Intelligent Automation & Soft Computing
DOI:10.32604/iasc.2021.015129
Article

Tech Science Press

ECG Encryption Enhancement Technique with Multiple Layers of AES and DNA Computing

Jamal Kh-Madhloom[1,2,*], Mohd Khanapi Abd Ghani[1] and Mohd Rizuan Baharon[1]

[1]BIOCORE Research Group, Faculty of Information & Communication Technology, Universiti Teknikal Malaysia Melaka, Melaka, 76100, Malaysia
[2]Computer Science Department, College of Computer Science and Information Technology, University of Wasit, Wasit, Iraq
*Corresponding Author: Jamal Kh-Madhloom. Email: jamalkh@uowasit.edu.iq
Received: 07 November 2020; Accepted: 23 January 2021

Abstract: Over the decades, protecting the privacy of a health cloud using the design of a fog computing network is a very important field and will be more important in the near future. Current Internet of Things (IoT) research includes security and privacy due to their extreme importance in any growing technology that involves the implementation of cryptographic Internet communications (ICs) for protected IC applications such as fog computing and cloud computing devices. In addition, the implementation of public-key cryptography for IoT-based DNA sequence testing devices requires considerable expertise. Any key can be broken by using a brute-force attack with ample computing power. Therefore, establishing a model of DNA cryptography is extremely necessary to improve the interaction between current and new technologies. In addition, the implementation of public-key cryptography for IoT-based DNA sequence testing devices requires considerable expertise. The proposed algorithm can create a stable hybrid encryption algorithm based on DNA layers and advanced encryption standard (AES) to shorten encryption time and increase protection capacity to suit the IoT health cloud systems. The proposed model can protect the DNA sequence over the fog computing cloud against plain text attacks by generating (I) main key, which is the key to the EAES encryption algorithm; (II) Rule 1 key, which represents the DNA base number of possible key probabilities; and (III) Rule 2 key, which represents the number of binding probabilities of the DNA helical structure. This key is built to achieve higher levels of protection. An ECG encryption enhancement technique with multilayer AES and DNA computing (MLAESDNA) is proposed in this study. Results show that MLAESDNA can secure IoT signals via cloud computing.

Keywords: ECG encryption; IoT; AES; DNA computing; MLAESDNA

1 Introduction

Cryptography is the science of securing the content of messages and communications. Cryptanalysis, the other subdiscipline, seeks to compromise or defeat the security achieved by cryptography. Mathematics is the foundation of cryptography and cryptanalysis. Cryptography is commonly associated with encryption, the transformation of data and information into a form that is unusable by a person who is not authorized to access that information. Historically, cryptography was used to protect the confidentiality of sensitive messages for military and diplomatic communications. Based on this traditional definition, cryptography can be seen as the science of encryption and decryption of messages, whose primary concern is to protect a message if it is disclosed to someone other than the intended recipient. With the expansion of information economy where transmission of sensitive information across untrusted media has become prevalent, the use of cryptography has become common practice not only with organizations but also with individuals; the scope of data transmission has exceeded the range of information sharing and entertainment to the core of industrial, scientific, and medical domains [1].

Recent implementations of cryptography considered cryptography as much more than the acts of encryption and decryption. While encryption and decryption techniques are used to secure sensitive information where confidentiality is important, other aspects of information security are implemented through encryption. These aspects include authentication of the message, sender, and recipient; the integrity of the message; and the nonrepudiation of the message transfer [2]. The term cryptography as used today refers collectively to techniques and applications used in protecting stored and transmitted information.

The Internet of Things (IoT) has enhanced the collection and sharing of data, and has made it more accessible to software applications and their users based on cloud computing and fog computing [3]. An IoT ecosystem consists of interconnected physical devices that support collection and exchange of data [4]. The concept of IoT has improved the connection between computer systems and the physical world. IoT provides numerous opportunities in different fields [5] including health care and telemedicine systems, which is one of the many industries that benefit from various IoT-based computer applications. It offers better health care to patients with improved treatment of various diseases [6]. Researchers in the field have been attempting to build better, secure IoT healthcare applications to expand healthcare services and provide remote care facilities for patients with chronic illnesses [5]. These applications use various sensory devices to collect patient data as ECG signals and administer treatment. The components of a typical IoT system are sensor devices, cloud-based interfaces, machine algorithms, and wireless sensor network (WSN). Sensor devices are used to collect data for the human body such as ECG signals, whereas WSN provides communication facilities [7]. Collected data are processed by the algorithms to perform the necessary analysis [8]. Moreover, cloud services offer storage facilities for collected data and allow access to users [9], whether patients or healthcare professionals. Secured [10] and protected IoT healthcare applications are required to improve the well-being of patients considering the security and privacy threats they pose to patients' lives and other implications such as privacy violations and financial risks. This study discusses the privacy and security issues in IoT healthcare applications by analyzing the components of application architecture.

Fog computing was developed to bridge the gap between IoT devices and data centers. The main purpose of fog computing is to speed up computing processing [3]. Cloud computing is not feasible for many IoT applications; therefore, fog computing is a perfect alternative. Fog computing is suitable for many IoT services because it has many extensive benefits such as reduced latency, decreased bandwidth, and enhanced security [5]. However, the characteristics of fog raise new security and privacy issues [6]. The existing security and privacy measures of cloud computing cannot be directly applied to fog [8]. Security implications of using fog computing for IoT systems are many features (the most important ones are confidentiality, integrity, and availability [CIA]) that we considered in the case of a. However, the characteristics of fog raise new security and privacy issues [3]. The existing security and privacy measures of cloud computing cannot be directly applied to fog computing [5,6,7]. Security implications of using fog computing for IoT systems are many features (the most important ones are CIA) that we considered in the case of a failure in defending the fog computing gateway [8,9,10].

The study aims to build a multilayer reliable system of DNA sequence incorporating DNA computing and the AES algorithm that can be implemented and integrated into the biological environment on DNA computers. This technique can secure the DNA sequence over cloud-based fog computing platforms against plain-text attacks via generation of main key and rule keys. The study introduces several contributions as (i) a multilayer encryption algorithm that incorporates DNA and the AES algorithm, (ii) a reliable encryption technique for IoT-based medical healthcare systems, (iii) an encryption technique with decrement of ECG message length and hence, decrement of complex mathematical operations, and (iv) an encryption technique that improves encryption power and provides higher security and more complexity to multilayer AES and DNA (MLAESDNA).

The remainder of this study is structured as follows. Section 2 displays the current related work. Section 3 offers the indepth process of the suggested model. Section 4 provides the experiment results and their discussion. Section 5 presents the conclusions.

2 Related Work

The rapidly growing applications of telemedicine and healthcare recently imposed the need for securing the transmission of medical data and records over the Internet or any other medium. This need motivated researchers to focus on the enhancements and modifications of existing encryption algorithms as well as develop new algorithms, as illustrated in Section 2.5. DNA inspired security encryption algorithm development due to the advanced, reliable method of encryption it is based on. Thus, several attempts have been made to enhance the standard security and encryption algorithms inspired by the DNA method of encryption.

The current study focuses on comparing and analyzing the encryption enhancement trials in the steganography sector as reported in the literature published during the previous years. A general trend is to strengthen a new encryption algorithm and counter the great power of computing, especially the new generation of quantum computing device. Traditional cryptographic systems are built on strong mathematical and theoretical bases. Therefore, several researchers are interested in developing a new DNA-based AES encryption algorithm, as mentioned in Tab. 1. This algorithm will be helpful in all technologies that typically deal with an extensive number of connected devices and sensitive data stores, and exchange data between those devices. Thus, security and privacy are important factors in such applications and related applications. Moreover, the platforms must be able to achieve the data security requirements in every approach. During the study, we explore various AES enhancement studies using several approaches. Further, we examine 11 peer-reviewed articles and analyze the proportion of the enhancement of encryption algorithms for data transmission in different applications. Recently, many research articles related to AES have focused on using DNA to increase the power of encryption and have proposed algorithms to overcome the problems as high capacity, unpredictable, high-deterioration steganography techniques, where the data will not be visible to hackers even though the system is hacked. We focus on the AES algorithm and DNA addressed in recent studies. We explore different types of methods and find that the new trend is using DNA in enhancing the AES algorithm. We also study and analyze the traditional methods and their issues in encryption. We demonstrate substantial research areas including AES and DNA hybrid system usage.

Table 1: AES enhancements attempts

Ref	Existing Methods	Technique	Advantages	Limitations
[11]	Modifying the AES algorithm to be used for image ciphering, especially HD	DNA computing and round-reduced AES block cipher integration	High security level	Not applied to network smart applications
[12]	Data encryption standard (DES). Triple DES (T-DES). Advanced encryption standard (AES)	New approach of the AES algorithm DNA cipher and key are merged and transmitted along a channel in protein form	Dynamic key generation Key value manipulation Improved security levels	Cipher and key overhead Protein bases are added to cipher and nth round key High computational cost
[13]	==	Deep learning encryption Key generation using genetic algorithm with NW algorithm	Hiding data in a DNA sequence and deep learning	Computation basis is storage capacity required for DNA
[14]	RSA DES NTRU	DNA based-bit-based design and implementation of AES DNA specification consideration	Building a complex DNA basis system is possible. Suits biological environment applicable for DNA machines	As strong and robust as the standard algorithm
[15]	==	DNA-based DNAES sequences with silent mutations	Applicable to any type of data Applicable for biological environment DNA sequence hidden using cipher	Same security level as AES
[16]	==	Altering the AES MixColumns transformation DNA-inspired methods from processes and structure	Analysis of security new MixColumns Block cipher values tested using NIST Test Suite	Same key length of AES

Table 1 (continued).

Ref	Existing Methods	Technique	Advantages	Limitations
[17]	DES IDEA AES	Modified AES algorithm Reduced algorithm calculations Improving encryption performance	Adjustment of ShiftRow Transformation No additional operations or hardware needed Stronger video data security against statistical threats	Same key length of AES
[18]	(ECC) GEO encryption T-DES	Hybrid confidentiality algorithm	Strong IoT data confidentiality	Same key length of AES

The AES encryption algorithm is a common encryption algorithm developed by NIST to replace DES [19]. To encrypt and decrypt data packets, the procedure of AES algorithm first encrypts 10 iterations for 128-bit encryption keys at the first step, 12 encoding iterations for 192-bit encryption keys at the second step, and 14 encoding iterations to 256-bit encryption keys to produce the final encrypted message. Fig. 1 illustrates the flowchart of the AES [20]. These steps can be defined as follows:

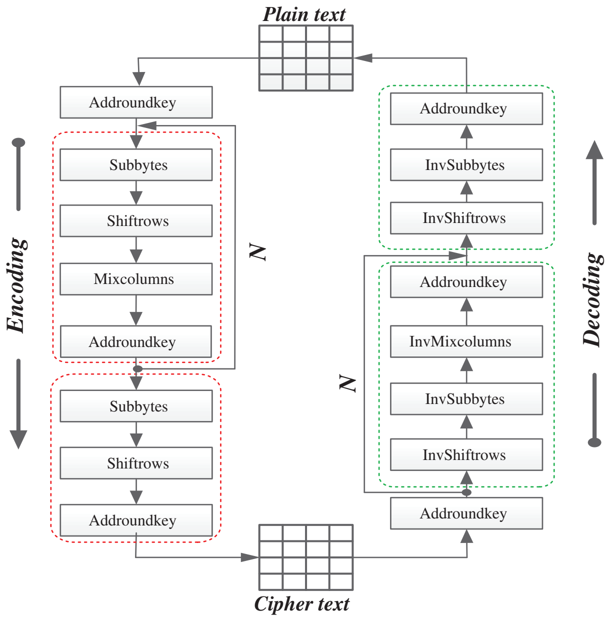


Figure 1: Flowchart of the AES algorithm [25]

a) SubBytes transformation: AES consists of a 128-bit block of data, which means every entity in the database consists of 16 bytes. Additional byte transformation requires that every entity of a data record is converted into another form of data using an eight-bit Rijndael S-box [21].

b) ShiftRows transformation: In this easy transposition, data in the remaining three lines of the state that are dependent on the row position are transformed in one cycle to another location. In the following line, a one-byte circular shift to the left is done. In rows 3 and 4, two- and three-byte circular transformations to the left are executed sequentially [22].

c) Mix Columns transformation: This transposition is similar to multiplying the states represented by columns with a matrix [23]. The values of the column vector are multiplied by a matrix with constant values as a polynomial rather than numbers.

d) AddRound Key transformation: The current state and the encryption key are XORed in this transformation. Hence, this transformation is the inverse of its own. The transformation is composed of several steps. The initial step, AddRound Key operation, is performed followed by the processing of data block that consists of SubBytes, ShiftRows via round function, Mix Columns, and AddRound Key transformation [14]. This procedure is performed iteratively based on the length of the key. The same sequence is followed for the decryption operation. The key schedules are made identical for encryption and decryption transformations by performing Inv-SubBytes, Inv-ShiftRows, Inv-Mix Columns, and AddRound Key [24].

In Singh [19], DNA computing and round-reduced AES block cipher are combined; in the existing method, images of dimensions use n × m = 256 × 256 pixels. However, it has not been applied to any smart network application. Moving picture experts' group-based encryption algorithm for video AES is proposed with modification in ShiftRows transformation. Operation or hardware is not required in addition to the original AES [19]. High-definition image encryption algorithm based on AES algorithms was proposed in Albahar et al. [25]. The well-known AES with a more secure block cipher algorithm was introduced; the limitation of this technique is longer processing time, and the number of rounds is reduced by attacks on the encryption algorithm [26]. With the same objective, a hybrid RSA- and AES-based encryption algorithm for securing user's data was introduced in the cloud [27]. The power of using RSA and AES encryption algorithm provides three encryption keys, namely, (i) public key for encryption, (ii) private key, and (iii) secret key for decryption.

In the existing research, an AES algorithm for data security is designed to provide more security using a Polybius square matrix, thus increasing the number of rounds. Another work generated the key using chaotic maps where encryption is accomplished using AES [26]. To improve encryption speed, authors synchronized the unit of key expansion where a RoundKey is generated in each clock cycle, and keys are stored and retrieved from the key RAM in the same clock cycle. Moreover, in Zhang et al. [27], authors introduced an encryption technique for digital images based on the AES encryption algorithm and concluded that their proposed technique can cope with the effect of encryption and decryption.

DNA computing refers to the concept of using biological neurons and molecules, rather than digital computers, to perform complex computations. This area of science was recently explored by an American scientist named Leonard Adelman. His contributions showed how biological molecules can be implemented and studied to solve complex mathematical computations. Initially, no relationship was observed between molecules and cryptography, but excessive research in this area established a new field of science that related the biological molecules and the science of encryption to enhance the features and capabilities of biological molecules for the science of cryptography [28,29]. DNA computer is a group of DNA strands that are collected together to solve a computational problem. Technology enables selecting proper strands and manipulating the solution, showing how huge, complex computational problems can be solved faster than the traditional computer that requires distinct processing and memory capabilities. DNA is utilized to solve such problems for the following reasons: [30].

a) It supports much denser information than traditional computers that require 1,000,000,000,000 cubic nanometers to store storage media, such as videotapes.

b) The DNA processes operations in parallel using trillions of strands because each operation on a test tube of DNA is carried out on all strands in the tube in parallel.

The linear operation of traditional computers implies that data can be manipulated in one block after another. For example, chemical reactions in biological environments occur in a parallel fashion, and every step composing these reactions influences numerous strands within the DNA sequence. Using the DNA computer for these calculations is much beneficial because it requires less energy and memory space than conventional computers [31,32]. The field of DNA computing involves not only biology scientists but also scientists from different disciplines such as computer sciences, physics, chemistry, and mathematics.

3 Methodology

Current cryptographic algorithms have a mathematical basis. DNA-inspired algorithms are combinations of current and new cryptographic technology. This section describes in detail MLAESDNA based on data encryption that integrates AES and DNA computing. MLAESDNA aims to enhance security by increasing the key length size using the DNA layers around the AES algorithm, which leads to preventing the piracy of the illegal users.

MLAESDNA, as shown in Fig. 2, uses DNA computing techniques to increase encryption layer with the AES algorithm. The different transformations of the encryption are sequentially applied to the state. The transformation layers are ECG binarization, DNA conversion, AddRoundKey, SubBytes, ShiftRows, Mix Columns, and DNA swapping [22]. It represents a block diagram for the encryption/decryption using the multiple layers of DNA and AES algorithms of the proposed model. MLAESDNA aims to strengthen the AES algorithm and reach the highest level of security that depends on the key length in DNA and AES algorithms. In the AES algorithm, the same operations are performed many times on a fixed number of rounds. The number of rounds depends on the key size. The proposed algorithm uses a block cipher size of 128 bits and N of rounds.

DNA encryption is used to increase the key length, which adds more complexity to the AES such that it becomes immune in a manner that adapts the technological development. The key length of the proposed algorithm is (24 × 2128 × 3 × 10) bits and is calculated as follows:

- **First Key:**

The DNA key size is 24. This key introduces the DNA sequence where the probability of the key could be

$$\begin{aligned} A &= 11\ \underline{\text{or}}\ 10\ \underline{\text{or}}\ 01\ \underline{\text{or}}\ 00,\\ T &= 11\ \underline{\text{or}}\ 10\ \underline{\text{or}}\ 01\ \underline{\text{or}}\ 00,\\ C &= 11\ \underline{\text{or}}\ 10\ \underline{\text{or}}\ 01\ \underline{\text{or}}\ 00,\\ G &= 11\ \underline{\text{or}}\ 10\ \underline{\text{or}}\ 01\ \underline{\text{or}}\ 00. \end{aligned}$$

- **Second Key:**

The standard AES main key size is 2128.

- **Third Key:**

The key size is 3 according to three different DNA bases which can be represented as one of the following:

$$(A = C,\ G = T) \text{ or } (A = G,\ T = C) \text{ or } (A = T,\ G = C).$$

- **Fourth Key:**

The standard AES round number 10.

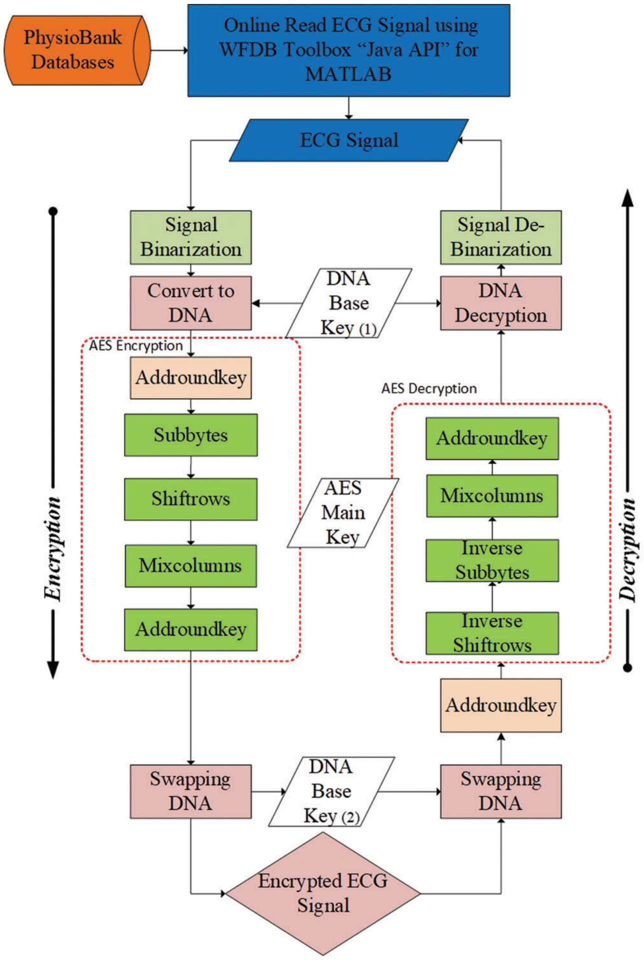


Figure 2: Flowchart of the MLAESDNA

The steps of the proposed method are explained in the following sections. The steps executed during the operation of the algorithm are outlined, and each following step is essential for the procedure of the algorithm and designed according to the algorithm design considerations to produce better algorithm performance metrics.

3.1 ECG Binarization "Preprocessing"

This step intends to convert ECG signals into binary bits using MATLAB functions to suit DNA conversion, as illustrated in Fig. 3.

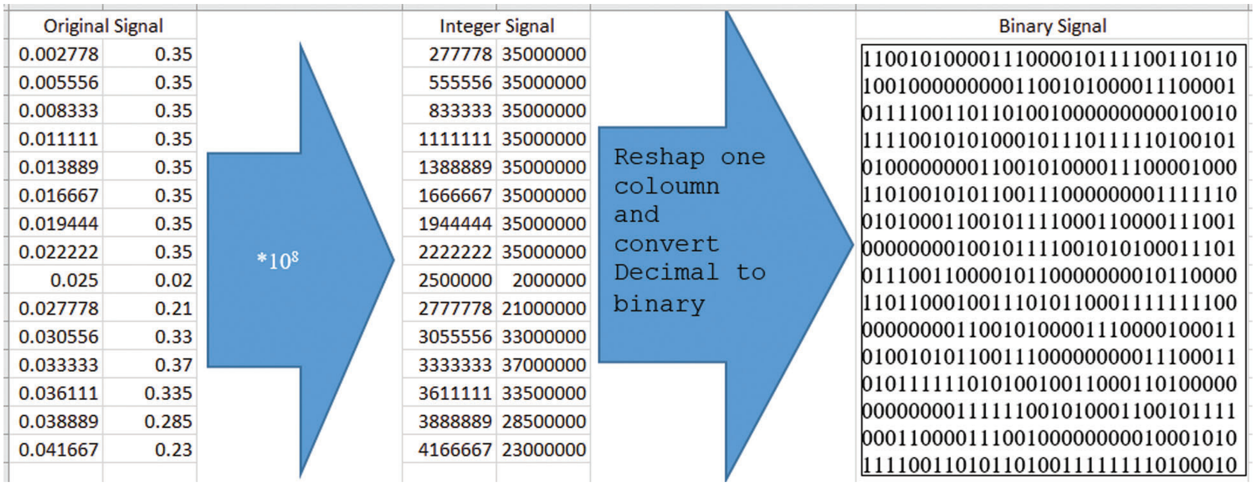
3.2 Conversion to DNA

DNA encryption starts with transforming the binary message obtained from the previous step through the variable DNA bases into a DNA helix. Tab. 2 illustrates DNA Rule (1) "key size is 24." Fig. 4 illustrates a sample of DNA helix conversion.

3.3 SubBytes Operation

The SubBytes operation is a nonlinear byte substitution that operates on each byte of the state independently, as shown in Fig. 5. The substitution table (S-Box) is invertible and independent of any input, and precalculated forms are used. Each byte of the state is then substituted by the value in the S-Box whose index corresponds to the value in the state:

$$b(x, y) = \mathrm{SBox}[\mathrm{a}(x, y)]$$



Original Signal	
0.002778	0.35
0.005556	0.35
0.008333	0.35
0.011111	0.35
0.013889	0.35
0.016667	0.35
0.019444	0.35
0.022222	0.35
0.025	0.02
0.027778	0.21
0.030556	0.33
0.033333	0.37
0.036111	0.335
0.038889	0.285
0.041667	0.23

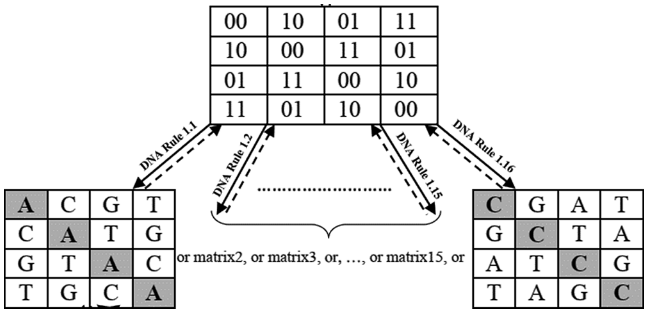
Integer Signal	
277778	35000000
555556	35000000
833333	35000000
1111111	35000000
1388889	35000000
1666667	35000000
1944444	35000000
2222222	35000000
2500000	2000000
2777778	21000000
3055556	33000000
3333333	37000000
3611111	33500000
3888889	28500000
4166667	23000000

Binary Signal
11001010000111000010111100110110
10010000000001100101000011100001
01111001101101001000000000010010
11110010101000101110111110100101
01000000001100101000011100001000
11010010101100111000000001111110
01010001100101111000110000111001
00000000100101111001010100011101
01110011000010110000000010110000
11011000100111010110001111111100
00000000110010100001110000100011
01001010110011100000000011100011
01011111101010010011000110100000
00000000111111001010001100101111
00011000011100100000000010001010
11110011010110100111111110100010

Figure 3: Flowchart of ECG preprocessing

Table 2: DNA rule (1) with key size = 24

Rules	A	T	C	G
Rule 1.1	11	10	01	00
Rule 1.2	11	10	00	01
Rule 1.3	11	00	10	01
Rule 1.4	00	11	10	01
Rule 1.5	00	11	01	10
Rule 1.6	00	01	11	10
Rule 1.7	01	00	11	10
Rule 1.8	01	00	10	11
Rule 1.9	01	10	00	11
Rule 1.10	10	01	00	11
Rule 1.11	10	01	11	00
Rule 1.12	10	11	01	00
Rule 1.13	10	11	00	01
Rule 1.14	10	00	11	01
Rule 1.15	00	10	11	01
Rule 1.16	00	10	01	11



00	10	01	11
10	00	11	01
01	11	00	10
11	01	10	00

A	C	G	T
C	A	T	G
G	T	A	C
T	G	C	A

C	G	A	T
G	C	T	A
A	T	C	G
T	A	G	C

Figure 4: Sample of DNA Helix conversion

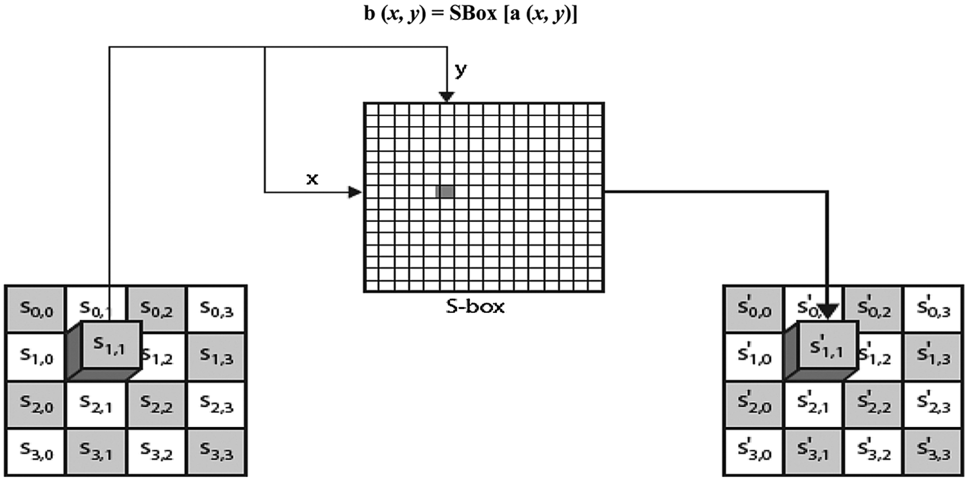
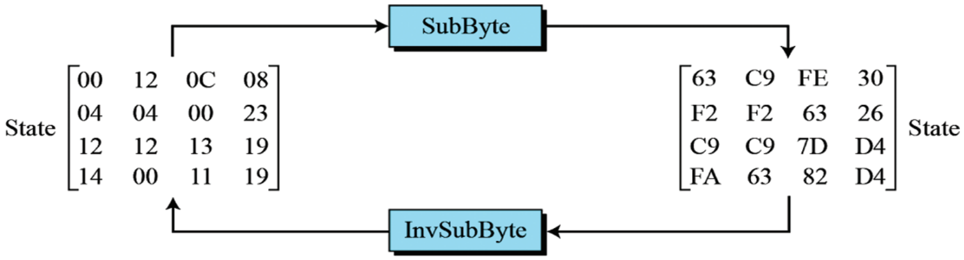

b (x, y) = SBox [a (x, y)]
y
x
S-box
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{1,1}$ $s_{1,2}$ $s_{1,3}$
$s_{2,0}$ $s_{2,1}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,1}$ $s_{3,2}$ $s_{3,3}$
$s'_{0,0}$ $s'_{0,1}$ $s'_{0,2}$ $s'_{0,3}$
$s'_{1,0}$ $s'_{1,1}$ $s'_{1,2}$ $s'_{1,3}$
$s'_{2,0}$ $s'_{2,1}$ $s'_{2,2}$ $s'_{2,3}$
$s'_{3,0}$ $s'_{3,1}$ $s'_{3,2}$ $s'_{3,3}$

Figure 5: SubBytes operation representation

The inverse of SubBytes is the same operation using the inversed S-Box, which is also precalculated, and is a SubBytes step, as shown in Fig. 6.



$$\text{State}\begin{bmatrix} 00 & 12 & 0C & 08 \\ 04 & 04 & 00 & 23 \\ 12 & 12 & 13 & 19 \\ 14 & 00 & 11 & 19 \end{bmatrix} \qquad \begin{bmatrix} 63 & C9 & FE & 30 \\ F2 & F2 & 63 & 26 \\ C9 & C9 & 7D & D4 \\ FA & 63 & 82 & D4 \end{bmatrix}\text{State}$$

Figure 6: SubBytes and inverse of subbytes transformations

In the SubBytes step, each byte in the state is replaced with its entry in a fixed eight-bit lookup table, S; b (i, j) = S (i, j). Tabs. 3 and 4 illustrate S-Box and Inverse S-Box table, respectively. The SubBytes transformation and InvSubBytes transformation is the inverse of each other.

Table 3: S-Box table

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Table 4: Inverse S-Box table

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

3.4 ShiftRows Operation

In this operation, each row of the state is cyclically shifted to the left, depending on the row index.

a) The 1st row is shifted 0 positions to the left.

b) The 2nd row is shifted 1 position to the left.

c) The 3rd row is shifted 2 positions to the left.

d) The 4th row is shifted 3 positions to the left.

The inverse of ShiftRows is the same cyclically shift but to the right. It is needed later for decoding. In the ShiftRows step, bytes in each row of the state are shifted cyclically to the left. The number of places each byte is shifted differs for each row. Figs. 7 and 8 show ShiftRows schema and example of ShiftRows and InvShiftRows, respectively.

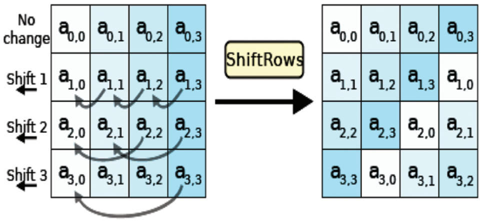
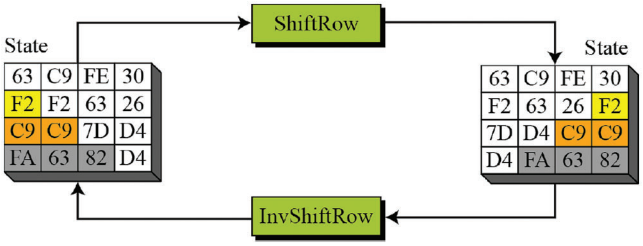

No change
Shift 1
Shift 2
Shift 3
ShiftRows
$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,1}$ $a_{1,2}$ $a_{1,3}$ $a_{1,0}$
$a_{2,2}$ $a_{2,3}$ $a_{2,0}$ $a_{2,1}$
$a_{3,3}$ $a_{3,0}$ $a_{3,1}$ $a_{3,2}$

Figure 7: ShiftRows Schema



63	C9	FE	30
F2	F2	63	26
C9	C9	7D	D4
FA	63	82	D4

63	C9	FE	30
F2	63	26	F2
7D	D4	C9	C9
D4	FA	63	82

Figure 8: Inverse ShiftRows Schema

3.5 Mix Columns Operation

In the Mix Columns step, the four bytes of each column of the state are combined using an invertible linear transformation. The Mix Columns function takes four bytes as input and outputs four bytes, where each input byte affects all four output bytes. Together with ShiftRows, Mix Columns provides diffusion in the cipher. During this operation, each column is multiplied by the known matrix that for the 128-bit key is

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}$$

The multiplication operation is defined as follows: Multiplication by 1 means no change, multiplication by 2 means shifting to the left, and multiplication by 3 means shifting to the left and then performing XOR with the initial unshifted value. After shifting, a conditional XOR with 0x1B should be performed if the shifted value is larger than 0xFF, as illustrated in Fig. 9. The Mix Columns step can also be viewed as a multiplication by a particular MDS matrix in a finite field. This process is described further in the article Rijndael mix columns.

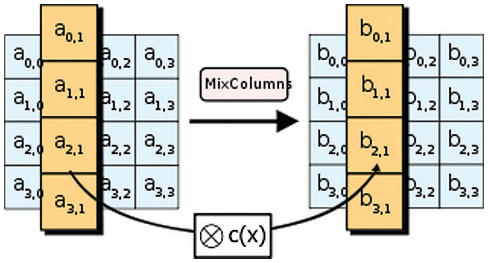

$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
MixColumns
$\otimes$ c(x)
$b_{0,0}$ $b_{0,1}$ $b_{0,2}$ $b_{0,3}$
$b_{1,0}$ $b_{1,1}$ $b_{1,2}$ $b_{1,3}$
$b_{2,0}$ $b_{2,1}$ $b_{2,2}$ $b_{2,3}$
$b_{3,0}$ $b_{3,1}$ $b_{3,2}$ $b_{3,3}$

Figure 9: Mix Columns Operations

3.6 AddRoundKey Operation

In this operation, a Round Key is applied to the state by a simple bitwise XOR. The Round Key is derived from the Cipher Key by the means of the key schedule. The Round Key length is equal to the block key length 128 bits. Fig. 10 illustrates the AddRoundKey operation, where the subkey is combined with the state. For each round, a subkey is derived from the main key using Rijndael's key schedule, and each subkey is the same size as the state. The subkey is added by combining each byte of the state with the corresponding byte of the subkey using bitwise XOR. In the AddRoundKey step, each byte of the state is combined with a byte of the round subkey using the XOR operation.

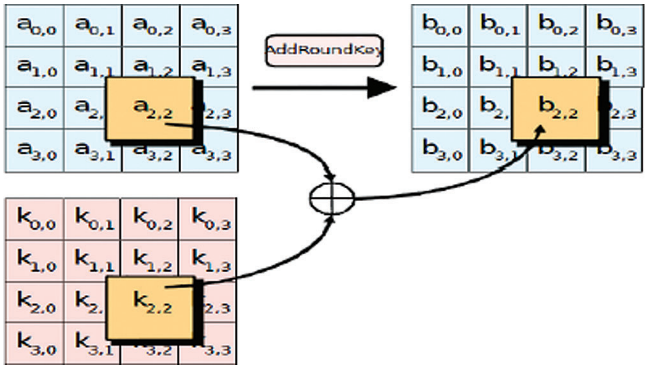

$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,}$ $a_{2,2}$ $_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
AddRoundKey
$b_{0,0}$ $b_{0,1}$ $b_{0,2}$ $b_{0,3}$
$b_{1,0}$ $b_{1,1}$ $b_{1,2}$ $b_{1,3}$
$b_{2,0}$ $b_{2,}$ $b_{2,2}$ $_{2,3}$
$b_{3,0}$ $b_{3,1}$ $b_{3,2}$ $b_{3,3}$
$k_{0,0}$ $k_{0,1}$ $k_{0,2}$ $k_{0,3}$
$k_{1,0}$ $k_{1,1}$ $k_{1,2}$ $k_{1,3}$
$k_{2,0}$ $k_{2,}$ $k_{2,2}$ $_{2,3}$
$k_{3,0}$ $k_{3,1}$ $_{3,2}$ $_{3,3}$

Figure 10: AddRoundKey operation

3.7 DNA Swapping

DNA swapping starts with turning the hexadecimal message obtained from the AddRoundKey step into a binary message. This message is transformed through the variable DNA bases into a DNA helix. Then, it is ciphered through the DNA bases to present a wholly different outcome that is returned into binary text. Once again, the message is transformed into a decimal message. Tab. 5 illustrates the sequence of DNA swapping operation for key size = 3.

Table 5: DNA Rule (2)

No. of Rule	DNA sequence
Rule 2.1	A = T, C = G
Rule 2.2	A = C, T = G
Rule 2.3	A = G, T = C

4 Experimental Results and Security Analysis

The experimental analysis includes different security tests and results such as keyspace analysis, statistical analysis, numerical analysis, differential analysis, and encryption quality. These tests are the most considerable tests to demonstrate the satisfactory security of the proposed algorithm. The PhysioBank dataset, a large, growing archive of well-characterized digital recordings of physiologic signals and related data for use by the biomedical research community, is used in this study.

4.1 Simulation Environment

The proposed technique was simulated using a reliable simulation tool, namely, "MATLAB version (2017b)." Tab. 6 shows the specifications of simulation. The experiments were performed using Microsoft Windows platform deployed on a machine with the following specifications:

Table 6: Simulation machine specifications

Specification	Details
Model	Dell Inspiron 5000 series
CPU	4 GHz Intel Core i7-5500U
CPU speed	3.40 GHz (dual-core, 4 MB cache, up to 3 GHz with Turbo Boost)
Generation	8th generation
Graphics	AMD Radeon R7 M265
Memory	16 GB
Storage	2TB HDD
OS	Microsoft Windows 10 version 1909 build 18363.693

4.2 Encryption and Decryption Time Analysis

Encryption and decryption time can be used to calculate the encryption and decryption throughput of the algorithms. The performance parameters include the time taken by the algorithm for the encryption and decryption of input ECG signals. To avoid biased results, the experiment was run 10 times, and the average of the results was considered the average of the experiment. Tab. 7 shows the Encryption and decryption execution time.

Table 7: Encryption and decryption execution time with different rounds "signal length = 1000"

ECG Name	AES Only		2-rounds AES and DNA		5-rounds AES and DNA		10-rounds AES and DNA	
	ET (s.)	DT (s.)	ET (s.)	DT (s.)	ET (s.)	DT (s.)	ET (s.)	DT (s.)
Mitdb/100	25.77	50.03	2.40	3.66	5.87	8.71	11.35	17.84
Mitdb/105	23.14	50.70	2.41	3.80	5.98	9.28	14.03	18.54
:	:	:	:	:	:	:	:	:
Mitdb/217	24.89	53.21	2.68	4.35	6.37	10.03	12.38	17.86
Mitdb/219	24.27	54.30	2.50	4.31	6.10	10.06	13.61	18.67

*ET (s.) = Encryption time (second), DT (s.) = Decryption time (second).

4.3 Security Analysis

A complete investigation was conducted on the security of the proposed encryption technique. Several security analysis methods are used to test a cipher's resistance to different types of attacks. Keyspace analysis is used to measure the resistance to brute-force attack. Histogram, correlation analysis of the adjacent values, and correlation analysis of the original and encrypted ECG signal are used to measure the resistance to statistical attack. Numerical analysis, for instance entropy, is a measurement of randomness. Mean square error (MSE) is used to evaluate the performance of implemented focus measures to the ECG signal quality.

4.4 Keyspace Analysis

The proposed algorithm was compared with other works, and the results suggest that the proposed algorithm needs (5.179340 × 1027) years to be broken or hacked. Tab. 8 shows that the proposed algorithm is a better security against brute-force attacks and needs a long time for breaking compared with other works.

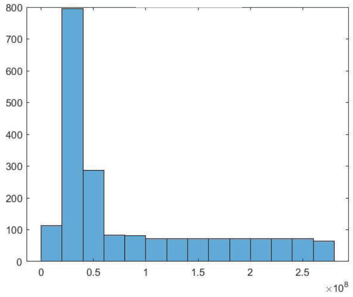
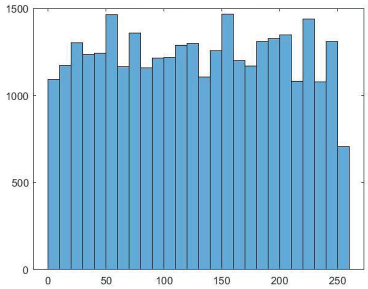
Table 8: Results of decryption breaking time

Technique Name	Key Length (bits)	Keyspace	Amount of Time for Breaking (years)
AES Original	128	2^{128}	1.078950×10^{25}
LEA: [32]	128	2^{128}	1.078950×10^{25}
MLAESDNA	$4 \times 128 \times 3 \times 10$	$2^4 \times 2^{128} \times 3 \times 10$	5.179340×10^{27}

4.5 Histogram Analysis

The proposed algorithm was applied to various ECG signals. Tab. 9 shows that the histograms of the cipher-ECG signal are very uniform and remarkably different from those of the plain ECG signal, which makes statistical analysis attacks on the encrypted ECG signal very difficult.

Table 9: Histograms of original and encrypted ECG signals used in simulations

ECG Name	Original Signal	Cipher Signal
Mitdb/100		

4.6 Correlation Analysis

To test the effectiveness of the cryptosystem, the correlation between two contiguous values was examined in the plain ECG signal and the cipher ECG signal using the following procedure: First, 50 pairs (horizontal, vertical, and diagonal) of adjacent values from the original ECG signal and the encrypted ECG signal were randomly selected. Then, the correlation coefficient of each pair was calculated [33]. Tab. 10 shows the distribution of adjacent pixel pairs of the plain ECG signal and its cipher ECG signal in the horizontal, vertical, and diagonal directions. The pixel pairs of plain ECG signals are mostly located nearby the diagonal line in the graph.

4.7 Information Entropy Analysis

Entropy is one of the most important features that define the level of randomness and uncertainty in an ECG signal and is widely used to measure the uniform distribution of pixel gray-level in the ECG signal. The entropy is close to 8; therefore, the diffusion is good and produces a high disorder at output [34–36]. The entropy information of the encrypted ECG signals is shown in Tab. 11. The results demonstrate that the information entropies of the cipher ECG signals are close to the ideal value, which can verify that the cipher ECG signal of the proposed algorithm has good randomness.

Table 10: Horizontal, vertical, and diagonal correlations of original and encrypted ECG signals

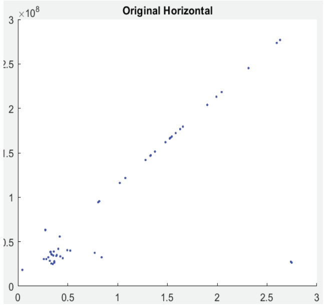
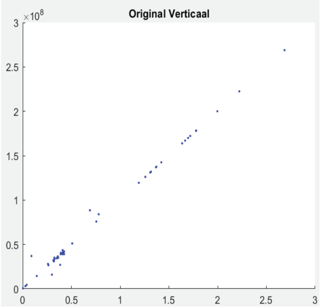
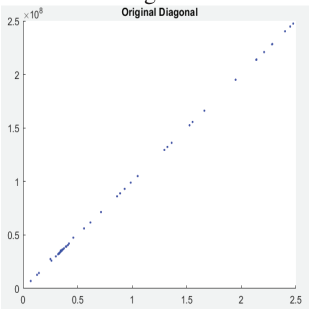
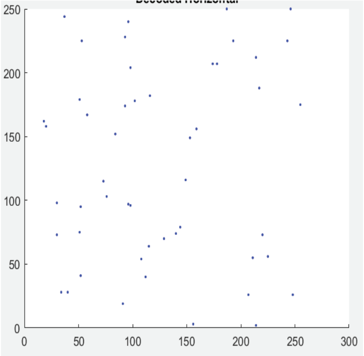
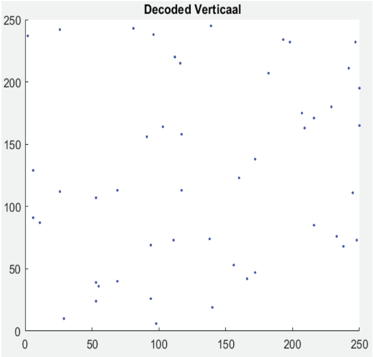
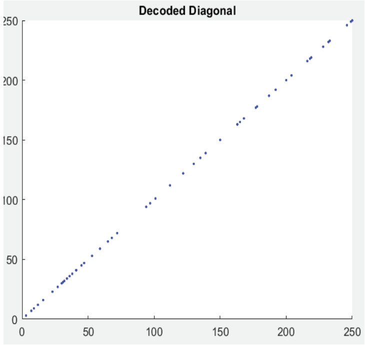
MITDB/100		
Horizontal Correlation of Original 	Vertical Correlation of Original 	Diagonal Correlation of Original 
Horizontal Correlation of Encrypted 	Vertical Correlation of Encrypted 	Diagonal Correlation of Encrypted 

Table 11: Results of information entropy of original and encrypted ECG signals "signal length = 1000"

ECG Signal Name	Original ECG Signal		Encrypted ECG Signal	
	AES	MLAESDNA	AES	MLAESDNA
Mitdb/100	0.00620406	0.00620406	0.0494641	0.0362337
Mitdb/105	0.0114078	0.0114078	0.0535951	0.0452119
:	:	:	:	:
Mitdb/217	0.0114078	0.0114078	0.0906764	0.0452119
Mitdb/219	0.00620406	0.00620406	0.0494641	0.0362337

4.8 Mean Square Error

To measure the encryption strength of the proposed algorithm, several quantitative metrics such as MSE are utilized to estimate the variance between the encrypted ECG signal and the original ECG signal. Tab. 12

illustrates the MSE statistical metric result, which is used to evaluate the proposed algorithm. MSE is a very simple, very common distortion measure. The value of MSE represents the difference between the original ECG signal and the decrypted ECG signal. The smaller the MSE is, the better the result.

Table 12: MSE of "Signal Length = 1000"

ECG Signal Name	AES	MLAESDNA
Mitdb/100	0	0
Mitdb/105	0	0
:	:	:
Mitdb/217	0	0
Mitdb/219	0	0

Security is the major issue of any encryption technique. A good encryption algorithm should encounter most kinds of recognized attacks. Keyspace analysis is used to measure the resistance to brute-force attack. In the proposed algorithm, the keyspace is equal to (24 × 2128 × 3 × 10). This value exceeds the effective key size necessary to ensure computational security against future brute-force attacks. The histograms of the ciphered ECG signals are clearly steady and considerably different from those of the original ECG signals, which means performing statistical cryptanalysis on the ciphered ECG signal is very difficult. The correlation coefficient values indicate that the value distribution of the cipher ECG signals show a wide distortion of the correlation among values. Thus, the value information cannot be obtained from the adjacent values. Moreover, the information entropies of the cipher ECG signals are close to the ideal value, which can verify that the cipher ECG signal of the proposed algorithm has a good randomness. Therefore, the proposed algorithm is strongly resistant to differential attacks. These results are achieved due to the strong process of confusion and diffusion of the proposed algorithm. MSE is used to evaluate the performance of implemented focus measures. Remarkably better results are achieved with the proposed algorithm. The conducted experiments and results of various statistical measures demonstrate the resistance of the proposed algorithm to classical types of attack.

The results of histogram analysis, the correlation among adjacent values, the entropy results, and the MSE results demonstrate that MLAESDNA is resistant to statistical attacks. These results are related to the high sensitivity of the three different keys and the high randomization of DNA computing. Furthermore, this study could make a breakthrough into the era of cryptographic algorithm design and implementation in medical fog-computing-based healthcare applications. The implementation of the proposed technique in other domains may be altered by variable platform architectures. The study is also limited to securing medical messages other than ECG signals whose transmission requirements and metrics may vary. The proposed technique also encountered the generation and processing of four keys offered by DNA rules. These combinations may result in encryption and processing time that may be a critical issue in public health safety and emergency IoT-based applications.

5 Conclusion

This study presents MLAESDNA, a multilayer encryption algorithm incorporating DNA computing and AES algorithm. Increasing the key length has many advantages in IoT, especially in medical health systems, because it decreases the ECG message length and the complex mathematical operations that use more resources and take a longer time to process. MLAESDNA uses four keys offered by DNA rules, which improves encryption power and provides higher security and more complexity. The required decryption breaking time is remarkably increased more than 48 times of the breaking time using the original algorithm. Combining the concept of AES and DNA computing successfully enhances the encryption/decryption processes. The results show that MLAESDNA is better than the original AES algorithm and other algorithms. The results of the experiments conclude that MLAESDNA provides a high level of security, integrity, efficiency, and robustness. MLAESDNA fulfils the requirements needed to transfer the ECG signals over insecure healthcare system channels. In general, the area of joint encryption is a rich area for research. In the future work, the speed of the encryption and decryption execution time will be enhanced by integrating the quantum computing concept with MLAESDNA, applying parallel processing for MLAESDNA, and applying MLAESDNA on all medical signals in industry.

Acknowledgement: The authors thank the BIOCORE Research Group, Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka.

Funding Statement: The author(s) received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. Nagaraj, G. S. V. P. Raju and V. Srinadth, "Data encryption and authetication using public key approach," *Procedia Computer Science*, vol. 48, no. 2, pp. 126–132, 2015.

[2] M. R. Ogiela and L. Ogiela, "Cognitive cryptography techniques for intelligent information management," *International Journal of Information Management (IJIM)*, vol. 40, no. 2, pp. 21–27, 2018.

[3] A. A. Mutlag, M. K. Abd Ghani, N. Arunkumar, M. A. Mohammed and O. Mohd, "Enabling technologies for fog computing in healthcare IoT systems," *Future Generation Computer Systems*, vol. 90, no. 3, pp. 62–78, 2019.

[4] S. A. Mostafa, S. S. Gunasekaran, A. Mustapha, M. A. Mohammed and W. M. Abduallah, "Modelling an adjustable autonomous multi-agent internet of things system for elderly smart home," *Advances in Intelligent Systems and Computing*, vol. 953, pp. 301–311, 2020.

[5] A. M. Rahmani, T. N. Gia, B. Negash, A. Anzanpour, I. Azimi *et al.,* "Exploiting smart e-health gateways at the edge of healthcare internet-of-things: A fog computing approach," *Future Generation Computer Systems*, vol. 78, no. 7, pp. 641–658, 2018.

[6] A. A. Mutlag, M. K. Abd Ghani, M. A. Mohammed, M. S. Maashi and O. Mohd, "MAFC: Multi-agent fog computing model for healthcare critical tasks management," *Sensors*, vol. 20, no. 7, pp. 1853, 2020.

[7] I. Medvediev, O. Illiashenko, D. Uzun and A. Strielkina, "IoT solutions for health monitoring: Analysis and case study," in *IEEE 9th Int. Conf. on Dependable Systems, Services and Technologies (DESSERT)*. IEEE, pp. 163–168, 2018.

[8] K. H. Abdulkareem, M. A. Mohammed, S. S. Gunasekaran, M. N. Al-Mhiqani, A. A. Mutlag *et al.,* "A review of fog computing and machine learning: Concepts, applications, challenges, and open issues," *IEEE Access*, vol. 7, pp. 153123–153140, 2019.

[9] A. Chacko and T. Hayajneh, "Security and privacy issues with IoT in healthcare," *EAI Endorsed Transactions on Pervasive Health and Technology*, vol. 4, no. 14, pp. 155079, 2018.

[10] M. K. A. Ghani, M. A. Mohammed, M. S. Ibrahim, S. A. Mostafa and D. A. Ibrahim, "Implementing an efficient expert system for services center management by fuzzy logic controller," *Journal of Theoretical and Applied Information Technology*, vol. 95, no. 13, pp. 3127–3135, 2017.

[11] E. Shehab, A. K. Farag and A. Keshk, "An image encryption technique based on DNA encoding and round-reduced AES block cipher," *International Journal of Computer Applications*, vol. 107, no. 20, pp. 1–7, 2014.

[12] B. M. Krishna, H. Khan, G. L. Madhumati, B. Lohitha, E. Bhavitha *et al.,* “FPGA implementation of DNA based AES algorithm for cryptography applications,” *International Journal of Pure and Applied Mathematics*, vol. 115, no. 7, pp. 525–530, 2017.

[13] S. Kalsi, H. Kaur and V. Chang, “DNA cryptography and deep learning using genetic algorithm with NW algorithm for key generation,” *Journal of Medical Systems*, vol. 42, no. 1, pp. 1021, 2018.

[14] M. Sabry, M. Hashem, T. Nazmy and M. E. Khalifa, “Design of DNA-based advanced encryption standard (AES),” in *IEEE Seventh Int. Conf. on Intelligent Computing and Information Systems (ICICIS)*, IEEE, pp. 390–397, 2015.

[15] H. M. Bahig and D. I. Nassr, “DNA-based AES with silent mutations,” *Arabian Journal for Science and Engineering*, vol. 44, no. 4, pp. 3389–3403, 2019.

[16] A. H. Al-Wattar, R. Mahmod, Z. A. Zukarnain and N. Udzir, “A new DNA based approach of generating key dependent mix columns transformation,” *International Journal of Computer Networks & Communications*, vol. 7, no. 2, pp. 93–102, 2015.

[17] P. Deshmukh and V. Kolhe, “Modified AES based algorithm for MPEG video encryption,” in *International Conf. on Information Communication and Embedded Systems (ICICES2014)*. S.A. Engineering College, Chennai, pp. 1–5, 2014.

[18] P. M. Chanal and M. S. Kakkasageri, “Hybrid algorithm for data confidentiality in internet of things,” in *10th Int. Conf. on Computing, Communication and Networking Technologies (ICCCNT)*. IEEE, pp. 1–5, 2019.

[19] G. Singh, “A paper of encryption algorithms (RSA, DES, 3DES and AES) for information security,” *International Journal of Computer Applications*, vol. 67, no. 19, 2013.

[20] N. Drucker, S. Gueron and V. Krasnov, “Making AES great again: The forthcoming vectorized AES instruction,” in *16th Int. Conf. on Information Technology–New Generations (ITNG 2019)*, Cham: Springer, pp. 37–41, 2019.

[21] M. M. Wong, M. L. D. Wong, C. Zhang and I. Hijazin, *Circuit and system design for optimal lightweight AES encryption on FPGA*. Singapore: Nanyang Technological University, 2018.

[22] P. Dixit, A. K. Gupta, M. C. Trivedi and V. K. Yadav, “Traditional and hybrid encryption techniques: A survey,” in *Networking Communication and Data Knowledge Engineering*, pp. 239–248, 2018.

[23] A. Ibrahim and G. Dalkiliç, “An advanced encryption standard powered mutual authentication protocol based on elliptic curve cryptography for RFID, proven on WISP,” *Journal of Sensors*, vol. 2017, pp. 1–10, 2017.

[24] S. Panghal, S. Kumar and N. Kumar, “Enhanced security of data using image steganography and AES encryption technique,” *International Journal of Computer Applications*, vol. 42, 2016.

[25] M. A. Albahar, O. Olawumi, K. Haataja and P. Toivanen, “Novel hybrid encryption algorithm based on AES, RSA, and twofish for bluetooth encryption,” *Journal of Information Security*, vol. 09, no. 02, pp. 168–176, 2012.

[26] P. Kumar and S. B. Rana, “Development of modified AES algorithm for data security,” *Optik (Stuttg)*, vol. 127, no. 4, pp. 2341–2345, 2016.

[27] Q. Zhang and A. Qunding, “Digital image encryption based on advanced encryption standard (AES) algorithm,” in *Proc.—5th Int. Conf. on Instrumentation and Measurement, Computer, Communication, and Control, IMCCC 2015*, pp. 1218–1221, 2016.

[28] M. A. Khan and K. Salah, “IoT security: review, blockchain solutions, and open challenges,” *Future Generation Computer Systems*, vol. 82, no. 15, pp. 395–411, 2018.

[29] L. Cardelli, “Two-domain DNA strand displacement,” *Electronic Proceedings in Theoretical Computer Science*, vol. 26, pp. 47–61, 2010.

[30] S. Namasudra and G. C. Deka, “*Advances of DNA computing in cryptography*,” in *Advances of DNA computing in cryptography*, 2018.

[31] M. Elhoseny, G. Ramírez-González, O. M. Abu-Elnasr, S. A. Shawkat, N. Arunkumar *et al.,* “Secure medical data transmission model for IoT-based healthcare systems,” *IEEE Access*, vol. 6, pp. 20596–20608, 2018.

[32] H. M. El Hennawy, A. E. Omar and S. M. Kholaif, “Design of LEA: Link encryption algorithm new proposed stream cipher algorithm,” in *31st National Radio Science Conference (NRSC)*, IEEE, pp. 82–91, 2014.

[33] J. Wu, X. Liao and B. Yang, “Image encryption using 2D Hénon-Sine map and DNA approach,” *Signal Processing*, vol. 153, no. 7, pp. 11–23, 2018.

[34] X. J. Tong, M. Zhang, Z. Wang and J. Ma, “A joint color image encryption and compression scheme based on hyper-chaotic system,” *Nonlinear Dynamics*, vol. 84, no. 4, pp. 2333–2356, 2016.

[35] T. Xiang, K. W. Wong and X. Liao, “Selective image encryption using a spatiotemporal chaotic system,” *Chaos An Interdisciplinary Journal of Nonlinear Science*, vol. 17, no. 2, pp. 023115, 2007.

[36] R. Enayatifar, A. H. Abdullah, I. F. Isnin, A. Altameem and M. Lee, “Image encryption using a synchronous permutation-diffusion technique,” *Optics and Lasers in Engineering*, vol. 90, pp. 146–154, 2017.