International Journal of Electrical and Computer Engineering (IJECE)
Vol. 12, No. 1, February 2022, pp. 478~488
ISSN: 2088-8708, DOI: 10.11591/ijece.v12i1.pp478-488

Multi-round encryption for COVID-19 data using the DNA key

Bassam Al-Shargabi, Mohammed Abbas Fadhil Al-Husainy
Faculty of Information Technology, Middle East University, Amma, Jordan

Article Info

Article history:

Received Sep 17, 2020
Revised Jun 23, 2021
Accepted Jul 9, 2021

Keywords:

COVID 19
Data security
DNA
Encryption algorithm
Privacy

ABSTRACT

The need for a reliable and fast encryption algorithm to encrypt medical data for patients is an extremely important topic to be considered especially during pandemic times such as the pandemic COVID-19. This pandemic forced governments and healthcare institutions to monitor COVID-19 patients. All the patient's data or records are also shared among healthcare researchers to be used to help them find vaccines or cures for this pandemic. Therefore, protecting such data (images, text) or records face an ever-increasing number of risks. In this paper, a novel multi-round encryption algorithm based on deoxyribonucleic acid (DNA) is proposed. The significance of the proposed algorithm comes from using a different random key to perform simple and fast encryption operations on multiple rounds to achieve a high level of confusion and diffusion effects in encrypted data. Experiments were conducted using a set of datasets of various types such as Excel sheets, images, and database tables. The experiments were conducted to test the performance and security level of the proposed encryption algorithm against well-known algorithms such as data encryption standard (DES) and advanced encryption standard (AES). The experiments show an outstanding performance regarding the encryption time, key size, information entropy, and the avalanche effects.

Corresponding Author:

Bassam Al-Shargabi
Faculty of Information Technology, Middle East University
Air-port Street, Amman, Jordan
Email: bshargabi@meu.edu.jo

1. INTRODUCTION

The health records of patients are put away in local databases or the Cloud. From one perspective such data can be advantageously obtained from the internet from any place at any time by physicians, insurance specialists, healthcare institutes, and researchers particularly during pandemics such as COVID-19. The whole globe is forced to lock-down as a consequence of the COVID-19 pandemic. During such a crisis, health care institutes and researchers are working to find a cure or vaccines through analyzing and exploring the records of COVID-19 patients. The patient record collection, analysis, processing, and up-to-date sharing of such valuable data records are an essential part for researchers to collaborate to find vaccines or a cure for this pandemic.

In addition to the health care crisis, the patient records that include personal and medical data become appealing and can become the focus for intruders. Thus, protecting such data or records face an ever-increasing number of risks. For example, the attacker might use data mining methods to profile patients based on their healthcare records. Additionally, such records stored in local data storage or the cloud can be exploited by system administrators to bargain or sell the patient's data. With these threats, there are undeniably growing open concerns about the privacy and security of patients' records [1], [2].

Furthermore, governments and healthcare institutions use mobile applications to track COVID-19 patients as a means to help them control the spread of this pandemic. This kind of application and its data are profoundly significant. There must be a law to be enforced to ensure and protect the privacy of patients, banking subtleties, shopping records, and so forth. The governments or a trusted third party as in [3], they forced to exploit encryption models to force data security and privacy laws on COVID-19 tracking applications to guarantee the privacy and safety of the society during this outrageous pandemic [4].

For the protection of the patient's sensitive data and records, the timely sharing of this data among physicians and healthcare institutions of many encryption methods have been deployed in the literature to secure transmission and sharing of this sensitive data in a secure manner while preserving the privacy of the patients. Regardless of the fact that data encryption or decryption systems are the traditional way to protect and secure transmission and sharing of data from hackers for a large number of years. At the moment encryption methods are the most broadly utilized procedures to protect sensitive data transmission. The symmetric encryption methods are extensively used [5]-[10], where only one key is utilized for encryption and decryption of data such as Hill Cipher, data encryption standards (DES), 3DES, and advanced encryption standard (AES). Other encryption and decryption methods were deployed where at least two distinctive yet related keys are utilized; one for encryption and another for decryption. These methods are commonly called public-key systems. The Symmetric methods are relatively faster and increasingly difficult to break, yet the key distribution is a major concern in such methods. Although the Rivest–Shamir–Adleman (RSA) is one of the well-known public-key systems it is not effective for encryption of huge amounts of medical data but it is increasingly advantageous to managing key [5]. On the other hand, the asymmetric methods are slower and less secure, yet proved its advantageous for key distribution issues [5], [11].

The main objective of this paper is to propose an encryption algorithm for COVID-19 patients' records or any medical records to be stored and transmitted securely while preserving the privacy of the patient's data. The proposed encryption algorithm has been developed using a deoxyribonucleic acid (DNA) random sequence and performs a set of fast encryption operations in several rounds using different keys generated for each round. The key generation in the proposed algorithm is randomly selected based on DNA and mapping table to ensure a high level of confusion and diffusion in the proposed algorithm.

The rest of the paper is organized as follows: section 2 examines the most recent related work. The proposed encryption algorithm is presented in section 3. Experimental results and evaluation of the proposed model presented in section 4. Finally, the conclusion about this paper is drawn in section 5.

2. RELATED WORK

Several methods have been introduced in the literature to tackle the issue of creating strong and fast encryption methods for medical data such as COVID-19 patients' data where data can be textual, images, and relational data. In this section, we present and discuss some of the most related work: A hybrid encryption model was proposed in [10], where they proposed a P-AES encryption algorithm for medical data records that are stored in the Cloud. Authors implied that most medical healthcare institutes hold their data in the Cloud about their patients. Medical data storage in the Cloud allows real-time sharing of the data among physicians and researchers. To overcome the security problem of storing medical records in the Cloud, the authors proposed the P-AES algorithm. The P-AES algorithm is a combination of improved AES and RSA to ensure the secrecy of data. They used an improved AES to encrypt and decrypt data while the key distribution was handled using the RSA. Unfortunately, the P-AES algorithm can only be applied to textual data and it is not suitable for images such as X-RAY and CT for COVID-19 patients.

Attribute-based encryption algorithms to secure sharing of medical data in [12]. The algorithm is based on using public-key encryption for physicians to encrypt patient data and message communication between the Cloud and physicians. Many attributes were used in their encryption algorithm to differentiate who can access medical data and Cloud servers where data resides to ensure security while preserving the privacy of patient medical data as in [13].

To protect the confidentiality of patients' medical data an approach based on using steganography was proposed in [14], to ensure that such is not processed or altered by unauthorized users. Their approach is based on using double based pixels allocation and three Bit Invert System before encrypting medical data to improve security level. The authors also applied affine cipher to encrypt medical data and utilized Huffman coding to reduce the data encryption before the embedding process to increases payload ability.

An investigation of using homomorphic encryption to preserve the privacy of genomic data in [1]. The authors used cryptographic keys to encrypt genomic variants data to apply it in the i2b2 framework to enable researchers to use such data. The authors claimed that the use of homomorphic encryption not only ensures the confidentiality but also preserved the privacy when quires are genomic data at i2b2 but such i2b2 framework is accessed through a local network.

Gabetta *et al.* [15], the authors implemented a new encryption algorithm that is based on the use of four block ciphers to encrypt data. They used binary tree traversal for multi-bit word substitution and 2D array for the diffusion process. Such an approach requires a larger memory size and time-consuming regarding encryption time compared to AES. A similar approach in [16] was introduced to improve the security of Amazigh text. The approach implemented the elliptic curve cryptography (ECC) and the technique of binary tree traversal. The use of ECC still suffers from binary curves.

An auto-generated key was introduced to encrypt images [17]. The method is based on a block cipher that exploited any digital file as a seed for a secret key generation. A block size of 32 was used with variable key size but yet the size of the key is breakable because of its small size. The digital file was used to create a substitution table to encrypt the image. Moreover, they embedded part of the encryption key within the encrypted image. Unlike our proposed encryption algorithm, we used multi-round encryption for the image where the key is generated for each round based on DNA and Map-table for the substitution and transposition processes.

A multi-level DNA based encryption algorithm was introduced in [18], where they utilized DNA sequence or tape to generate the key based on the sensitivity of encrypted data. A random DNA selection process is based on blum blum shub (BBS). The BBS-DNA tape was exploited to produce new DNA tape to be used in substitution and transposition processes along with embedding DNA tape used in the encryption process to the encrypted data using the Hadamard matrix.

Another DNA based encryption algorithm for text data proposed in [19], where the key is randomly selected from the mapping DNA table. The text data is converted to American standard code for information interchange (ASCII) representation and then the ASCII representation is converted into the original binary code. The DNA table is used to code its match in the binary representation of text data to be encrypted using simple substitution rules but the key used in this method is only 32 bits and because of the small size it is easy to break.

An encryption method based on DNA for the memory limited devices to encrypt images is introduced in [20]. The method used the least signification bit (LSB) to embed patient information in images such as CT images. The image is then encrypted based on a set of rules formed based on DNA tape along with the use of a multi chaotic map. Other methods utilized the cloud and DNA sequence for creating strong encryption methods for protecting the security of data transmission [21]-[23]. Most of the abovementioned related work uses the DNA only to generate the key but in the proposed algorithm in this paper, we used DNA tape a mapping table to ensure a high level of confusion and diffusion in the proposed algorithm. In addition, most of the mentioned related work uses a single round but multi-round encryption with a different encryption key for each round will enhance security against any possible attacks.

3. THE POROPOSED ENCRYPTION ALGORITHM

It has been used and recommended by most well-known cryptosystems such as DES, 3DES, AES, and others [5], [24], [25]. To achieve a high level of protection for confidential data, it is necessary to take into account a set of factors:

- Use the largest size possible for the used key
- The key used should be as random as possible
- Key transmission is easy between users
- Encryption operations must be performed on data for several rounds
- Different keys must be used in each round
- Conducting the above factors should keep the encryption time at an acceptable level.

Based on these factors, a proposed encryption algorithm has been developed using a random DNA sequence and to perform a set of fast encryption operations in a number of rounds using different keys generated for each round. Figure 1 shows the general model of the proposed encryption algorithm. To begin with, here are some data structures that are defined and presented to make it easy to understand the operations performed in each step of the proposed algorithm:

- Source data (SD): a digital COVID-19 data file entered by a user to be encrypted. Initially, the source data is divided into 2-bit segments. Each 2-bit represents one of the four decimal numbers as shown in Table 1.
- Source data length (SDLength): refers to the length of the source data after converting the source data to the corresponding sequence of numbers (0, 1, 2, and 3). The SDLength is calculated using (1).

$$\text{SDLength} = (\text{Source Data Length (in bytes)}) \times 4 \quad (1)$$

- DNA sequence (DNA): refers to the DNA sequence used by the algorithm. The DNA sequence consists of a random sequence of four letters (A: Adenine, T: Thymine, C: Cytosine, and G: Guanine). The DNA letters are treated as a range of four decimal numbers (0, 1, 2, and 3) as shown in Figure 2(a). The numerical representation of the DNA sequence is depicted in Figure 2(b). DNA sequence is treated by the algorithm as random sequence of four decimal numbers (0, 1, 2, and 3) as shown in Figure 2(b).
- DNA length (DNALength): refers to the length (in letter) of the DNA sequence.
- Number of round (NRound): represents the number of rounds in which the encryption operations are performed in the algorithm.
- Substitution index (SIndex): refers to the index of the part of the DNA sequence that is used during the substitution operation in the algorithm.
- Transposition index (TIndex): refers to the index of the part of the DNA sequence that is used during the transposition operation in the algorithm.

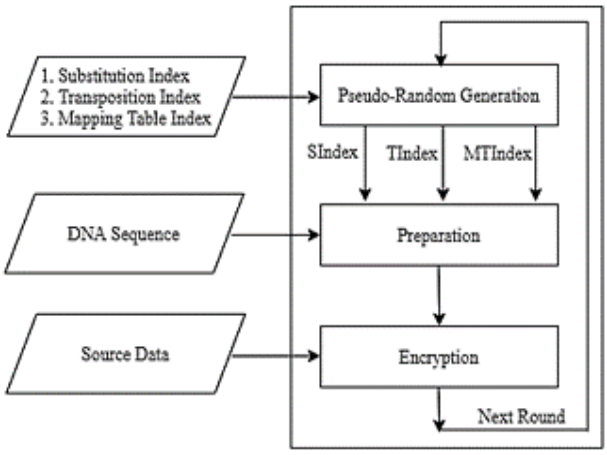


Figure 1. General model of the proposed image encryption algorithm

Table 1. 2-bit binary numbers and the corresponding decimal representation

Binary	Decimal
00	0
01	1
10	2
11	3

DNA Letter	Numerical Representation
A	0
T	1
C	2
G	3

(a)

DNA sequence

A	C	G	T	A	T	…

Numerical representation of the DNA sequence

0	2	3	1	0	1	…

(b)

Figure 2. The DNA letters are treated as a range of four decimal numbers as in (a) mapping table of DNA letters and the corresponding numbers, (b) DNA sequence and its numerical representation

- Mapping table (MT): refers to a table that has two columns that link each of the DNA letters/numbers (0, 1, 2 and 3) to a non-repeated letter/number extracted from the DNA sequence. This table will be used during the transposition operation in the algorithm. Table 2 shows an example of the mapping table (MT).
- Mapping table index (MIndex): refers to the index of the part of the DNA sequence that is used to build the mapping table (MT) used during the transposition operation in the algorithm.
- Encrypted data (ED): a digital COVID-19 data file that is produced after completing the encryption operations on the source data. It is treated by the algorithm as a collection of bytes.

Table 2. An example of the mapping table (MT)

DNA Letter	Extracted Letters from DNA Sequence
A	C
T	T
C	G
G	A

As shown in Figure 1, the proposed encryption algorithm includes three main phases before encrypting the secret of the COVID-19 data. The user should transmit the key that will enter to the algorithm using a secret channel. This key consists of only three numbers (SIndex, TIndex and MIndex). These numbers represent three indices in a previously determined DNA file that is existed in one of the public DNA datasets.

3.1. Pseudo-random generation phase

To ensure that the values of SIndex, TIndex and MIndex used are different in each round of the algorithm, even the initial values entered by the user as input to the algorithm. A pseudo random generation algorithm has been adopted in the proposed algorithm that based on a Seed value calculated from the current SIndex, TIndex and MIndex to produce new random values. The Seed value is calculated using (2).

$$\text{Seed} = (\text{SIndex} + \text{TIndex} + \text{MIndex}) \quad (2)$$

The calculated Seed value from in (2) is used by the adopted random generation algorithm to generate new values for SIndex, TIndex and MIndex.

3.2. Preparation phase

Three preparation operations are performed to set the necessary parameters for the encryption phase, the three operations are:

- Set the DNA sequence to be used in the substitution operation. This sequence is starting from the SIndex index to the (SIndex+SDLength)-1. It will start from index 0 in the DNA sequence whenever the end of the DNA sequence is reached.
- Set the DNA sequence to be used in the transposition operation. This sequence is starting from the TIndex index to the (TIndex+SDLength)-1. It will start from index 0 in the DNA sequence whenever the end of the DNA sequence is reached.
- Build the mapping table (MT) by extracting a non-repeated letter/number from the DNA sequence starting from the index MIndex. The extracted letters/numbers are sequentially filled (from top to down) in the second column of MT. This yields to assign a random letter/number to each DNA letter/number. It will start from index 0 in the DNA sequence whenever the end of the DNA sequence is reached.

3.3. Encryption phase

Substitution and transposition are the two main operations performed in this phase. These operations help to achieve the necessary level of confusion and diffusion in the source data to produce encrypted data. The description of the two operations are as follows:

- Substitution operation: This operation is conducted by performing XOR logical operation between the 2-bit of each number in the SD with the corresponding 2-bit of the number in the DNA sequence starting from the index SIndex. Table 3 shows the truth table of the XOR logical operation. The confusion effect will occur in the SD as a result of this operation due to the changes made in SD bits. Every number in SD is changed many times by performing XOR operation with a different number in each round of the algorithm.

Table 3. Truth table of XOR logical operation

Bit X	Bit Y	Bit Z
0	0	0
0	1	1
1	0	1
1	1	0

- Transposition operation: The following pseudo-code explains how to implement this operation.
  a. First1=TIndex;
  b. Last1= (TIndex+SDLength)-1;

c. First2=0;
d. Last2= SDLength-1;
e. While (Start≠End)
f. Based on the MT table,
g. If (DNA[First1] ↔ DNA[Last1]) then
h. SWAP(SD[First2], SD[Last2])
i. First1= First1+1;
j. Last1= Last1-1;
k. First2= First2+1;
l. Last2= Last2-1;
m. EndWhile

The diffusion effect will occur in the SD as a result of this operation due to the changes in positions made in SD numbers. Every number in SD is transferred to different positions in SD in each round of the algorithm. The above three phases are repeated several times depending on the value of NRound. This will help to achieve more confusion and diffusion effects in the SD. The use of different random indices (SIndex, TIndex and MIndex) as a key in each round will add more difficulties in front of the attackers to guess all these numbers.

4. RESULTS AND DISCUSSION

Several COVID-19 datasets [26]-[29] were used in the experiments to test the performance of the proposed encryption algorithm. These datasets contain a large number of files of various types such as Excel sheets, images, and database table. Figure 3 shows examples of these files. The key size, encryption time, security level, information entropy and avalanche effect measurements are used to compare the quality of the proposed encryption algorithm with a set of well-known techniques such as DES and AES.

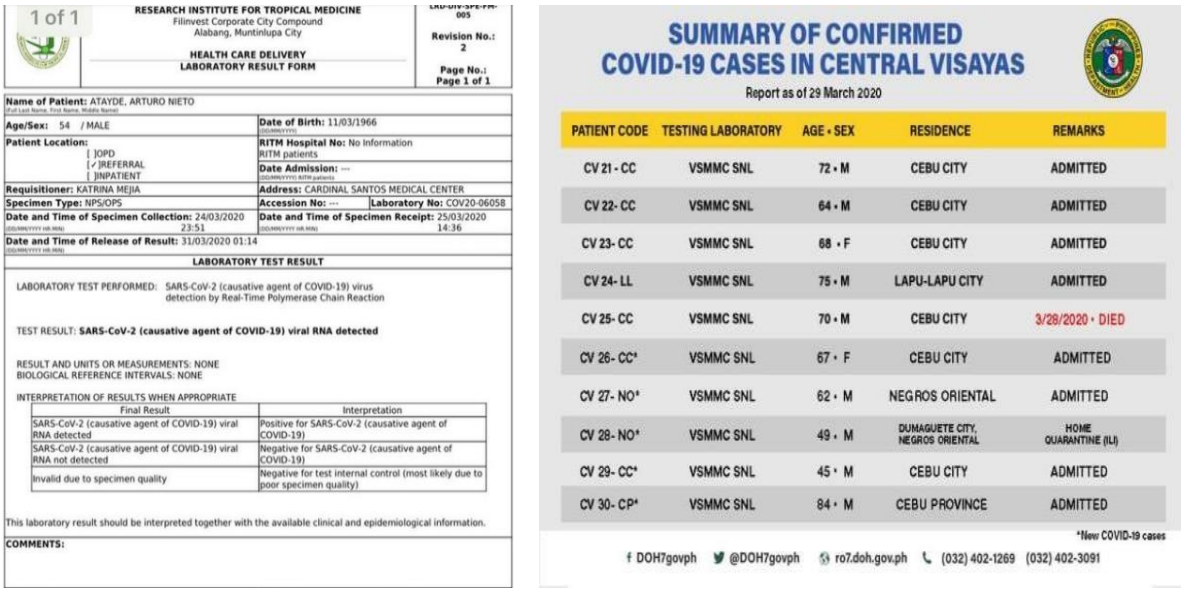


Figure 3. Examples of COVID-19 data files

4.1. Key size

The user's key that is entered into the proposed encryption algorithm consists of four integers (SIndex, TIndex, MIndex, and NRound) as mentioned before. Each integer represents 16 bits. This means that the total number of bits of the four integers is equal (16 bit×4=64 bits); because the algorithm is implemented for a number of rounds. Therefore, it uses different SIndex, TIndex and MIndex in each round.

In addition, the existence of thousands of DNA files available in global databases that can be previously determined by the user to use in the algorithm. Therefore, the total number of bits of the key can be estimated using (3). If we assume that the algorithm is implemented for only five rounds except the bits needed to represent the available number of DNA files in the global databases. This means that the calculated

Key Size from (3) is equal ((48 bits×5)+16 bits)=436 bits. Certainly, this is sufficient to make it difficult to break the key by attackers.

$$\text{Key Size}_{\text{bit}} = \left((48 \text{ bits} \times \text{NRound}) + 16 \text{ bits} + (\log(\text{Number of Available DNA})/\log(2))\right) \quad (3)$$

When comparing the size of the key used by the proposed algorithm with a set of well-known encryption techniques. It shows that the proposed encryption algorithm succeeded in using the largest possible size of the key. Table 4 presents the size of the key used by the proposed encryption algorithm and other well-known encryption techniques.

Table 4. Size of key used in the proposed algorithm and well-known encryption techniques [9], [19]

Encryption System	Size of the Key (bit)
Proposed algorithm	Much more than 64
Blowfish	32
AES	256
DES	56
3DES	168
Twofish	128

4.2. Encryption and decryption time

One of the main challenge points in developing a multi-round secure encryption method is keeping the encryption time at an acceptable rate. The development of the proposed encryption algorithm has taken into account the use of rapid operations, where the XOR logical operation is used instead of any other math operation that may take a variety of calculations. Also using the small MT mapping table helps reduce the time needed to implement the algorithm.

A large number of files of various types that are taken from COVID-19 datasets, for example (Excel sheets, images, and database tables) were encrypted during the experiments. These files have been encrypted using the proposed encryption algorithm and using the two well-known encryption techniques DES and AES. Tables 5 and 6 show the encryption and decryption times that were recorded in the experiments (using different types of files of different sizes) to achieve the same confusion and diffusion rate in the encrypted data by using the proposed algorithm and the two methods DES and AES. The encryption and decryption times recorded by the proposed encryption algorithm are clearly competitive. This encourages the use of the proposed encryption algorithm in the area of information protection.

Table 5. The encryption time of the proposed algorithm and the two methods DES and AES

Data	Data Size (KB)	Encryption time (sec)		
		Proposed	DES	AES
Data 1	2.25	1.982	2.89	2.45
Data 2	1.199	1.231	2.41	1.917
Data 3	196	0.238	0.500	0.352
Data 4	201	0.368	0.327	0.501

Table 6. The decryption time of the proposed algorithm and the two methods DES and AES

Data	Data Size (KB)	Decryption time (sec)		
		Proposed	DES	AES
Data 1	2.25	1.871	2.80	2.30
Data 2	1.199	1.199	2.35	1.878
Data 3	196	0.201	0.458	0.332
Data 4	201	0.342	0.300	0.480

4.3. Security level

Two types of tests were used in the experiments to assess the quality of the proposed encryption algorithm regarding the level of protection achieved. A good encryption method is the one that can achieve the highest possible ration of confusion and diffusion effects in the encrypted data. These effects can be measured numerically by calculating the peak signal to noise ratio (PSNR) using (4) and (5) respectively. Table 7 shows the PSNR recorded in the experiments (using different types of files of different sizes) by implementing the proposed algorithm and the two methods DES and AES.

$$\text{NMAE} = \frac{\sum_{k=0}^{\text{SDLength}-1} |\text{SD}(k) - \text{ED}(k)|}{\text{SDLength}} \times 100 \quad (4)$$

$$\text{PSNR}_{\text{db}} = 10.\log_{10}\left(\frac{\text{Max}_{\text{SD}}^2}{\text{NMAE}}\right) \quad (5)$$

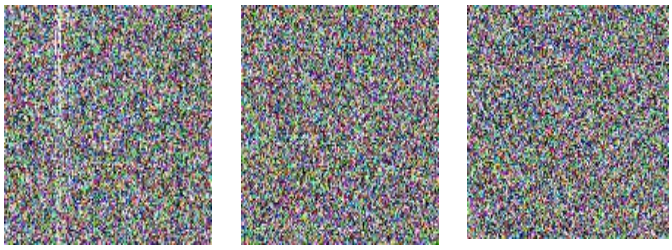
Where: *MaxSD* is the maximum possible byte value in the source data *SD* and db refers to a decibel.

Obviously, the PSNR recorded by the proposed encryption algorithm during the experiments is very close to other known encryption methods DES and AES. To clarify the progress of confusion and diffusion effects that are achieved in the encrypted data in each round, one image file chosen and implement the proposed algorithm for a number of rounds. The encrypted images produced in these rounds have been depicted in Figure 4 (a)-(c).

Also, the confusion and diffusion effects achieved can be statistically tested using the histogram of byte values in source data compared with byte values in encrypted data. A good encryption method is the one that achieves the highest rate of flatness in the encrypted data histogram. Figure 5 (a)-(d) show examples of the histograms of source and encrypted data generated using the proposed algorithm and the two methods DES and AES. The success of achieving a good flatness in the histogram of values of bytes by the proposed algorithm (similar to DES and AES methods) means that there is good confusion and diffusion effects are happening in the encrypted data.

Table 7. The PSNR values of the proposed algorithm and the two methods DES and AES

Data	Data Size (KB)	PSNR (db)		
		Proposed	DES	AES
Data 1	2.25	5.501	5.462	5.401
Data 2	1.199	5.625	5.412	5.403
Data 3	196	5.423	5.417	5.404
Data 4	201	6.533	6.546	6.543



(a) (b) (c)

Figure 4. The implementation of the proposed algorithm in a number of rounds: (a) one round, (b) three rounds, (c) ten rounds

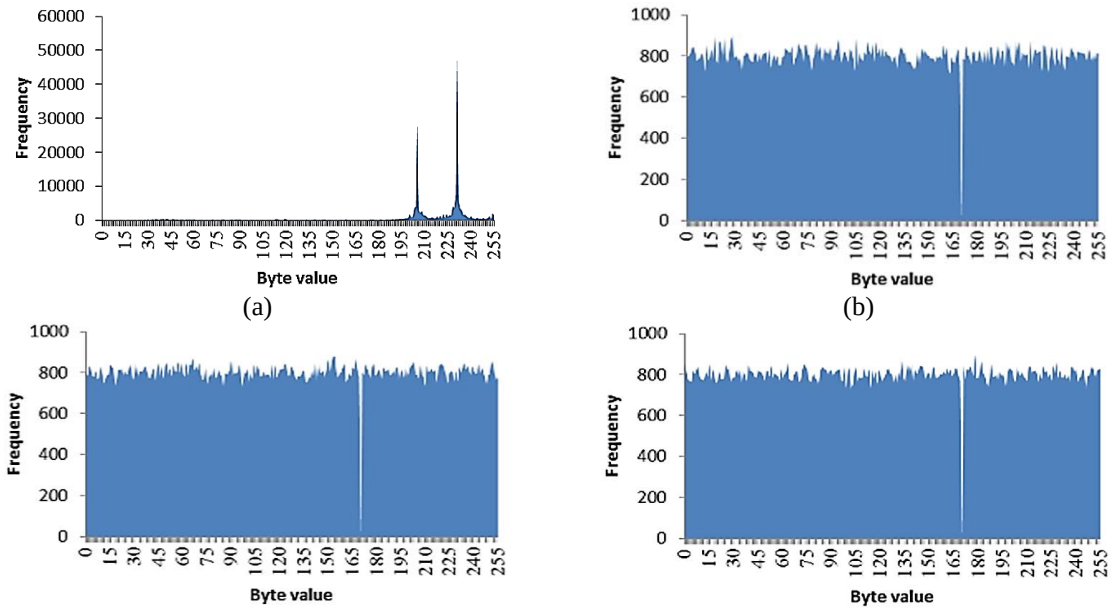


Figure 5. Histogram of (a) source data, (b) encrypted data using the proposed algorithm, (c) encrypted data using DES method, and (d) encrypted data using AES method

4.4. Information entropy

Information entropy is an essential feature of the randomness of an input image. Entropy is the average (expected) amount of information from the data [30], [31]. For a digital image, it is hard to predict the content if its information entropy is high. Here, the entropy is calculated by (6).

$$\text{Entropy} = -\sum_{i=1}^{n} P_i \,.\, \log_2(P_i) \quad (6)$$

where n is the number of different data values and P_i is the probability of occurring the data value.

The information entropy of the source and encrypted data are listed in Table 8. The entropy values in Table 8 indicate that the proposed encryption algorithm achieved a competitive level of randomness in the encrypted data compared with the other known encryption methods DES and AES. Furthermore, the information entropy value of the proposed algorithm is almost 8, which shows that it is complicated to carry out a successful attack.

Table 8. Entropy values of the source and encrypted data

Data	Data Size (KB)	Entropy			
		Source	Proposed	DES	AES
Data 1	2.25	4.171996	7.991457	7.999849	7.999841
Data 2	1.199	4.177464	7.998691	7.999015	7.998972
Data 3	196	4.182701	7.999013	7.999332	7.999311
Data 4	201	4.266782	7.998982	7.998990	7.998995

4.5. Avalanche effect

The avalanche effect test is a numeric metric used to check the sensitivity of the encryption method to any small changes in the parameters [32]. To develop a high-quality encryption method, it should take into consideration that when there is a slight change in input (either in the key or the source data), this should cause significant changes in the encrypted data. As shown in (7) is used to calculate the number of bits that will be changed in the encrypted data when a few bits changed in the encryption key.

Table 9 shows the calculated values of the avalanche effect using (7) for "Data 2". Table 10 shows the average value of the avalanche effect during the experiments on different data files using the proposed method compared with those values of the avalanche effect test for DES and AES encryption methods. The results in Tables 9 and 10 confirm that the proposed method achieved an acceptable average value of the avalanche effect compared to those of DES and AES algorithms.

$$\text{Aval. Effect} = \frac{Number\ of\ changed\ bits\ in\ key\ used}{Total\ number\ of\ bits\ in\ encrypted\ data} \quad (7)$$

Table 9. The recorded values of the avalanche effect for "Data 2"

Number of bits changed in the encryption key	Avalanche test value (%)
1	50.133
3	50.222
10	50.407

Table 10. The average of avalanche effect values for the proposed algorithm and the two methods DES and AES

Encryption Method	Average of the avalanche effect values (%)
Proposed	50.254
DES	50.236
AES	50.344

5. CONCLUSION

In this paper, we introduced a multi-round encryption algorithm based on DNA for COVID-19 medical data. The algorithm relies on the use of multi-round encryption, wherein each round, the key is randomly selected based on DNA and mapping table to ensure a high level of confusion and diffusion of the proposed algorithm. The proposed algorithm proved a superior performance regarding key size, time of encryption compared to DES and AES. The low PSNR value proved that the proposed algorithm achieves the

highest rate of flatness in the encrypted data histogram value and uniform histogram. Security analysis using information entropy and avalanche effects shows that the proposed algorithm has a high level of security and can endure all types of attacks compared with other algorithms.

ACKNOWLEDGEMENTS

The authors are grateful to the Middle East University, Amman, Jordan for the full financial support granted to this research project.

REFERENCES

[1] J. L. Raisaro *et al.*, "Protecting privacy and security of genomic data in i2b2 with homomorphic encryption and differential privacy," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 15, no. 5, pp. 1413-1426, 2018, doi: 10.1109/TCBB.2018.2854782.

[2] R. Becker, A. Thorogood, J. Ordish, and M. J. S. Beauvais, "COVID-19 research: Navigating the european general data protection regulation," *Journal of Medical Internet Research*, vol. 22, no. 8, pp. 1-14, 2020, Art. no. e19799, doi: 10.2196/19799.

[3] A. Berke, M. Bakker, P. Vepakomma, K. Larson, and A. Sandy Pentland, "Assessing disease exposure risk with location data: A proposal for cryptographic preservation of privacy," *arXiv preprint arXiv:2003.14412*, 2020.

[4] D. C. Nguyen, M. Ding, P. N. Pathirana, and A. Seneviratne, "Blockchain and ai-based solutions to combat coronavirus (COVID-19) -like epidemics : A survey," *IEEE Access*, vol. 9, pp. 95730-95753, 2020, doi: 10.1109/ACCESS.2021.3093633.

[5] A. Kumar and H. Sharma, "A survey on common symmetric encryption algorithms," *International Journal of Engineering and Technical Research (IJERT)*, vol. 3, no. 12, pp. 241-243, 2018.

[6] P. Hebbar, P. Hegde, S. Nayak, S. Kerni, and K. T. Rajgopal, "Study and performance evaluation of different symmetric key cryptography technique for encryption," *International Research Journal of Engineering and Technology (IRJET)*, vol. 6, no. 5, pp. 1151-1154, 2019.

[7] B. Al-Shargabi and O. Sabri, "Internet of things: An exploration study of opportunities and challenges," *2017 International Conference on Engineering & MIS (ICEMIS)*, 2017, pp. 1-4, doi: 10.1109/ICEMIS.2017.8273047.

[8] B. Al-Shargabi, S. Al-Jawarneh, and S. M. A. Hayajneh, "A cloudlet based security and trust model for e-government web services," *Journal of Theoretical and Applied Information Technology*, vol. 98, no. 1, pp. 27-37, 2020.

[9] M. A. F. Al-Husainy and H. A. A. Al-Sewadi, "Implementing binary search tree concept for image cryptography," *International Journal of Advanced Science and Technology (IJAST)*, vol. 130, pp. 21-32, 2019, doi: 10.33832/ijast.2019.130.03.

[10] F. Zhang, Y. Chen, W. Meng and Q. Wu, "Hybrid encryption algorithms for medical data storage security in cloud database," *International Journal of Database Management Systems (IJDMS)*, vol. 11, no. 1, pp. 412-416, 2017, doi: 10.5121/ijdms.2019.11104.

[11] A. Khurum and Z. Haider, "Securing data encryption," *engrXiv*, vol. 2019, 2019, doi: 10.31224/osf.io/ghbcu.

[12] A. M. Badr, Y. Zhang, and H. G. A. Umar, "Dual authentication-based encryption with a delegation system to protect medical data in cloud computing," *Electronics (Switzerland)*, vol. 8, no. 2, 2019, Art. no. 171, doi: 10.3390/electronics8020171.

[13] J. Han, W. Susilo, Y. Mu, and J. Yan, "Privacy-preserving decentralized key-policy attribute-based encryption," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 11, pp. 2150-2162, 2012, doi: 10.1109/TPDS.2012.50.

[14] M. M. Hashim, M. S. Taha, A. H. M. Aman, A. H. A. Hashim, M. S. M. Rahim, and S. Islam, "Securing medical data transmission systems based on integrating algorithm of encryption and steganography," *7th International Conference on Mechatronics Engineering ICOM*, 2019, pp. 1-6, doi: 10.1109/ICOM47790.2019.8952061.

[15] M. Gabetta, I. Limongelli, E. Rizzo, A. Riva, D. Segagni, and R. Bellazzi, "BigQ: a NoSQL based framework to handle genomic variants in i2b2," *BMC bioinformatics*, vol. 16, no. 1, 2015, Art. no. 415, doi: 10.1186/s12859-015-0861-0.

[16] F. Amounas, "A new encryption algorithm to increase security of amazigh text through tree traversal technique," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 5, no. 1, pp. 217-222, 2017, doi: doi:10.17762/ijritcc.v5i1.122.

[17] M. A. F. Al-Husainy, H. A. Al-Sewadi, and S. R. Masadeh, "Lightweight cryptosystem for image encryption using auto-generated key," *Journal of Engineering and Applied Sciences*, vol. 13, no. 17, pp. 7418-7425, 2018, doi: 10.3923/jeasci.2018.7418.7425.

[18] D. A. Zebari, H. Haron, S. R. M. Zeebaree, and D. Qader Zeebaree, "Multi-level of DNA encryption technique based on DNA arithmetic and biological operations," *ICOASE 2018 - International Conference on Advanced Science and Engineering*, 2018, pp. 312-317, doi: 10.1109/ICOASE.2018.8548824.

[19] B. R. Pushpa, "A new technique for data encryption using DNA sequence," *Proceedings of 2017 International Conference on Intelligent Computing and Control, I2C2 2017*, 2018, pp. 1-4, doi: 10.1109/I2C2.2017.8321834.

[20] R. I. Abdelfattah, H. Mohamed, and M. E. Nasr, "Secure image encryption scheme based on DNA and new multi chaotic map," *Journal of Physics: Conference Series*, vol. 1447, no. 1, 2020, Art. no. 012053, doi: 10.1088/1742-6596/1447/1/012053.

[21] G. Mogos, "Ciphertext-policy attribute-based encryption using quantum multilevel secret sharing scheme," *IAENG International Journal of Computer Science*, vol. 45, no. 4, pp. 500-504, 2018.

[22] M. Tagashira and T. Nakagawa, "Biometric authentication based on auscultated heart sounds in healthcare," *IAENG International Journal of Computer Science*, vol. 47, no. 3, 2020.

[23] M. Abbas, F. Al-husainy, I. Journal, M. A. F. Al-husainy, and B. Al-shargabi, "Secure and lightweight encryption model for IoT surveillance camera," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 2, pp. 1840-1847, 2020.

[24] P. Patil, P. Narayankar, D. G. Narayan, and S. M. Meena, "A comprehensive evaluation of cryptographic algorithms: DES, 3DES, AES, RSA and Blowfish," *Procedia Computer Science*, vol. 78, pp. 617-624, 2016, doi: 10.1016/j.procs.2016.02.108.

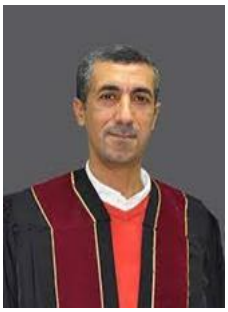
[25] M. F. Mushtaq, S. Jamel, A. H. Disina, Z. A. Pindar, N. S. A. Shakir, and M. M. Deris, "A survey on the cryptographic encryption algorithms," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 11, pp. 333-344, 2017, doi: 10.14569/IJACSA.2017.081141.

[26] Dimensions Resources, 2020, "Dimensions COVID-19 publications, datasets and clinical trials," *Dimensions*, doi: 10.6084/m9.figshare.11961063.v19.

[27] M. Talo, 2020, "COVID-19_X-Ray Image DataSet_Covid-19," Master muhammedtalo_COVID-19, GitHub.com. [Online]. Available: https://github.com/muhammedtalo/COVID-19

[28] IEEE8023, 2020, "ieee8023/covid-chestxray-dataset: We are building an open database of COVID-19 cases with chest X-ray or CT images," Github.com. [Online]. Available: https://github.com/ieee8023/covid-chestxray-dataset
[29] Data Hub, 2019, "Novel Coronavirus 2019-Dataset-DataHub-Frictionless Data," Datahub.io. [Online]. Available: https://datahub.io/core/covid-19
[30] H. Zhang, J. E. Fritts, and S. A. Goldman, "Entropy-based objective evaluation method for image segmentation," *Proceedings of the SPIE*, vol. 5307, pp. 38-49, 2003, doi: 10.1117/12.527167.
[31] W. Zhang, S. Wang, W. Han, H. Yu, and Z. Zhu, "An image encryption algorithm based on random hamiltonian path," *Entropy*, vol. 22, no. 1, 2020, Art. no. 73, doi: 10.3390/e22010073.
[32] M. Usman, I. Ahmed, M. I. Aslam, S. Khan, and U. A. Shah, "SIT: a lightweight encryption algorithm for secure internet of things," *arXiv preprint arXiv:1704.08688*, 2017.

BIOGRAPHIES OF AUTHORS



Bassam Al-Shargabi received his Ph.D. and M.Sc. in Computer Information Systems from the Arab Academy for Banking & Financial Sciences (Jordan) in 2009 and 2004, respectively. He received his BSc degree in Computer Science from Applied Science University (Jordan) in 2003. He has been an associate professor in Departments: Computer Information Systems, Faculty Information Technology, Isra University Amman-Jordan from 2014-2018. Currently, he is an Associate professor Department of Computer Information System, Faculty of Information Technology, Middle East University, Amman-Jordan. AL-Shargabi is an IEEE member. His current research interests are in Natural language processing, information retrieval, Data Security, and Service-Oriented architecture. He can be contacted at email: bassam20_152@yahoo.com.

Mohammed Abbas Fadhil Al-Husainy received the M.Sc. and Ph.D. degrees in 1996 and 2002, respectively. From 1997 to 2002, he was a lecturer in the Department of Computer Science, Al-Hadba University of Mosul. From 2002 to 2013. He has been an associate professor in Departments: Computer Science and Multimedia Systems, Faculty of Science and Information Technology, Al-Zaytoonah University of Jordan. Since 2014, he has been a professor in the Department of Computer Science, Faculty of Information Technology, Middle East University, Amman-Jordan. He taught many courses such as microprocessors, data structures, algorithms design and analysis, digital design systems, operating systems, cryptography, computer organization, programming languages. His research interests are in the broad field of algorithms design, including multi-media data processing, scheduling algorithms, information, system security, cryptography, and steganography algorithms. He can be contacted at email: m.abbas@gmail.com.