Healthcare Data Security in Cloud Storage Using Light Weight Symmetric Key Algorithm

Vimala Devi Parthasarathy
Department of Computer Science and Engineering,
SASTRA Deemed to be University, India
vimalasarathi@src.sastra.edu

Kalaichelvi Visvalingam
Department of Computer Science and Engineering,
SASTRA Deemed to be University, India
kalaichelvi2k@src.sastra.edu

Abstract: *Now-a-days, healthcare monitoring system is very much important system in the medical field to know the patient‘s health status immediately. In the proposed system, the sensors are fixed over the patient's body or placed at some distances around the body to collect patient's significant parameters like blood pressure, temperature, heart beat rate, etc. These parameters are collected by the healthcare professionals through some connectivity mechanisms like Bluetooth, ZigBee, etc. These significant data will be outsourced to the cloud storage in a secure way to avoid attack from the attackers. So, we need some protection mechanism to safeguard this information. This article proposes a light weight cryptographic algorithm (symmetric key) via random number key generation using Diffie-Hellman key exchange based on Elliptic Curve (ECDH) cryptography. As a result of substituting bytes (S-box) and folding (horizontal and vertical) operations, the proposed symmetric key algorithm achieves the foremost properties of cryptography such as confusion and diffusion quite well. Experimental results showed that the overall execution time of the proposed algorithm is superior to the standard Advanced Encryption Standard (AES) algorithm. The throughput rate of the proposed algorithm is 20.525095 KB/seconds whereas for the standard AES algorithm throughput rate is 18.727215 KB/seconds. So, the proposed algorithm is faster than the existing AES algorithm. Moreover, the construction of S-box, IS-box and the key generation procedures are entirely different in the proposed algorithm so, it increases the complexity for the attacker and it will create confusion to the attacker.*

Keywords: *Healthcare, RNG, ECDH, AES, cryptography.*

Received January 30, 2023; accepted June 12, 2023
https://doi.org/10.34028/iajit/21/1/5

1. Introduction

In recent years, the fast-growing Internet of Things (IoT) technology become popular in the field of healthcare monitoring. Wireless Body Area Networks (WBANs) is a kind of network used in healthcare field where the sensors are implanted over the human body (contact) or fixed at some distances around the human body (contactless) to collect vital symptoms of patients like temperature, heart rate, breathing rate, blood pressure, etc. IoT devices or sensors are resource constrained devices. The sensors will collect information from the patient's body over a period of time [24, 28]. Due to storage limitations, the healthcare professionals will gather these patient's information through some connectivity mechanism (ZigBee, Bluetooth, Wi-Fi, etc.,) and then it is outsourced (shown in Figure 1) to the cloud storage [3, 8, 14, 15, 25, 27]. To gain commercial or economic benefits, the attacker can disclose patients' sensitive information to organizations. Apart from these attacks, eavesdropping, collusion, impersonation is also possible during transmission. Obviously, any attack may happen to the collected information, alterations of patient's health information may lead to life threatening risk to the patient. Therefore, many researchers and scientists recommend to encrypt the healthcare data before it is outsourced to the cloud storage to provide confidentiality. The stored data is at risk, when the unauthorized person tries to change the data. So, security, privacy and integrity are the challenges in the field of healthcare monitoring system [1, 4, 6, 17]. These challenges can be achieved through cryptography technology [5]. Cryptography provides three important services such as Confidentiality Integrity and Authentication (CIA).

- *Confidentiality*: protects the data from unauthorized usage by means of encryption.
- *Integrity*: it verifies the correctness of the data.
- *Authentication*: it verifies the identification of communicating entity.

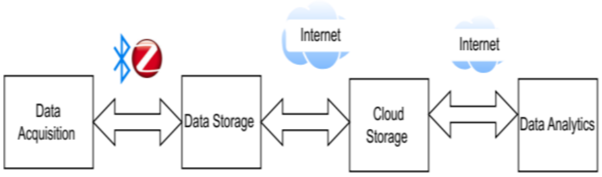


Figure 1. Phases involved in healthcare system.

Asymmetric key cryptography and symmetric key cryptography are two forms of cryptography. Confidentiality can be easily achieved through symmetric key cryptography but both confidentiality and authentication can be achieved using asymmetric key cryptosystem. Many algorithms are published in symmetric key cryptosystem such as Data Encryption Standard (DES), double DES, triple DES, Blowfish, IDEA, RC4, etc. Similarly, many algorithms are published in asymmetric key cryptosystem namely Rivest-Shamir-Adleman (RSA), Elliptic Curve Cryptography (ECC), ElGamal, Knapsack, etc. In addition, symmetric key cryptosystems are faster than asymmetric key cryptosystems, but key management in symmetric key cryptosystems is more challenging. The majority of existing schemes are characterized by high encryption, decryption, and key generation times. So, this article combines both symmetric key cryptography for encryption and asymmetric key cryptography for key exchange and key generation to reduce the computation time.

The main contributions of our work as follows:

1. A novel symmetric key encryption/decryption is proposed in this article for securing healthcare data.
2. In the proposed symmetric key algorithm, the main properties of cryptography such as confusion and diffusion are achieved well through substitute byte (S-box) operation, folding (horizontal and vertical) and add round key operations.
3. In the proposed algorithm, the keys are generated using Diffie-Hellman key exchange based on Elliptic Curve (ECDH) algorithm and new Random Number Generator (RNG) algorithm.
4. The proposed scheme takes less computation time for key generation, encryption, decryption and overall execution.

The article is organized as follows, section 2 covers some research works relevant to the proposed systems. Section 3 describes about the preliminary studies. Section 4 explains about the proposed encryption/decryption scheme. Section 5 presents the results and discussions. Finally, section 6 gives the conclusion.

2. Related Works

This section describes about the various methodologies used in healthcare monitoring system to provide security to the significant parameters like temperature, blood pressure, heartbeat rate, etc., Thabit *et al*. [23] proposed an algorithm to enrich data security in cloud computing applications via lightweight cryptography. The algorithm processes 16-bytes block of data using 16-bytes of key to encrypt the data. It employs logical operations such as XOR, shifts and swaps and XNOR to achieve Shannon's philosophy such as diffusion and confusion. Kumar and Rana [16] proposed modified Advanced Encryption Standard (AES) algorithm for providing security by increasing the number of rounds from 10 to 16 which leads to more security to the system. Polybius square method is used for generating the key. When the number of rounds is increased, the hacker will have to spend more time computing to break the system and the system will be more difficult to hack.

Msolli *et al*. [18] presented the new security approach called Shift-AES algorithm for wireless multimedia sensor network by replacing mix-column operation. The experiment results show that Shift-AES provides a minimum execution time for HD images and the best security results for all modes of encryption. SubBytes and ShiftRows transformations have been modified to enhance the AES algorithm. Round key dependence is modified in the SubBytes transformation, while randomization is used in the ShiftRows transformation. Based on experimental results obtained with the enhanced AES algorithm, the computation time is slightly longer than with the conventional AES algorithm. Even then, strong avalanche effect is achieved in the enhanced AES algorithm than the conventional AES algorithm has been reported by Abikoye *et al*. [2].

Zhang *et al*. [29] concluded that rectangle cryptosystem is optimized for 64-bit blocks with one of two key lengths, 80 bits or 128 bits, with only 25 rounds. Rectangle algorithm is based on Substitution-Permutation Network (SPN). A Stable IoT (SIT) light-weight encryption algorithm was developed by Usman *et al*. [26]. Eight bytes address is needed in order to encrypt the 64.bit data. Feistel and a uniform substitution permutation network make up the architecture of the proposed algorithm. In just five rounds of encryption, the algorithm provides significant security, according to simulations.

A cloud-based and cryptographic health monitoring scheme based on IoT sensors was proposed by Hu *et al*. [12]. A growing elderly population can benefit from embedded devices with cloud servers as they are able to provide more flexible services without the need to visit a hospital. Sensor-cloud models offer a variety of advantages, but they also present security vulnerabilities. In order to ensure the elderly's privacy, it is therefore necessary to design and integrate security issues, such as authentication and data confidentiality. Authentication and security are achieved with the proposed scheme. Robinson *et al*. [20] designed a healthcare system using cloud computing and IoT devices. Since the patient's data are stored in cloud, physicians were better able to solve their health problems on time because they had easier access to data needed to monitor patients' health. Gadde *et al*. [10] introduced Deoxyribonucleic Acid (DNA)-based modified ECC and robust S-box-based AES to ensure integrity and confidentiality of medical data. First the data were compressed using Improved Huffman coding and then encrypted. This algorithm helps secure communication in cloud. Munjal and Bhatia [19] compared two homomorphic algorithms, Paillier homomorphic and RSA for secure processing of data in cloud. The algorithms were examined based on homomorphic property, base method and efficiency and the results show that both are efficient in terms of privacy but result in additional overhead in terms of processing and communication.

Three different kinds of communication channels used by Chattopadhyay *et al*. [7] to design a secure IoT based healthcare such as:

1) From sensor nodes to Processing Unit (PU).
2) From the PU to the Gateway.
3) From the Gateway to the cloud storage.

Gia *et al*. [11] proposed a Low-Cost Health Monitoring model (LCHM) for collecting health information of various cardiac patients. The sensor nodes monitor and analyse Electrocardiograms (ECGs) in real time to efficiently process data from cardiac patients. In addition, sensor nodes collect respiration rate, and body temperature and transmit them in wireless communication mode to intelligent gateways to quickly make automated decisions to assist patients. A small test bed based on orange Pi one was used to test the performance of the LCHM model in terms of run time, but LCHM consumes more power during data acquisition and transmission.

Data generated by IoT devices can sometimes be difficult to store because of a lack of storage space. As a result, many researchers and organizations store their data in cloud environment. A cloud service provider's security strategy determines the security of data in the cloud. Suganya and Dhamodharan [22] explained the methods for data security in cloud. The third party audit scheme is also proposed to ensure the integrity. Shewale and Sankpal [21] proposed a body sensor network utilizing IoT-enabled wireless sensor nodes on a cloud computing platform. It explores the use of wireless sensor nodes with IoT built-in to send and receive data. There are three units in the system, monitoring unit, decision making unit and medicine unit. Monitoring unit is used to collect signals form the patient and send these informations to decision making unit. Doctors available in the decision making unit determines whether the patient’s health is normal or abnormal. Based on the Doctor’s suggestion the medicine unit gives alert to the patient’s mobile. In order to protect the healthcare system and maintain patient confidentiality, they considered privacy and security protocols. Islam *et al*. [13] designed e-Health systems with four steps architecture consisting of devices, data aggregation and processing, data storage and data analysis. They discussed the benefits and limitations of the system, as well as applications for which it would be useful.

3. Methodology

3.1. System Architecture

Figure 2 shows the overview of proposed system architecture. In the proposed system, sensors are implanted over the human body or placed at some distances around the human body (contactless sensors) to collect physiological parameters such as temperature, blood pressure, heart beat rate, etc., These physiological parameters are gathered by healthcare professional through some connectivity mechanisms like Bluetooth, ZigBee, etc., and then it is outsourced to the cloud storage. These parameters are very sensitive and these parameters may be attacked by the attackers while outsourcing into the cloud storage. So, we need some protection mechanisms for safeguarding theses sensitive parameters. Actually, security mechanisms are required in two main stages such as device security, communication security and storage security. Device security is that providing security in the device (sensors, IoT devices, etc.,) itself. In this article, we are concentrating only communication and storage security. Communication and storage security is that providing security to the data while in transit and at the storage level. In the proposed system, the healthcare professionals will gather vital parameters from the various sensors through some connectivity mechanism and then after applying some security mechanism these data are outsourced to the cloud storage. Then, the authorized expert/doctor will access the patient health information from the cloud storage and it will be analysed and the status will be informed to the patient through the healthcare professionals.

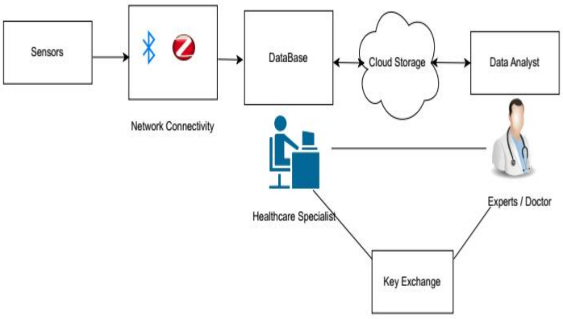


Figure 2. Proposed system architecture.

3.2. Key Exchange and Generation

The secret key is shared between the healthcare specialist and experts/doctor using ECDH key exchange and RNG. In this, the publicly known values are the elliptic curve equation Ep(a, b) and the generator or base point *G*. The Healthcare specialist will choose a secret key (*nA*) using RNG and the he will calculate his public key (*PA*) using Equation (1).

$$PA = nA * G \tag{1}$$

Similarly, the Expert/Doctor who is going to analyse the healthcare data will calculate private key (*nB*) and public key (*PB*) using Equation (2).

$$PB = nB * G \tag{2}$$

Then, both parties will share their public key with each other. Finally, both parties will calculate the shared secret key K_{AB} using other user's public key and their own private key which is illustrated in Figure 3. K_{AB} is a common point which has X-coordinate and Y-coordinate. These two coordinates will be the seed value for RNG() process. Then, both parties will generate the secret key which is used for encryption and decryption using Algorithms (1) and (2).

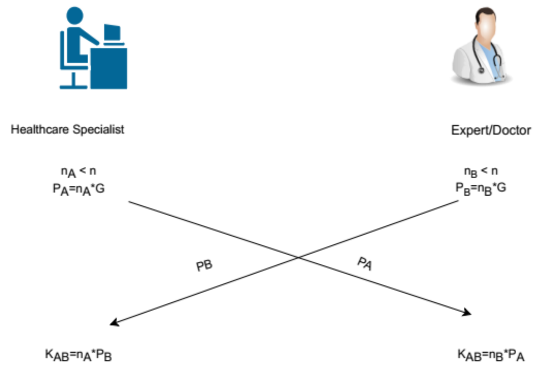


Figure 3. Key exchange between healthcare specialist and expert using ECDH.

Algorithm 1: Pseudo-code for Random Number Generator [RNG()]

```
Input: X0=Point X     Y0=Point Y
Output: Random Number
For i = 1 to 16do
    Ri =(Xi-1)2 mod 100
    Yi =(Yi-1) + i
    X i=Ri + Yi
   End For
```

Algorithm 2: Key Generation

Input: Elliptic curve parameters a, b, pn and Generator point G
Output: Generation of 16-bytes
1.Sender A will choose a private key nA where nA<n and he calculate
2.Similarly, Receiver B will choose his own private key nB where nB<n and he will calculate his own public key
3.Each user will inform their public key to other user
4.Then, Each one will calculate one common key(KAB) using their own private key and public key of other user i.e., AB (Point X,Point Y)
5. Call Function RNG()to generate 16 bytes.

- **Illustration**

Consider the elliptic curve equation $E_5(1, 1)$ and the generator point G= (0, 4)

1) Assume user A selects a random integer nA=2 and calculates his public key PA=2(0, 4) = (4, 3)
2) Similarly, user B selects a random integer nB=3 and calculates his public key PB=3(0, 4) = (2,4)
3) Then, both A and B will calculate the common KAB=2(2, 4) =3(4,3) = (2,1)

This common point (2, 1) is an input for RNG() process. Take X0=2 and Y0=1

- If i=1, R1=2^2 mod 100=4; Y1=1+1 =2; X1=4+2=6
- If i=2, R2=6^2 mod 100=36; Y2=2+2=4; 2=36+4=40

Like this, the first round 16-bytes sub-key {4, 36, 0, 49, 0, 56, 84, 69, 36, 24, 0, 29, 24, 76, 64, 25} are generated. The same procedure will be used for generating sub-keys for all the rounds.

3.3. SFX Encryption/Decryption

The proposed SFX encryption/decryption system consists of ten rounds and all rounds follow the same structure given in Figure 4. Algorithm (3) will process 128-bit at a time.

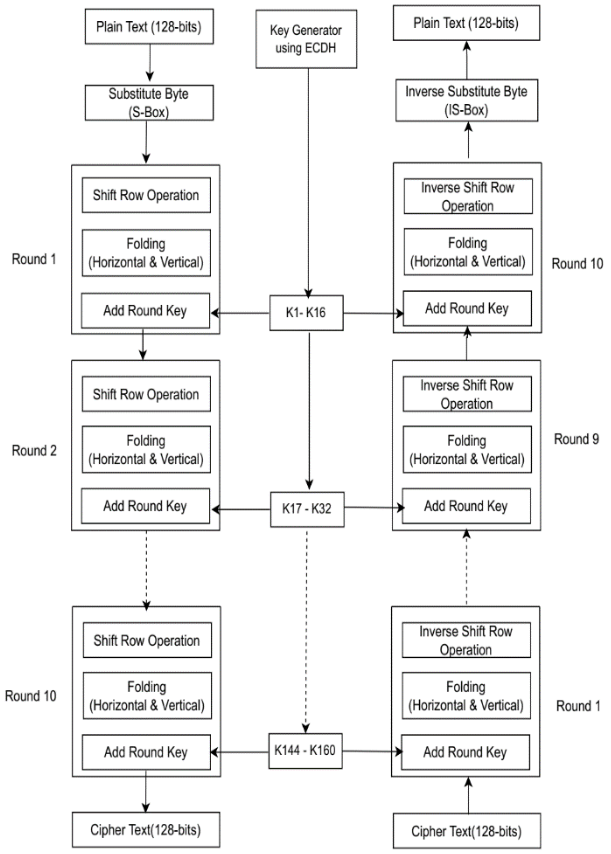


Figure 4. Proposed SFX encryption/decryption scheme.

Algorithm 3: Encryption

```
Input: 128-bit Plain text
Output: 128-bit Cipher text
1.Divide the message into blocks of 128-bits
2.Apply S-box operation
3. Then, it passes through ten rounds of operations. It consists
of three steps. They are given as follows:
For Round = 1 to 10 do
Step 1: Apply shift row operation for the output of s-box
operation
Step 2: Apply horizontal and vertical folding for output of
step1.
Step 3: Perform XOR operation with the key values
          End For
```

First, the substitute byte operation (S-box) is applied to the plain text then it passes through ten rounds. Each round consists of the following three steps such as Shift row operation, folding (horizontal and vertical) and add round key operations. The proposed Algorithms (1) and (2) will generate different keys for different rounds, including previous round keys. At the end of tenth round, it will produce 128-bit value as the cipher text.

At the decryption side, the same process should be performed on the received cipher text but using the key value in reverse order to obtain plain text. The steps are explained in Algorithm (4). Now, we will see how the plain text is processed in the forthcoming sections.

Algorithm 4: Decryption

```
Input: 128-bit Cipher text
Output: 128-bit Plain text
1. Divide the message into blocks of 128-bits
2. Then, it passes through ten rounds of operations. It consists of three steps.
  For Round = 1 to 10 do
  Step 1: Perform XOR operation with the key values
  Step 2: Apply horizontal and vertical folding for output of step1.
Step 3: Apply Inverse shift row operation for the output of s-box operation
          End For
```

3.3.1. Substitute Byte Operations

In the proposed system, we are considering the following 96 printable characters [given in Table 1] not all the 256 ASCII characters. The characters 0-31 and 128-256 in ASCII table are control characters and extended ASCII codes, both are represented by symbols. Misinterpretation of these symbols, results wrong encoding. So we have chosen 96 printable characters for our work. Based on the type, the entire 96 characters are partitioned into 20 blocks [Numbered from 0 to 19].

Table 1. List of printable ASCII characters.

Block number	Characters	No. of characters	ASCII value
0	Blank space ! “ #	4	32-35
1	$ % & ‘	4	36-39
2	( ) * +	4	40-43
3	, - . /	4	44-47
4	0 1 2 3 4	5	48-52
5	5 6 7 8 9	5	53-57
6	: ; < =	4	58-61
7	> ? @	3	62-64
8	A B C D E F G	7	65-71
9	H I J K L M	6	72-77
10	N O P Q R S T	7	78-84
11	U V W X Y Z	6	85-90
12	[\ ]	3	91-93
13	^ _ `	3	94-96
14	a b c d e f g	7	97-103
15	h i j k l m	6	104-109
16	n o p q r s t	7	110-116
17	u v w x y z	6	117-122
18	{ | }	3	123-125
19	~ del	2	126-127
Total		**96**	

For constructing S-box, we have taken a typical key K0 as follows numbered from 00 to 19 (since there are 20 blocks) without duplication. In the table row indicates the left value and column indicates right value of the ASCII number. The printable characters start from 32, so the row values start from 3. Based on the key value K0, the S-box table [given in Table 2] is constructed. The first key, K0, is selected at random, with no restrictions, from 0 to 20. The first and second row consist of even and odd numbers.

K0 → 00 02 04 06 08 10 12 14 16 18
01 03 05 07 09 11 13 15 17 19

For example, if the block number is 00, the corresponding ASCII values are 32, 33, 34 and 35. It is positioned from 0 to 3 in S-box table. For the block number 02, the characters are () *, +, and the ASCII values are 40, 41, 42, 43. The numbers from 4 to 7 are positioned S-box Table. Similarly, for all characters the values are fixed in S-box table. Inverse S-box [given in Table 3] is calculated only from S-box table.

Table 2. Substitution (S-box) operation.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
3			0	1	2	3	51	52	53	54
4	4	5	6	7	55	56	57	58	8	9
5	10	11	12	59	60	61	62	63	13	14
6	15	16	64	65	66	17	18	19	20	21
7	22	23	67	68	69	70	71	72	24	25
8	26	27	28	29	30	73	74	75	76	77
9	78	31	32	33	79	80	81	34	35	36
10	37	38	39	40	82	83	84	85	86	87
11	41	42	43	44	45	46	47	88	89	90
12	91	92	93	48	49	50	94	95		

In the proposed system, the plain texts are filled in 4x4 state matrix column by column. Initially, Substitute byte operation is applied on the state matrix. Now, let us consider the plain text is “NETWORK SECURITY” and the corresponding ASCII values are 78, 69, 84, 87, 79, 82, 83, 69, 67, 85, 83, 73, 84, and 89. These values are converted using S-box table. For example, consider the plain text ‘N’, the corresponding ASCII value is 78, using the S-box table it is converted to 24 (row7 and column 8). This procedure is repeated for subsequent characters in plaintext.

N	O	S	R	⇒	78	79	83	82	⇒	24	25	29	28
E	R	E	I		69	82	69	73		21	28	21	68
T	K	C	T		84	75	67	84		30	70	19	30
W	B s	U	Y		87	32	85	89		75	00	73	77

Plain text ASCII value Output of S-box

3.3.2. Shift Row Operation

Round functions begin with shift row operation which uses the output of S-box operation as input. For each row i in the S-box, shift left to i-1 times, where i is the number of rows. For example, if it is second row, each value of the second row is shifted to the left 1 time. After shifting the values in the second row once to the left, we get the following output: 28, 21, 68, and 21. The values in the third and fourth rows are similarly shifted by three and four times to left.

24	25	29	28	⇒	24	25	29	28
21	28	21	68		28	21	68	21
30	70	19	30		19	30	30	70
75	00	73	77		77	75	00	73

S-box Output Output of Shift Operations

3.3.3. Folding

The output of the shift operation is input for this step. Folding is a kind of mirror reflection. It exchanges the positions of the plain text. In this, there are two types of folding involved such as Horizontal Folding and Vertical Folding. Horizontal folding finds the mid row of the plain text, based on this the subsequent rows are exchanged. For example, in the following table the plaintext consists of four rows and the dark line defines mid row and mid column, the arrow indicate the exchange occurs. The horizontal folding is as follows,

Row 1 ⟷ Row 4
Row 2 ⟷ Row 3

Similarly, in vertical folding the columns are exchanged.

Column 1 ⟷ Column 4
Column 2 ⟷ Column 3

The values in row1 of the following table are 24, 25, 29, 28, and row4 has 77, 75, 00, and 73. The values of row1 are exchanged into row 4 and row4 into row1 using horizontal folding. The values of row1 become 77, 75, 00, 73, and row4 are 24, 25, 29, and 28 after applying horizontal folding. Likewise, the values can be switched between columns by vertical folding. This procedure is repeated for remaining rows and columns.

24	25	29	28		77	75	00	73		73	00	75	77
28	21	68	21		19	30	30	70		70	30	30	19
19	30	30	70	⇨	28	21	68	21	⇨	21	68	21	28
77	75	00	73		24	25	29	28		28	28	25	24

Out of Shift Operation | After Horizontal Folding | After Vertical Folding

3.3.4. Add Round Key

In this, 128-bit (16 bytes) key values are written in 4 x 4 key matrix column by column. It simply XOR with the output of the folding operation. For every round, the XOR operation is applied to each row of folding output and key, which gives cipher text at end of the encryption process. For example, the XOR between 73 and 04 gives 77. This procedure is repeated for remaining values.

73	00	75	77		04	00	36	44		77	00	111	97
70	30	30	19	⊕	36	56	24	96	⇨	98	38	06	115
21	68	21	28		96	84	64	00		117	16	85	28
28	29	25	24		09	69	81	01		21	88	72	25

Output of Folding | Round 1 Key | After XOR operation

3.3.5. Decryption

Decryption is the reverse process of encryption. At the end of tenth round, we will get 128-bit value as the Cipher text. Then the healthcare specialist will upload the scrambled data in the cloud storage. Then, the receiver (expert/doctor) has to extract and decrypt the information using the shared key. At the receiver side, the shared key should be used in reverse order i.e., in the first round he has to use sub key K144-K160 and in the last round he has to use K1-K16. Finally, the output of tenth round is applied to the inverse substitution (IS-box) operation to obtain plain text. IS-box is the reverse form of S-box. For example, the value 117 is mapped into 88 in the S-box table and in IS-box 88 is mapped in to 117.

Table 3. Inverse substitution (IS-box) operation.

	0	1	2	3	4	5	6	7	8	9
0	32	33	34	35	40	41	42	43	48	49
1	50	51	52	58	59	60	61	65	66	67
2	68	69	70	71	78	79	80	81	82	83
3	84	91	92	93	97	98	99	100	101	102
4	103	110	111	112	113	113	115	116	123	124
5	125	36	37	38	39	44	45	46	47	53
6	54	55	56	57	62	63	64	72	73	74
7	75	76	77	85	86	87	88	89	90	94
8	95	96	104	105	106	107	108	109	117	118
9	119	120	121	122	126	127				

3.3.6. Comparison between SFX and AES Algorithm

In order to show the advantages of proposed algorithm, it is compared with AES, in terms of key size, block size, Structure and number of rounds etc. The comparison between the proposed algorithm and the standard AES algorithm is given in Table 4.

Table 4. Comparison between SFX and AES algorithm in terms of architecture.

Terms	AES algorithm	Proposed algorithm
Structure	Substitution-Permutation	Substitution-Permutation
Block size	128-bits	128-bits
Key size	128, 192, 256 bits	128-bits
Number of rounds	10,12,14	10
Possible key	2^{128}, 2^{192}, 2^{256}	2^{128}
Mathematical operations	XOR, mixing, substitution, multiplication, shifting and addition	Substitution, shifting, mixing and XOR
S-box size	16 x 16	10 x 10
Key generation	S-box operation, shifting and XOR operation	Using ECDH key exchange and RNG

4. Results and Discussions

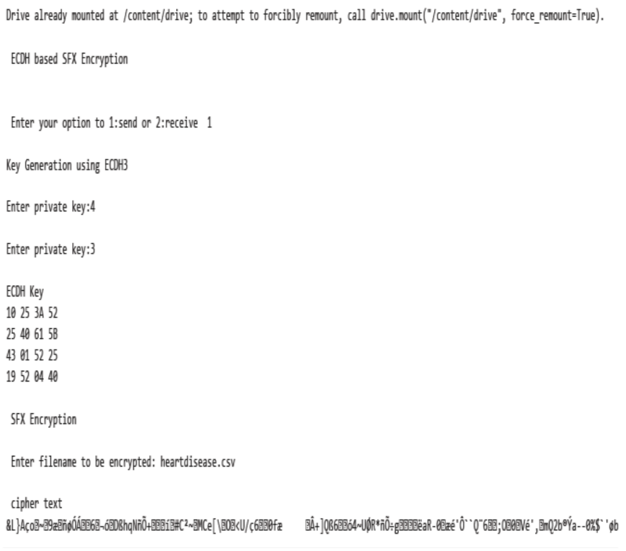
4.1. Experimental Setup

We have implemented our proposed algorithm on a laptop with the following configurations like Intel Core-2-Duo, CPU, @, 2.5, GHz, 4GB, RAM and Windows 10, with 64-bit OS. In our work, we have used the following values and variables for implementing our proposed algorithm which are given in Table 5.

Table 5. Variables and initialized values.

Key variables	Initialized values	Description
(a, b, p)	(1,1,5)	ECC Parameters
G	(0,4)	Generator or Base Point
nA	2	User A’s Private key
nB	3	User B’s Private key
PA	(4, 3)	User A’s Public key
PB	(2, 4)	User B’s Public key
K_{AB}	(2, 1)	Shared key

The implementation of the proposed encryption using Table 5 and Algorithm (3) is shown in Figure 5.



```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

 ECDH based SFX Encryption


 Enter your option to 1:send or 2:receive  1

Key Generation using ECDH3

Enter private key:4

Enter private key:3

ECDH Key
10 25 3A 52
25 40 61 5B
43 01 52 25
19 52 04 40

 SFX Encryption

 Enter filename to be encrypted: heartdisease.csv

 cipher text
&L}Aço▯~▯9æ▯ñøÓÁ▯▯6▯-ó▯D8hqNñÕ+▯▯▯i▯#C²~▯MCe[\▯O▯<U/ç6▯▯0fæ     ▯Â+]Qß6▯▯ó4~UØR*ñÕ÷g▯▯▯▯ëaR-0▯æé'Ô``Q"6▯▯;O▯0▯Vé',▯mQ2b®Ýa--0%$`'øb
```

Figure 5. Implementation of SFX encryption.

4.2. Dataset Collection

We have taken health record dataset for testing from the Cleveland database available in UCI repository [9]. The dataset is also used to classify whether the person is suffering from heart disease or not. A total of 14 attributes are included in this study such as age, sex, chest pain type, resting blood pressure, serum cholesterol, fasting blood sugar, electrocardiographic results at rest, maximum heart rate achieved, exercise induced angina, old peak, number of major vessels, and Thalassemia. There are 303 records in the dataset, with 13 input attributes and one output variable called target.

4.3. Experimental Tests

Many algorithms were published under symmetric key algorithms like DES, 2-DES, 3-DES, IDEA, Blowfish, AES etc. Many researchers are recommending AES algorithm for secure communication. So, we have tested our proposed algorithm with various file sizes that is ranged from 1KB to 1024 KB and compared with only existing standard AES algorithm. The data set used for testing is divided into different file sizes, consists of textual information and taken for analysis. Various performance measures such as Execution Time for encryption, decryption, overall execution time and throughput rate have been measured which are given in Tables 6 and 7 and in Figures 6, 7, 8, and 9.

Table 6. Comparison of encryption and decryption process between AES and proposed algorithm.

File Size (KB)	Encryption Execution Time (Seconds)		Decryption Execution Time (Seconds)	
	AES	Proposed	AES	Proposed
1 KB	0.013003	0.0001	0.014002	0.015626
5 KB	0.050021	0.031244	0.053013	0.031252
10 KB	0.14804	0.140627	0.100021	0.140638
100 KB	1.017256	0.906296	0.953236	0.703169
500 KB	4.96925	4.453416	5.028264	3.672115
1024 KB	14.264633	7.153923	11.761057	9.320138

Table 7. Overall execution time comparison between AES and proposed SFX algorithm.

File size (KB)	Overall execution time (Seconds)	
	AES	Proposed
1 KB	4.215763	5.853938
5 KB	4.097121	7.400573
10 KB	9.898433	11.500364
100 KB	8.006042	11.422093
500 KB	22.163837	17.063027
1024 KB	39.191128	26.661218
Total execution time	**87.572324**	**79.90121**
Average execution time	**14.595387**	**13.316869**
Throughput (KB/Sec)	**18.727149**	**20.525095**

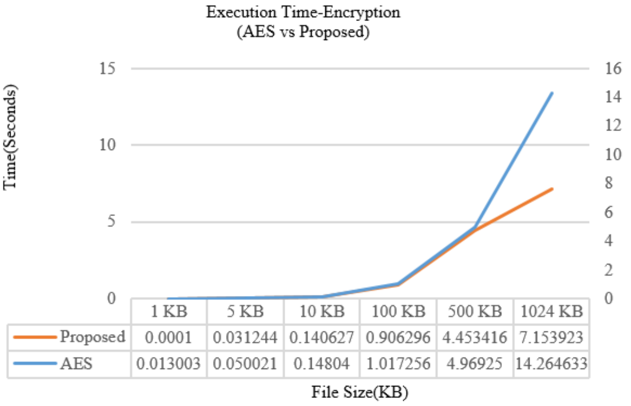


Figure 6. Computation time of encryption process between AES and proposed SFX.

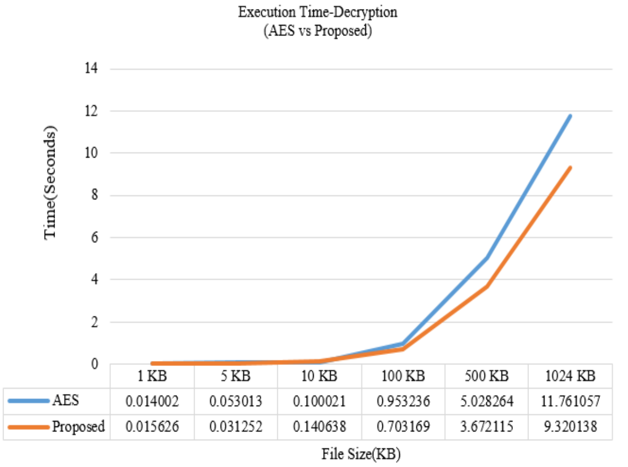


Figure 7. Computation time of decryption process between AES and proposed SFX.

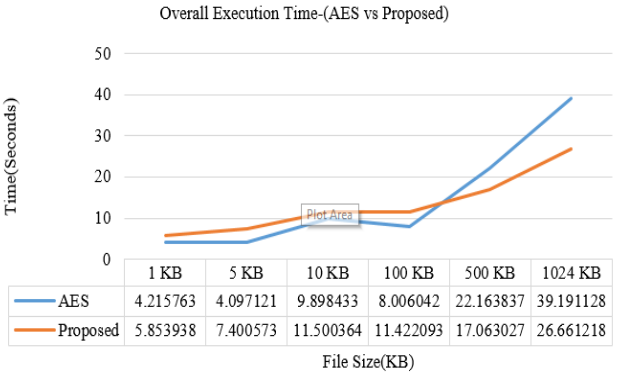


Figure 8. Overall computation time analysis between AES and proposed SFX algorithm.

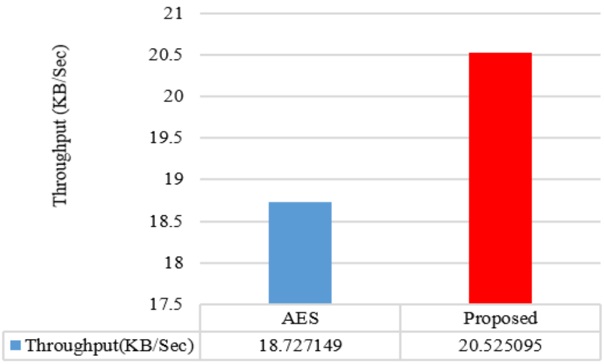


Figure 9. Throughput rate analysis between AES and proposed SFX algorithm.

4.4. Execution Time

The execution time of encryption and decryption process is an important factor to analyse the performance of an algorithm. Various file sizes have taken to measure the execution time of an encryption/decryption process and then compared with the standard AES algorithm. Table 6 shows the comparison between proposed algorithm and standard AES algorithm in terms of execution time for both encryption and decryption, which is also depicted in Figures 5 and 6. The x axis defines file size (KB) and y axis defines time (seconds). The proposed algorithm completes the execution process by 0.0001 and 0.015626 seconds for 1 KB file, whereas the AES algorithm takes 0.013003, 0.014002 seconds. Similarly, for 1024 KB the proposed algorithm uses 7.153923 and 9.320138 seconds to complete. The AES completes the process by 14.264633 seconds. From the analysis it is observed that the proposed algorithm is faster than the AES.The overall execution time include all the steps such as key generation process, encryption and decryption process which is also given in Table 7.

4.5. Throughput

The throughput rate only determines the speed of any cryptographic algorithm. It is measured using the following equation.

$$Throughput = \frac{Total\ file\ size}{Total\ execution\ time} \quad (3)$$

The throughput rate of proposed algorithm and the standard AES algorithm is shown in Table 7 and Figure 7. The throughput rate for proposed algorithm is 20.525095, which is greater than the AES.

- **Inference**

From the results, the following inferences are derived:

1. Proposed SFX algorithm takes lesser time to perform encryption/decryption process than the standard AES algorithm.
2. The overall execution time (key generation process, encryption and decryption process) of the proposed algorithm is better than the standard AES algorithm.
3. As compared to standard AES algorithm throughput rate of 18.727215 KB/seconds, the proposed algorithm has a throughput rate of 20.525095 KB/seconds. So, the proposed algorithm is faster than the existing AES algorithm.
4. In the proposed algorithm, Construction of S-box is entirely different. So, it increases the complexity for the attacker.
5. Keys are generated using ECDH key exchange algorithm and new RNG. So, it will be very difficult for the attacker to guess all keys.
6. In the proposed SFX algorithm, the sub-keys are generated using ECDH Key exchange and RNG() process but the sub-keys are generated using complex operations in the traditional AES algorithm. Moreover, the proposed SFX algorithm uses substitute byte (S-box) and inverse substitute byte (IS-box) operations. These tables are constructed entirely using different logic. So, it will create great confusion to the hackers and it will improve the strength of the algorithm. The proposed SFX algorithm uses only simple operations whereas in the traditional AES algorithm complex operation like Mix-column operation is used. So, the proposed SFX algorithm will take less time for the computation.

5. Conclusions

The collected significant health records got encrypted using the proposed SFX algorithm before it is outsourced to the cloud storage in a secure way to avoid attack from the attackers. To provide security, this article proposed a light weight symmetric key algorithm using Diffie-Hellman key exchange based on Elliptic Curve (ECDH) cryptography and new RNG. Claude-Shannon properties such as confusion and diffusion are achieved well through substitute byte (s-box) operation, folding (horizontal and vertical) and add round key operations. It was found that the proposed algorithm achieved superior than standard AES in terms of overall execution time (key generation process, encryption, and decryption process). The throughput rate of the proposed algorithm is 20.525095 KB/seconds whereas for the standard AES algorithm throughput rate is 18.727215 KB/seconds. So, the proposed algorithm is faster than the existing AES algorithm. Moreover, the construction of S-box, IS-box and the key generation procedures are entirely different in the proposed algorithm. So, it will increase the complexity for the attacker and it will create confusion to the attacker.

Compliance with Ethical Standards

Research Data Policy and Data Availability

The dataset used in this article is referred from UCI repository. The corresponding URL is https://archive.ics.uci.edu/ml/datasets/Heart+Disease

Conflict of Interest

None of the authors have received any research grants. None of the authors have received a speaker honorarium from any company. All authors declare that none of them has any conflict of interest.

References

[1] Abba Ari A., Ngangmo O., Titouna C., Thiare O., Mohamadou A., and Gueroui A., "Enabling Privacy and Security in Cloud of Things: Architecture, Applications, Security and Privacy Challenges," *Applied Computing and Inform*atics, vol. 169, pp. 1-13, 2019. https://doi.org/10.1016/j.aci.2019.11.005

[2] Abikoye O., Haruna A., Abubakar A., Akande N., and Asani S., "Modified Advanced Encryption Standard Algorithm for Information Security," *Symmetry*, vol. 11, no. 12, pp. 1484, 2019. https://doi.org/10.3390/sym11121484

[3] Aceto G., Persico V., and Pescapé A., "Industry 4.0 and Health: Internet of Things, Big data, and Cloud Computing for Healthcare 4.0," *Journal of Industrial Information Integration*, vol. 18, pp. 100129, 2020. https://doi.org/10.1016/j.jii.2020.100129

[4] Amin R., Kumar N., Biswas G., Iqbal R., and Chang V., "A Lightweight Authentication Protocol for IoT-enabled Devices in Distributed Cloud Computing Environment," *Future Generation Computer Systems*, vol. 78, pp. 1005-1019, 2018. https://doi.org/10.1016/j.future.2016.12.028

[5] Arumugam S., "An Effective Hybrid Encryption Model using Biometric Key for Ensuring Data Security," *The International Arab Journal of Information Technology*, vol. 20, no. 5, pp. 796-807, 2023. https://doi.org/10.34028/iajit/20/5/12

[6] Cha S., Hsu T., Xiang Y., Yeh K., "Privacy Enhancing Technologies in the Internet of Things: Perspectives and Challenges," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2159-2187, 2019. DOI:10.1109/JIOT.2018.2878658

[7] Chattopadhyay A., Nag A., Ghosh D., and Chanda K., "A Secure Framework for IoT-Based Healthcare System," *in Proceedings of the International Ethical Hacking Conference*, Kolkata, pp. 383-393, 2018. https://doi.org/10.1007/978-981-13-1544-2_31

[8] Dang L., Piran M., Han D., Min K., and Moon H., "A Survey on Internet of Things and Cloud Computing for Healthcare," *Electronics*, vol. 8, no. 7, pp. 768, 2019. https://doi.org/10.3390/electronics8070768

[9] Detrano R., Janosi A., Steinbrunn W., and Pfisterer M., UCI Machine Learning Repository, Heart Disease, https://archive.ics.uci.edu/dataset/45/heart+disease, Last Visited, 2023.

[10] Gadde S., Amutharaj J., and Usha S., "A Security Model to Protect the Isolation of Medical Data in the Cloud using Hybrid Cryptography," *Journal of Information Security and Applications*, vol. 73, pp. 103412, 2023. https://doi.org/10.1016/j.jisa.2022.103412

[11] Gia T., Jiang M., Sarker V., Rahmani A., Westerlund T., Liljeberg P., and Tenhunen H., "Low-Cost Fog-assisted Health-Care IoT System with Energy Efficient Sensor Nodes," *in Proceedings of the 13th International Wireless Communications and Mobile Computing Conference*, Valencia, pp. 1765-1770, 2017. DOI:10.1109/IWCMC.2017.7986551

[12] Hu J., Chen C., Fan C., and Wang K., "An Intelligent and Secure Health Monitoring Scheme Using IoT Sensor Based on Cloud Computing," *Journal of Sensors*, vol. 2017, pp. 1-11, 2017. https://doi.org/10.1155/2017/3734764

[13] Islam M., Humaira F., and Nur F., "Healthcare Applications in IoT," *Global Journal of Medical Research-B Pharma, Drug Discovery, Toxicology and Medicine*, vol. 20, pp. 21-23, 2020. https://globaljournals.org/GJMR_Volume20/E-Journal_GJMR_(B)_Vol_20_Issue_2.pdf

[14] Ismail Y., *Internet of Things (IoT) for Automated and Smart Applications*, IntechOpen, 2019. http://dx.doi.org/10.5772/intechopen.90022

[15] Khatoon N., Roy S., and Pranav P., *Intelligent Systems Reference Library*, Springer Nature, 2020. https://doi.org/10.1007/978-3-030-39119-5_6

[16] Kumar P. and Rana S., "Development of Modified AES algorithm for Data Security," *Optik*, vol. 127, no. 4, pp. 2341-2345, 2016. https://doi.org/10.1016/j.ijleo.2015.11.188

[17] Maksimovi´c M., "Improving Computing Issues in the Internet of Things Driven E-Health Systems," *in Proceedings of the International Conference for Young Researchers in Informatics, Mathematics, and Engineering*, Kaunas, pp. 14-17, 2017. https://ceur-ws.org/#Vol-1852

[18] Msolli A., Helali A., and Maaref H., "New Security Approach in Real-time Wireless Multimedia Sensor Networks," *Computers and Electrical Engineering*, vol. 72, pp. 910-925, 2018. https://doi.org/10.1016/j.compeleceng.2018.01.016

[19] Munjal K. and Bhatia R., "Analysing RSA and PAILLIER Homomorphic Property for Security in Cloud," *Procedia Computer Science*, vol. 215, pp. 240-246, 2022. https://doi.org/10.1016/j.procs.2022.12.027

[20] Robinson Y., Presskila X., and Lawrence T., *Intelligent Systems Reference Library*, Springer

Nature, 2020. https://doi.org/10.1007/978-3-030-39119-5_3

[21] Shewale A. and Sankpal S., "IOT and Raspberry Pi based Smart and Secure Health Care System using BSN," *International Journal for Research in Applied Science and Engineering Technology*, vol. 8, no. II, pp. 506-510, 2020. DOI:10.22214/ijraset.2020.2077

[22] Suganya S. and Dhamodharan P., "Enhancing Security for Storage Services in Cloud Computing," *in Proceedings of the International Conference on Current Trends in Engineering and Technology*, Coimbatore, pp. 396-398, 2013. DOI:10.1109/ICCTET.2013.6675995

[23] Thabit F., Alhomdy S., Al-Ahdal A., and Jagtap S., "A New Lightweight Cryptographic Algorithm for Enhancing Data Security in Cloud Computing," *Global Transitions Proceedings*, vol. 2, no. 1, pp. 91-99, 2021. https://doi.org/10.1016/j.gltp.2021.01.013

[24] Tuli S., Basumatary N., Gill S., Kahani M., Arya R., Wander G., and Buyya R., "HealthFog: An Ensemble Deep Learning-based Smart Healthcare System for Automatic Diagnosis of Heart Diseases in Integrated IoT and Fog Computing Environments," *Future Generation Computer Syst*ems, vol. 104, pp. 187-200, 2020. https://doi.org/10.1016/j.future.2019.10.043

[25] Tyagi S., Agarwal A., and Maheshwari P., "A Conceptual Framework for IoT-Based Healthcare System using Cloud Computing," *in Proceedings of the 6th International Conference-Cloud System and Big Data Enginee*ring, Noida, pp. 503-507, 2016. DOI:10.1109/CONFLUENCE.2016.7508172

[26] Usman M., Ahmed I., Aslam M., Khan S., and Shah U., "SIT: A Lightweight Encryption Algorithm for Secure Internet of Things," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 1, 2017. DOI:10.14569/IJACSA.2017.080151

[27] Wilt T., Versluis A., Goedhart A., Talboom-Kamp E., and Van Delft S., "General Practitioners' Attitude Towards the Use of eHealth and Online Testing in Primary Care," *Clinical eHealth*, vol. 3, pp. 16-22, 2020. https://doi.org/10.1016/j.ceh.2020.02.002

[28] Yeh K., "A Secure IoT-Based Healthcare System with Body Sensor Network," *IEEE Access*, vol. 4, pp. 10288-10299, 2016. DOI:10.1109/ACCESS.2016.2638038

[29] Zhang W., Bao Z., Lin D., Rijmen V., and BoHan Y., "Rectangle: A Bit-Slice Lightweight Block Cipher Suitable for Multiple Platforms," *Science China Information Sciences*, vol. 58, pp. 1-15, 2015. DOI: 10.1007/s11432-015-5459-7

Vimala Devi Parthasarathy received her M.E. degree in Computer and Communication Engineering from Anna University, Chennai in 2007. She is currently working as an Assistant Professor in the department of Computer Science and Engineering since 2007 at SRC, SASTRA Deemed University-Kumbakonam. Her research interests include Cryptography, Body Area networks and Cloud Computing.

Kalaichelvi Visvalingam received her M. E. degree in Computer Science and Engineering from Annamalai University, Chidamabaram in 2004 and Ph.D. degree in Information and Communication Engineering from Anna University, Chennai in 2013. She is working as an Associate Professor in the department of Computer Science and Engineering since 2004 at SRC, SASTRA Deemed University – Kumbakonam. She has published more than 23 papers in various referred journals. Her research interests include Cryptography, Steganography and security issues in various IT fields.