parameter that ranges from 0 to 1, and r is the second chaotic logistic parameter from 0 to 4. The behavior of the chaotic logistic map depends on the selecting value of r, so when r is equal to (Karule & Nagrale, 2016; Rachh et al., 2012; Wu et al., 2015):

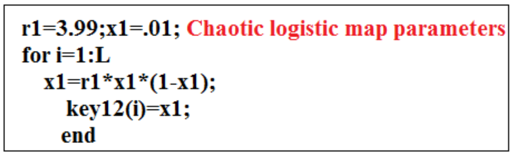
1. From 0 to 1, the chaotic function yields fixed and stable values close to zero.

2. From 1 to 3, the function yields static and unchanging values around (r-1)/r).

3. 3 to 3.7, the function creates values using the periodic attractor.

4. 3.7 to 4, the function behaves as a chaotic function.

CPK can be generated using the sequence of operations shown in Fig. 7. CPK is extremely sensitive to slight changes in the values of r and x; each change in this parameter results in a new CPK, as seen in Fig. 8.



```
r1=3.99;x1=.01; Chaotic logistic map parameters
for i=1:L
   x1=r1*x1*(1-x1);
      key12(i)=x1;
     end
```

Fig. 7. Sequence of operations to generate CPK

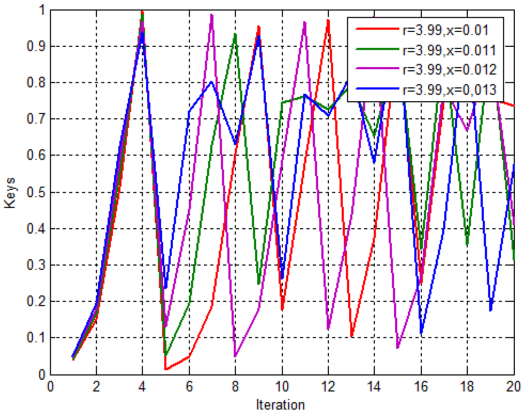


Fig. 8. Sensitivity of CPK

Generating a chaotic logistic map key requires processing time and memory space size; the requirements will increase when increasing the key size, as shown in Table 2:

Table 2
Chaotic logistic map key generation requirements

Key length	Processing time(second)	Memory space size(byte)
10	0.0000001	122
100	0.0010	932
200	0.0010	1832
400	0.0030	3632
500	0.0050	4532
1000	0.0140	9032
2000	0.0510	18032
3000	0.0580	27032
4000	0.0610	36032
5000	0.0670	45032
10000	0.1070	90032
100000	10.6570	900032

Increasing the chaotic key length will rapidly increase the required key generation time (Fig. 9). Thus, the encryption time will be increased.

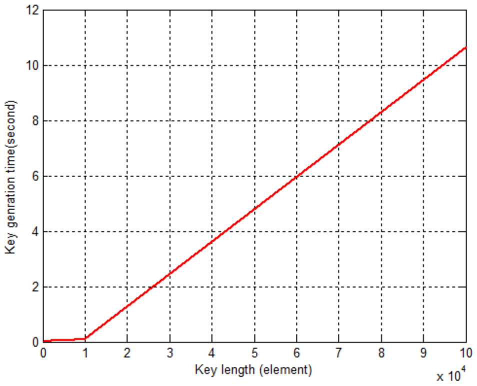
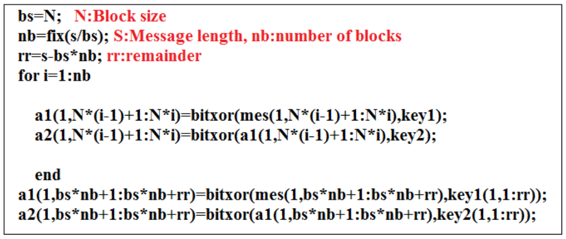


Fig. 9. Chaotic key generation time vs. key length



```
bs=N;   N:Block size
nb=fix(s/bs); S:Message length, nb:number of blocks
rr=s-bs*nb; rr:remainder
for i=1:nb

   a1(1,N*(i-1)+1:N*i)=bitxor(mes(1,N*(i-1)+1:N*i),key1);
   a2(1,N*(i-1)+1:N*i)=bitxor(a1(1,N*(i-1)+1:N*i),key2);

   end
a1(1,bs*nb+1:bs*nb+rr)=bitxor(mes(1,bs*nb+1:bs*nb+rr),key1(1,1:rr));
a2(1,bs*nb+1:bs*nb+rr)=bitxor(a1(1,bs*nb+1:bs*nb+rr),key2(1,1:rr));
```

Fig. 10. Message-blocking sequence of operations

Messages of large lengths must be broken into blocks to minimize the encryption-decryption time; each block can be encrypted-decrypted using a chaotic key of minimal size. Fig. 10 depicts the message-blocking sequence. As illustrated in Fig. 11, the proposed approach employs the PK information to produce two keys, image key (key 1) and chaotic key (key 2), for use in the encryption and decryption processes.

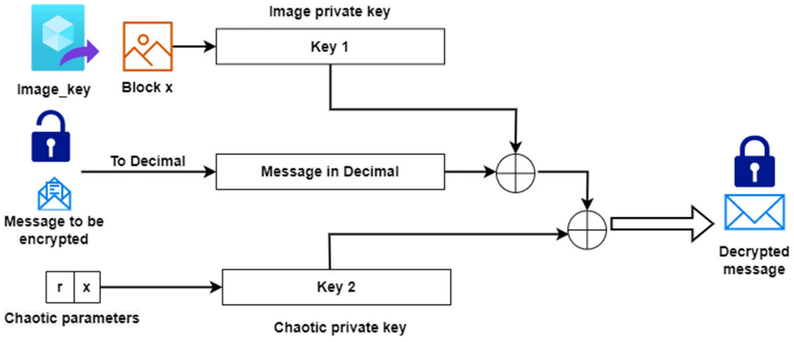
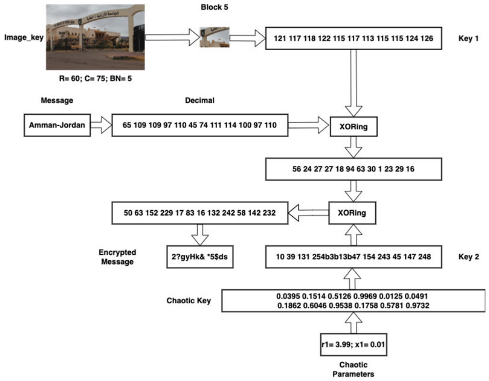


Fig. 11. The proposed approach



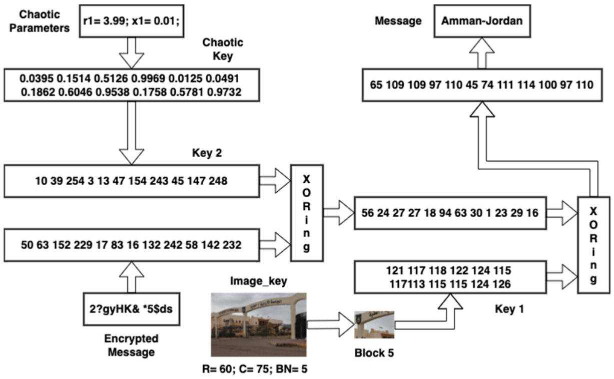


Fig. 12. Encryption phase example

Fig. 13. Decryption phase example

The following is the algorithm that describes the suggested method:

Encryption phase

Inputs:

Image_key, R, C, BN, L, r, x, message to be encrypted.

Output:

Encrypted message

Process

1. Get the message

```
            2. Get the PK
            3. Get the image key
            4. Use the information in PK to generate key 1 and key 2
            5. If the message length is greater than 400, apply message blocking
            6. Apply XORing the message (or message block) with key 1
            7. Apply XORing the results of step 6 with key 2 to get the encrypted message (message block)

    Decryption phase
     Inputs:
         Image_key, R, C, BN, L, r, x, encrypted message
     Output
         Decrypted message
     Process
            1. Get the encrypted message
            2. Get the PK
            3. Get the image key
            4. Use the information in PK to generate key 1 and key 2
            5. If the message length is greater than 400, apply message blocking
            6. Apply XORing the message (or message block) with key 2
            7. Apply XORing the results of step 6 with key 1 to get the decrypted message (message block)
```

4. Experimental results and implementation

Matlab was used to program the suggested approach, which was then performed using the images displayed. The essential information regarding these images is shown in Table 3.

Table 3
Selected images and basic information

Image number	Resolution(pixel)	Size(byte)	Image number	Resolution(pixel)	Size(byte)
H1	1880×2816	15882240	M1	151×333	150849
H2	1880×2816	15882240	M2	360×480	518400
H3	2848×4272	36499968	M3	1071×1600	5140800
H4	1880×2816	15882240	M4	981×1470	4326210
H5	1880×2816	15882240	M5	165×247	122265
H6	1880×2816	15882240	M6	360×480	518400
H7	1880×2816	15882240	M7	183×275	150975
H8	1880×2816	15882240	M8	183×275	150975
H9	1824×2736	14971392	M9	600×1050	1890000
H10	720×1280	2764800	M10	1144×1783	6119256

Several studies were carried out utilizing various images and messages, and the findings were examined using several forms of analysis, as detailed below:

4.1 Sensitivity analysis

As was shown earlier in this paper, the generated keys 9 key 1 and key 2) are extremely sensitive to even modest changes in the PK component, changing the image_key, or/and the message length (L), or/and changing R, or/and changing C, or/and changing BN will generate new values for key 1. Changing the chaotic parameters r and x will generate unique values for key 2.

4.2 Security Analysis

The suggested approach generates key 1 using a color picture; this image must be kept private and not transmitted. The produced keys 1 and 2 are determined by the PK component values; each component has a double data type that needs 64 bits; hence, the number of permutations is enormous, and equation 5 depicts the key space:

$$Ks = (2^{64})^5$$

The PK space is greater than 2100, so it can resist all brute force attacks.

4.3 Quality analysis

MSE, PSNR, and CC were determined for each run between the original message and the encrypted message, as well as between the original message and the decrypted message. The MSE, PSNR, and CC values between the original and encrypted images were always 0 and 1; this indicates that the message was retrieved during the decryption step. Several messages (short and long) were encrypted-decrypted using the suggested approach; Tables 4 and 5 show the computed MSE, PSNR, and CC values between the original and encrypted messages; image M3 was used as an image key.

Table 4
Cryptography for short messages (quality parameters)

Message length (byte)	MSE	PSNR	CC
10	10033	18.5318	0.0869
50	11692	17.1587	-0.0780
100	12947	15.9817	-0.0853
200	12371	16.5942	-0.1335
300	12567	16.4372	-0.1313
400	11950	16.9408	-0.0853
500	11973	16.9210	-0.0634
600	11349	17.4568	-0.0589
700	11048	17.7256	-0.0240
800	11236	17.5566	-0.0509
900	11832	17.0395	-0.0825
1000	11853	17.0215	-0.0771

Table 5
Long messages cryptography (quality parameters)

The length of the message (K byte)	MSE	PSNR	CC
1	12092	16.8224	-0.0749
10	11226	17.5652	-0.0357
50	11542	17.2879	-0.0583
100	11444	17.3731	-0.0535
200	11511	17.3144	-0.0571
300	11521	17.3057	-0.0576
400	11498	17.3257	-0.0569
500	11529	17.2990	-0.0583
1000	11498	17.3257	-0.0568

Table 4 and Table 5 illustrate that MSE values are always high and PSNR values are always low. The CC values, on the other hand, are always close to zero, indicating that the original messages were deleted, resulting in the encrypted messages. As a result, the suggested approach meets the quality criteria.

4.4 Speed analysis

We took several messages treated using the proposed method to do this analysis. The encryption and decryption times were measured, and Table 6 and Table 7 show the obtained results.

Table 6
Short messages encryption times (image_key M3)

Message length (byte)	Key1 time	Key 2 time	Encryption time	Total encryption time	Throughput
10	0.0480	0.0010000	Closed to 0	0.0490	204.0816
50	0.0470	0.0010000	Closed to 0	0.0480	1041.7
100	0.0480	0.0010	Closed to 0	0.0490	2040.8
200	0.0480	0.0020	Closed to 0	0.0500	4000.0
300	0.0470	0.0030	Closed to 0	0.0500	6000.0
400	0.0480	0.0040	Closed to 0	0.0520	7692.3
500	0.0500	0.0050	Closed to 0	0.0550	9090.9
600	0.0470	0.0070	Closed to 0	0.0540	11111
700	0.0470	0.0090	Closed to 0	0.0560	12500
800	0.0470	0.0110	Closed to 0	0.0580	13793
900	0.0480	0.0130	Closed to 0	0.0610	14754
1000	0.0640	0.0190	Closed to 0	0.0830	12048
Average				0.0554	7856.3

Table 7
Long messages encryption times (image_key H3)

Message length (K byte)	Key1 time	Key2 time	Encryption time	Total encryption time	Throughput
1	0.0340	0.0015	0.0000059000	0.0355	28820
10	0.0391	0.0384	0.000027100	0.0776	132040
50	0.0695	2.4455	0.000069700	2.5150	20357
100	0.0933	10.9888	0.00011770	11.0822	9240.0
200	0.2210	61.2343	0.00024290	61.4555	3332.5
300	0.2188	152.3863	0.00043450	152.6056	2013.0
400	0.3142	277.1126	0.00057730	277.4274	1476.4
500	0.4010	446.8925	0.00077310	447.2942	1144.7
1000	0.8923	1824.2	0.0028	1825.1	561.0804

Table 5 and Table 6 show that increasing the message size will rapidly increase the time required to generate the chaotic key (key 2), improving the total encryption time. Thus, the efficiency will drop by decreasing the method throughput, as shown in Fig. 14.

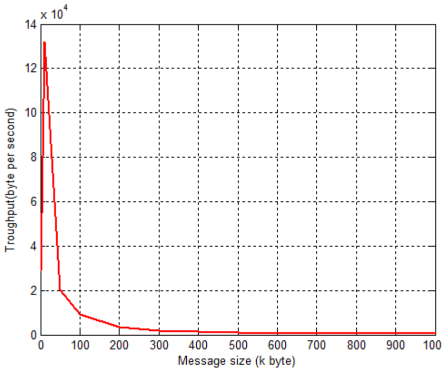


Fig. 14. Throughput drops when increasing message size

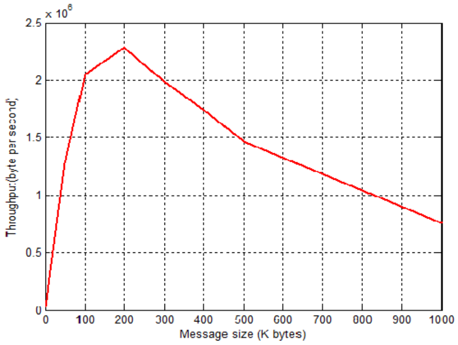


Fig. 15. Throughput vs. block size

Messages may be separated into blocks to address the preceding issue; each block must be encrypted and decrypted independently. Table 8 demonstrates how to increase method efficiency by separating messages into 400-byte pieces. We can see that for a message of size =1000 K bytes, it is preferable to choose a block size =1000 bytes; this allows us to twice the speed of cryptography. Fig. 15 illustrates the link between throughput and block size.

Table 8
Messages blocking results, block size = 400 byte

Message length (K byte)	Key1 time	Key2 time	Encryption time	Total encryption time	Throughput
1	0.0332	0.0020	0.000072400	0.0353	29041
10	0.0342	0.0020	0.000050560	0.0366	279580
50	0.0328	0.0020	0.0053	0.0402	1274900
100	0.0328	0.0020	0.0150	0.0499	2051800
200	0.0330	0.0019	0.0547	0.0896	2286600
300	0.0341	0.0021	0.1187	0.1549	1983500
400	0.0332	0.0020	0.2000	0.2352	1741700
500	0.0350	0.0020	0.3128	0.3498	1463800
1000	0.0346	0.0020	1.3213	1.3579	754130
1000 (block size = 1000)	0.0352	0.0028	0.6408	0.6787	1508800

The suggested approach was utilized to process a message of 1000 K bytes in length; several block sizes were employed. Table 9 displays the results:

Table 9
Message encryption results for 1000 Kbytes

Block size (byte)	Key1 time	Key2 time	Encryption time	Total encryption time	Throughput
100	0.0331	0.0018	5.2533	5.2882	193640
200	0.0328	0.0018	2.7544	2.7891	367140
300	0.0336	0.0019	1.6471	1.6827	608550
400	0.0338	0.0020	1.2236	1.2595	813050
500	0.0336	0.0021	0.9869	1.0226	1001400
600	0.0355	0.0023	0.8220	0.8598	1190900
800	0.0333	0.0025	0.6439	0.6798	1506400
900	0.0334	0.0025	0.5592	0.5951	1720800
1000	0.0336	0.0027	0.5218	0.5581	1834800
2000	0.0339	0.0048	0.2944	0.3331	3073900
3000	0.0368	0.0064	0.2715	0.3147	3253800
4000	0.0357	0.0091	0.1311	0.1759	5821900
10000	0.0372	0.0379	0.0583	0.1335	7673000
20000	0.0470	0.1426	0.0342	0.2238	4574900
50000	0.0636	1.9302	0.0198	2.0137	508520
100000	0.1010	10.9692	0.0134	11.0836	92389
Average 2139700 = 2089.6 K bytes per second					

Results of encrypting 1000 Kbytes message from Table 9 show that selecting a block size equal to 10000 will give the optimal speed (maximum throughput). The method throughput depends on the times required to generate key 1, the time to create key 2, and the encryption time; Figures (16 thru 20) show how these parameters are related to the block size.

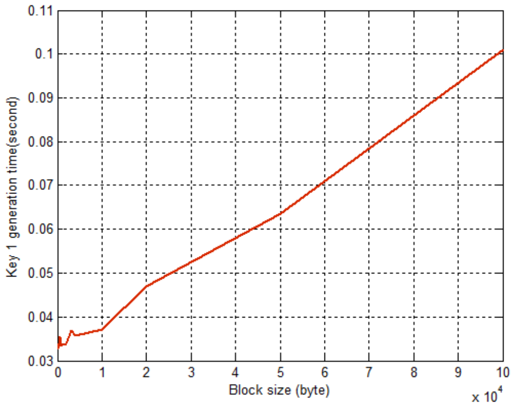


Fig. 16. Key 1 generation time vs. block size

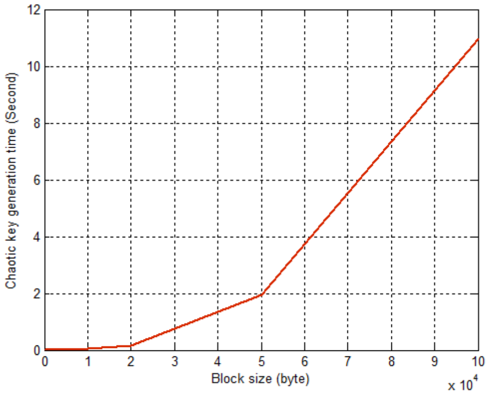


Fig. 17. Key 2 generation time vs. block size

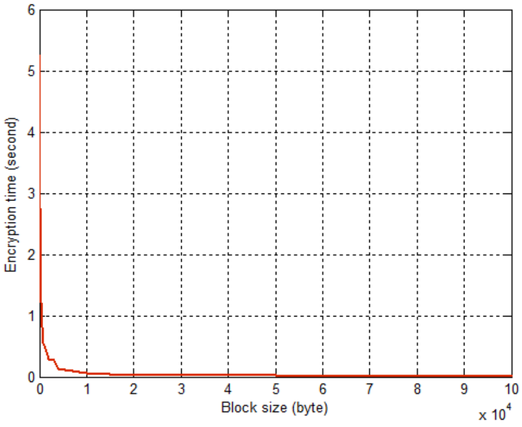


Fig. 18. Encryption time vs. block size

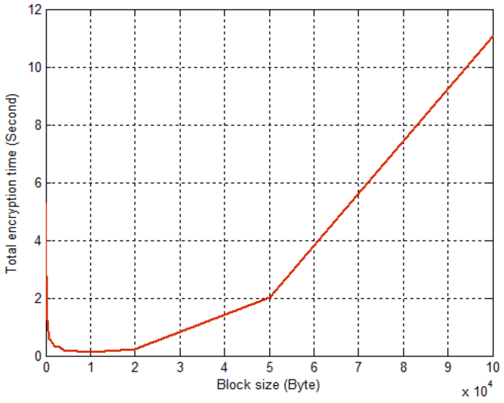


Fig. 19. Total encryption time vs. block size

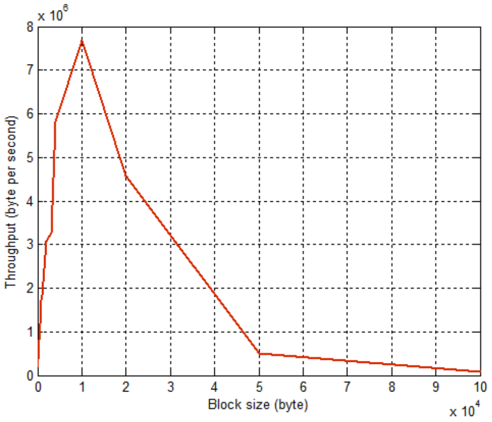


Fig. 20. Block size vs. throughput

Compared with other methods, the proposed method improved, as shown in Table 10.

Table 10
Methods of throughput comparison

Method	Throughput (Kbyte per second)
BDNA	180.4
BF	159.6
DES	33.32
AES	126.8
DNA	20.92
Proposed	2089.6

5. Conclusion

It is vital to protect sensitive data, particularly secret and private communications. This research presents a revolutionary data cryptography technique. The new method will be able to encrypt and decode any size communication. The proposed method will use a complex private key with a complicated structure. It presented an efficient and highly secure technique of message cryptography. The approach used a color picture as an image key, a complex PK, and a key structure that is strong enough and cannot be guessed or hacked. The image key key1 and the chaotic logistic map key2 were generated using the 5 items in PK that had double data types. These keys were very sensitive to tiny changes in the PK components; altering one or more components results in a new key, and employing these keys during the decryption phase results in a destructive message. The suggested approach was implemented using numerous messages and images, and the results were assessed using a variety of methods. During the encryption and decryption stages, the suggested technique produced outstanding MSE, PSNR, and CC values. Based on the MSE, PSNR, and CC values acquired during the encryption and decryption phases, the suggested technique matched the quality criteria by deleting the message during the encryption phase and recovering a message similar to the source one during the decryption phase. The acquired encryption and decryption times, as well as the predicted throughputs, demonstrated that the proposed technique is extremely efficient and outperforms other known message cryptography methods. Short and long messages were employed, and it was suggested that large messages be divided into blocks to improve method throughput. The suggested technique's throughputs were compared to other current data cryptography methods, and it was discovered that the proposed approach improved by greatly boosting the cryptography process's throughput.

References

Abu-Faraj, Alqadi, Z., Al-Ahmad, B., Aldebei, K., & Ali, B. (2022). A Novel Approach to Extract Color Image Features Using Image Thinning. *Applied Mathematics, 16*(5), 665-672.

Abu-Faraj, M. a., Al-Hyari, A., & Alqadi, Z. (2022). A Complex Matrix Private Key to Enhance the Security Level of Image Cryptography. *Symmetry, 14*(4), 664.

Abu-Faraj, M. a. M., & Alqadi, Z. A. (2021a). Improving the Efficiency and Scalability of Standard Methods for Data Cryptography. *International Journal of Computer Science & Network Security, 21*(12spc), 451-458.

Abu-Faraj, M. a. M., & Alqadi, Z. A. (2021b). Rounds Reduction and Blocks Controlling to Enhance the Performance of Standard Method of Data Cryptography. *International Journal of Computer Science & Network Security, 21*(12spc), 648-656.

Al-Hyari, A., Al-Taharwa, I., Al-Ahmad, B., & Alqadi, Z. (2022). CASDC: A Cryptographically Secure Data System Based on Two Private Key Images. *IEEE Access, 10*, 126304-126314.

Al-Hyari, A., Aldebei, K., Alqadi, Z. A., & Al-Ahmad, B. (2022). Rotation left digits to enhance the security level of message blocks cryptography. *IEEE Access, 10*, 69388-69397.

Biryukov, A., & De Cannière, C. (2011). Data encryption standard (DES). *Encyclopedia of Cryptography and Security*, 295-301.

Elmanfaloty, R. A., & Abou-Bakr, E. (2020). An image encryption scheme using a 1D chaotic double section skew tent map. *Complexity, 2020*.

Fang, D., & Sun, S. (2020). A new secure image encryption algorithm based on a 5D hyperchaotic map. *PloS one, 15*(11), e0242110.

Furht, B. (2008). *Encyclopedia of multimedia*: Springer Science & Business Media.

Idbeaa, T., Abdul Samad, S., & Husain, H. (2016). A secure and robust compressed domain video steganography for intra- and inter-frames using embedding-based byte differencing (EBBD) scheme. *PloS one, 11*(3), e0150732.

Kang, D., Jung, J., Lee, D., Kim, H., & Won, D. (2017). Security analysis and enhanced user authentication in proxy mobile IPv6 networks. *PloS one, 12*(7), e0181031.

Karule, K. P., & Nagrale, N. V. (2016). Comparative analysis of encryption algorithms for various types of data files for data security. *International Journal of Scientific Engineering and Applied Science, 2*(2), 495-498.

Khan, M., Masood, F., Alghafis, A., Amin, M., & Batool Naqvi, S. I. (2019). A novel image encryption technique using hybrid method of discrete dynamical chaotic maps and Brownian motion. *PloS one, 14*(12), e0225031.

Khan, M., & Waseem, H. M. (2018). A novel image encryption scheme based on quantum dynamical spinning and rotations. *PloS one, 13*(11), e0206460.

Kharel, R., Busawon, K., & Ghassemlooy, Z. (2011). Modified chaotic shift keying using indirect coupled chaotic synchronization for secure digital communication *Chaos Theory: Modeling, Simulation and Applications* (pp. 207-214): World Scientific.

Mua'ad, M., & Zubi, M. (2020). Analysis and implementation of kidney stones detection by applying segmentation techniques on computerized tomography scans. *Italian Journal and Applied Mathematics*(43-2020), 590-602.

Mua’ad, M., Aldebei, K., & Alqadi, Z. A. (2022). Simple, Efficient, Highly Secure, and Multiple Purposed Method on Data Cryptography. *Traitement du Signal, 39*(1), 173-178.

Pavithra, S., & Ramadevi, E. (2012). Study and performance analysis of cryptography algorithms. *International Journal of Advanced Research in Computer Engineering & Technology, 1*(5), 82-86.

Rachh, R. R., Mohan, P., & Anami, B. S. (2012). Efficient implementations for AES encryption and decryption. *Circuits, Systems, and Signal Processing, 31*(5), 1765-1785.

Sohal, M., & Sharma, S. (2022). BDNA-A DNA inspired symmetric key cryptographic technique to secure cloud computing. *Journal of King Saud University-Computer and Information Sciences, 34*(1), 1417-1425.

Thiyagarajan, B., & Kamalakannan, R. (2014). *Data integrity and security in cloud environment using AES algorithm.* Paper presented at the International Conference on Information Communication and Embedded Systems (ICICES2014).

Usama, M., & Zakaria, N. (2017). Chaos-based simultaneous compression and encryption for Hadoop. *PloS one, 12*(1), e0168207.

Vijayalakshmi, C., Lavanya, L., & Navya, C. (2016). A Hybrid Encryption Algorithm Based On AES and RSA. *International Journal of Innovative Research in Computer and Communication Engineering, 4*(1), 909-917.

Wang, C., Li, Y., Xia, X., & Zheng, K. (2014). An efficient and provable secure revocable identity-based encryption scheme. *PloS one, 9*(9), e106925.

Wang, J., & Kissel, Z. A. (2015). *Introduction to network security: theory and practice*: John Wiley & Sons.

Wu, X., Li, Y., & Kurths, J. (2015). A new color image encryption scheme using CML and a fractional-order chaotic system. *PloS one, 10*(3), e0119660.

Xue, X., Zhou, D., & Zhou, C. (2020). New insights into the existing image encryption algorithms based on DNA coding. *PloS one, 15*(10), e0241184.

Zhang, M., Zhang, S., & Harn, L. (2019). An efficient and adaptive data-hiding scheme based on secure random matrix. *PloS one, 14*(10), e0222892.