Research Article

AES Based Enhanced Technique with Reduced Time Complexity

Nishant Agnihotri[a], Aman Kumar Sharma[b]

[a] Ph.D. Research Scholar, Department of Computer Science, Himachal Pradesh University, Shimla, India.
[b]Professor of Computer Science, Himachal Pradesh University, Shimla, India.

Abstract: Lately, security has been emerging as the main issue for a plethora of web-based and other applications. The level of vulnerability of data is growing at two different levels, server security, and data transfer security respectively. While server security is handled by the firewall and other technologies, data transfer security is handled by using various cryptographic techniques. Symmetric data encryption and non-symmetric data encryption are two prime techniques used in cryptography. In the proposed research, an enhanced symmetric key is used for encryption purposes to ensure the integrity of data. Furthermore, this paper has proposed an enhanced AES-based algorithm that is capable of enhancing throughput without tampering with the security parameters of the existing technique. It has been found after implementation of the proposed system that the computation time for the proposed system has improved by 25% and 14% respectively for the mentioned test cases compared to the existing AES algorithm.

Keywords: Advanced Encryption Standard, Symmetric, Asymmetric, Enhanced AES, Data Security.

1. Introduction

The flow of data over the web has emerged as a new trend and it's been used to a great extent, therefore, ensuring the security of the network is a new and very important challenge. To communicate securely over an unsecured network, we mostly adopt cryptography as a solution. if we adhere to robustness, genetic algorithms are nowadays gaining quite much popularity. GA offers significant benefits over other optimization techniques.

Furthermore, changing the form of a plain text and then converting back it to its original form is called enciphering and deciphering respectively. The terms encipher and decipher are already used as standards by ISO 7498-2[**Ashish Khanna et.al**.].

1.1 Cryptographic Techniques

To ensure security adequately over a network, it becomes important to promulgate cryptographic techniques. cryptographic algorithms are mainly segregated into two basic categories namely symmetric encryption algorithms and asymmetric cryptographic algorithms. In other words, they are also known as public-key cryptographic algorithms and private key cryptographic algorithms. Figure 1 has shown a basic comparison between both techniques.

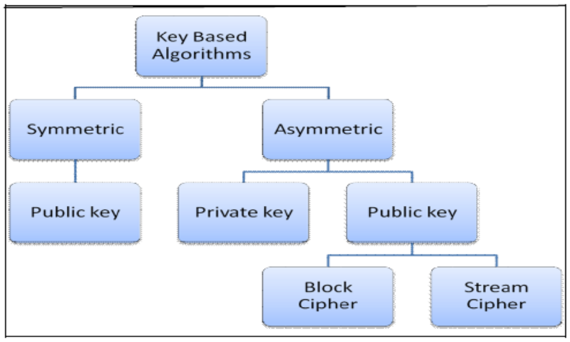


Figure. 1 Classification of Key-based Algorithms [**Ashish Khanna et.al**]

Symmetric encryption is the oldest process and has been shown in Figure 2. Subsequently, till the time sender and receiver are aware of the secret key which is a combination of any word number or merely shuffling of any string [**Communication Network Security", Vol. 3**], can encrypt and decrypt the message.

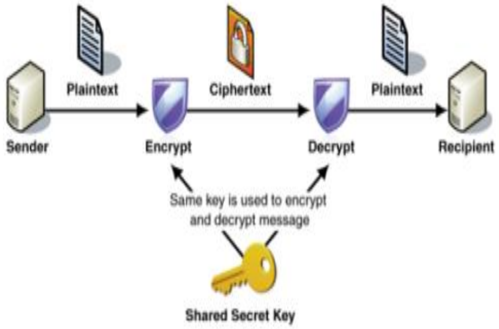


Figure. 2 (Symmetric Key) [**RizkyRiyaldhi et. Al.**]

However, the major challenge while dealing with a secret key occurs when it has to be sent over the internet while ensuring its secrecy because leakage of the key can lead to decryption of message at any point. Therefore, the sharing of keys is an extremely important phase of symmetric key cryptography [**Introduction to cryptography, ggu**].

Furthermore, to overcome this issue, public-key cryptography is another alternative where 2 keys are used. The explanation has been given in Figure 3.

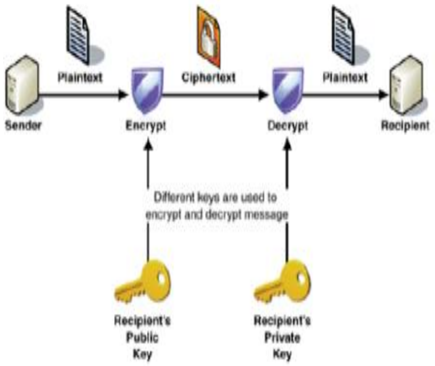


Figure.4 (Asymmetric Key)[RizkyRiyaldhi et. Al]

A public key is made freely available to anyone who might want to send you a message. A second, private key is kept secret so that only communicators have access to it. In the case of symmetric cryptography once a message is encrypted using a public key can only be decrypted by the receiver after applying the same cryptographic algorithm after applying the specific private key. However, the public key can be passed without any fear over the network, but this technique takes more time in terms of execution when encrypting and decrypting data. [**K. Ragunath Reddy et. Al**.][**Bhawna Bhatt**]. Whereas in asymmetric encryption, the public key is discovered by adhering to some technique. Most of the time digital signature is used for this purpose, which is a package of information like details of the user, details about issuing authority, public key, etc. whenever requirement of communication takes place between two parties, a query is extended to a third party which in turns return a copy of certificate followed by extraction of a key from that certificate [**Jagbir Dhillon et.al.**].

1.1.1 Data Encryption Standard

Data Encryption Standard (DES)[**Sumartono, Isnar et.al**.], falls under the category of the symmetric-key algorithm. After its adoption by the National Bureau of standards in 1997, it is popular as Federal Information Processing Standards. In DES [**RizkyRiyaldhi et. Al**], 56 bits key is divided into eight 7-bit blocks. To get an odd number an additional $8^{\text{-bit}}$ block is added to every block [**K. Ragunath Reddy et. Al**][**Jan Sher Khan et.al.**]. Since 8 bits are used just for parity checking, the effective key length is only 56 bits. After splitting bits, shifting operation takes place followed by XOR operation and then output is for expansion boxes and S-Boxes to complete encryption to assure privacy [**Xiaokun Yang et.al.**].

1.1.2 Advanced Encryption Standard

Advanced Encryption Standard (AES)[**Federal Information**] emerged into a picture due to the small key size of DES[**SomberSingh et.al.]**, not considered to be much secure, also triple DES which was its improved version of DES was slow.AES was chosen as a successor to DES after NIST's (National Institute of Standards and Technology) announcement for the requirement of AES and invitation for submissions from interested cryptographers. The process lasted from Jan 1997 to October 2000 when out of the final five submissions, 'Rijndael' was selected as the proposed AES. Rijndael is a symmetric block cipher. It can process a block size of 128 bits and AES supports a cipher key size of 128,192 and 256 bits respectively. Any other lengths of input and cipher keys are not allowed by the AES standard. Also, the size of the block of Rijndael should be multiples of 32 which allows 128 bits as a minimum, and the maximum is taken as 256[**Nentawe Y. Goshwe et.al.**][**Sandeep Singh et.al.]**. All the operations in AES are made on the state which is a rectangular array of bytes having a block size of 4X4. Row size is fixed as 4, but column size usually varies, and the column size is block size divisible by 32[**RizkyRiyaldhi et.al.**]. AES algorithm has the same length of 128 bits for input, output, and state. Also, the number of execution round for the algorithm depends on the size of keys like 10,12 and 14 for a key size of 128,192, and 256 respectively [**Ashish Khann**]. [Table 1] gives a brief description of the AES combination.

.

Table. 1. AES Description

	Key Length	**Block Size**	**Number of Rounds**
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

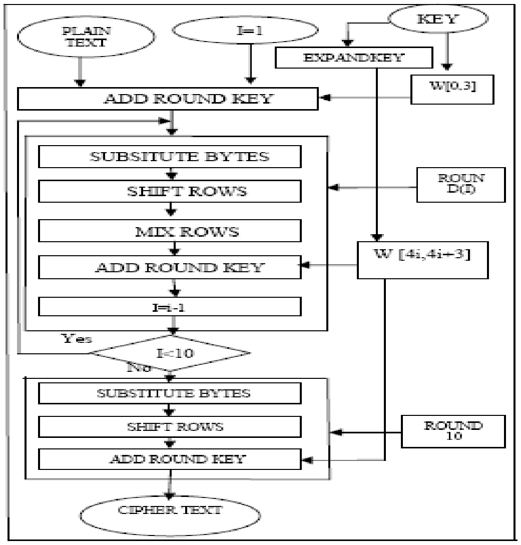


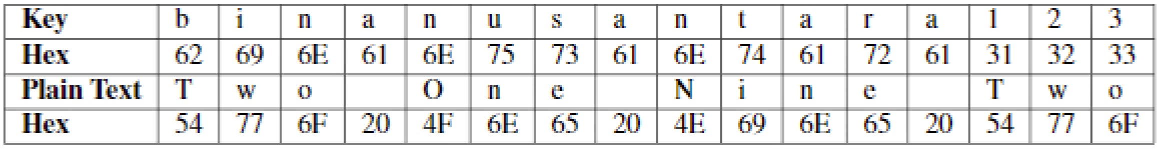
Figure. 5 Block diagram of AES encryption

2.Proposed Improved AES

2.1. Proposed algorithm

AES[**Federal Information]** key of 128 bits is specified. Each segment of the key is 16 bits in length. Characters specified in the plain text will be converted to the hexadecimal code based on a specified encoding scheme. The generated code of the plain text to be specified as the state data, which requires to be processed thereon. Table 2, shows the hex coding matrix. The table shows the value for each digit in the plain text to be replaced with hexadecimal coding.

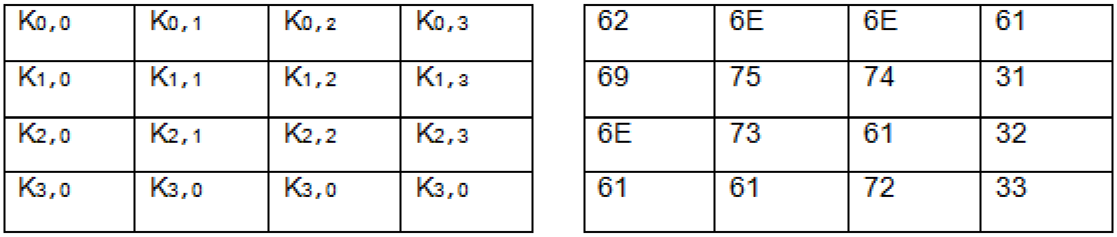
Table. 2. Hexa coding matrix



Key	b	i	n	a	n	u	s	a	n	t	a	r	a	1	2	3
Hex	62	69	6E	61	6E	75	73	61	6E	74	61	72	61	31	32	33
Plain Text	T	w	o		O	n	e		N	i	n	e		T	w	o
Hex	54	77	6F	20	4F	6E	65	20	4E	69	6E	65	20	54	77	6F

The code generated in the public key will be subdivided into a matrix of 4*4 as specified.

Table. 3. Matrix for rotation



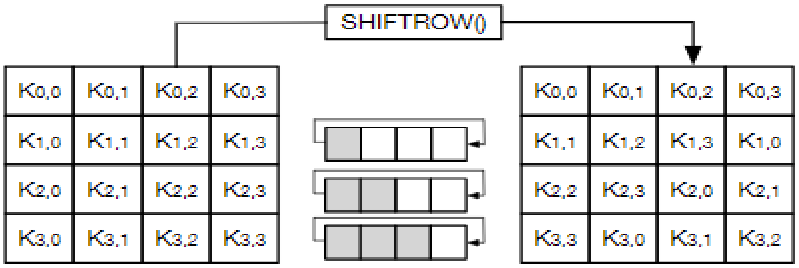
$K_{0,0}$	$K_{0,1}$	$K_{0,2}$	$K_{0,3}$
$K_{1,0}$	$K_{1,1}$	$K_{1,2}$	$K_{1,3}$
$K_{2,0}$	$K_{2,1}$	$K_{2,2}$	$K_{2,3}$
$K_{3,0}$	$K_{3,0}$	$K_{3,0}$	$K_{3,0}$

62	6E	6E	61
69	75	74	31
6E	73	61	32
61	61	72	33

[Table 3] shows the matrix for the relation of each row of the data. The plain text will be replaced with the hexadecimal coding and will be used for representing different aspects along the rows.

- The generated key will be processed with four sequential steps.
- The first step is for the conversion of the plain text to the nonlinear byte substitution using S.box generated
- The second step is for shifting each row of the matrix.
- The third step includes the dot matrix operation with the addition of the XOR.
- The fourth step includes the addition of the round key with the XORed data.

Table. 4. Rows rotation



$K_{0,0}$	$K_{0,1}$	$K_{0,2}$	$K_{0,3}$
$K_{1,0}$	$K_{1,1}$	$K_{1,2}$	$K_{1,3}$
$K_{2,0}$	$K_{2,1}$	$K_{2,2}$	$K_{2,3}$
$K_{3,0}$	$K_{3,1}$	$K_{3,2}$	$K_{3,3}$

$K_{0,0}$	$K_{0,1}$	$K_{0,2}$	$K_{0,3}$
$K_{1,1}$	$K_{1,2}$	$K_{1,3}$	$K_{1,0}$
$K_{2,2}$	$K_{2,3}$	$K_{2,0}$	$K_{2,1}$
$K_{3,3}$	$K_{3,0}$	$K_{3,1}$	$K_{3,2}$

The above table shows the row rotation for the matrix represented in [Table 4]. Each row rotation is recorded into the matrix for showing the whole data in the table or we can say matrix.

Table. 5. Multiplication matrix

$$\begin{bmatrix} 02\ 03\ 01\ 01 \\ 01\ 02\ 03\ 01 \\ 01\ 01\ 02\ 03 \\ 03\ 01\ 01\ 02 \end{bmatrix} x \begin{bmatrix} K_{0,0} & K_{0,1} & K_{0,2} & K_{0,3} \\ K_{1,1} & K_{1,2} & K_{1,3} & K_{1,0} \\ K_{2,2} & K_{2,3} & K_{2,0} & K_{2,1} \\ K_{3,3} & K_{3,0} & K_{3,1} & K_{3,2} \end{bmatrix} = \begin{bmatrix} 2*K_{0,0}+3*K_{0,1}+1*K_{0,2}+1*K_{0,3} \cdots \cdots \cdots \\ 1*K_{0,0}+2*K_{0,1}+3*K_{0,2}+1*K_{0,3} \cdots \cdots \cdots \\ 1*K_{0,0}+1*K_{0,1}+2*K_{0,2}+3*K_{0,3} \cdots \cdots \cdots \\ 3*K_{0,0}+1*K_{0,1}+1*K_{0,2}+2*K_{0,3} \cdots \cdots \cdots \end{bmatrix}$$

$$\left| \begin{array}{c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c} K_{0,0} & K_{1,0} & K_{2,0} & K_{3,0} & K_{0,1} & K_{1,1} & K_{2,1} & K_{3,1} & K_{0,2} & K_{1,2} & K_{2,2} & K_{3,2} & K_{0,3} & K_{1,3} & K_{2,3} & K_{3,3} \\ \oplus & \oplus & \oplus & \oplus & \oplus & \oplus & \oplus & \oplus & \oplus & \oplus & \oplus & \oplus & \oplus & \oplus & \oplus & \oplus \\ D_{0,0} & D_{1,0} & D_{2,0} & D_{3,0} & D_{0,1} & D_{1,1} & D_{2,1} & D_{3,1} & D_{0,2} & D_{1,2} & D_{2,2} & D_{3,2} & D_{0,3} & D_{1,3} & D_{2,3} & D_{3,3} \end{array} \right|$$

[Table 5] shows that the matrix is multiplied with each other. The results for the database entity will be shown in the table. The coded data is XOR with the other matrix entry. Furthermore, the need of improving the time complexity encountered when in this fast-moving world these algorithms have to be amalgamated with some other security techniques. If these algorithms get clubbed in any possible way without tempering or enhancing any parameter, they might advent a more secure system yet a slower one. Whereas, if any change in their functionality is done to improve their performance, they can certainly make a more fast and equally secure system. Hence in this research, work is done on enhancing the time of computation of AES and it has been ensured that its security should not get tempered.

2.2 Proposed Methodology

Enhancement of proposed AES over the existing AES would be achieved by including circular reference and mapping of array shift row by having index value which is directly mapped to the key state index. The access of the index is comparatively fast in the proposed scheme and the proposed improvement has been applied to the existing AES. The proposed system is proposed to be tested for two different cases with variable length lengths of plain text, both for encryption and decryption. Initial emphasis has been made only ontime parameters considering the quantitative nature for comparison for existing and proposed techniques.

INDEX	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
VALUE	0	5	10	15	4	9	14	3	8	13	2	7	12	1	6	11

Table. 6. Values after rotation

[Table 6] here in this methodology represents the shifting of the position of values after the rotation process takes place. Values on the positions mentioned will move according to the value given in the value column below the index column.

Table. 7. Proposed hash matrix for row recording

$$\begin{bmatrix} 63 & 7C & \ldots & AB & 76 \\ CA & 82 & \ldots & 72 & C0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ E1 & F8 & \ldots & 28 & DF \\ SC & A1 & \ldots & BB & 16 \end{bmatrix} x\,[2] = \begin{bmatrix} C6 & F8 & \ldots & 4D & EC \\ 8F & 1F & \ldots & E4 & 9B \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ D9 & EB & \ldots & 50 & A5 \\ 03 & 59 & \ldots & 6D & 2C \end{bmatrix}$$

$$\begin{bmatrix} e_{0,0} & e_{0,1} & e_{0,2} & e_{0,3} \\ e_{1,0} & e_{1,2} & e_{1,2} & e_{1,3} \\ e_{2,0} & e_{2,1} & e_{2,2} & e_{2,3} \\ e_{3,0} & e_{3,1} & e_{3,2} & e_{3,3} \end{bmatrix} = \begin{bmatrix} d_{0,0} & d_{0,1} & d_{0,2} & d_{0,3} \\ d_{1,0} & d_{1,2} & d_{1,2} & d_{1,3} \\ d_{2,0} & d_{2,1} & d_{2,2} & d_{2,3} \\ d_{3,0} & d_{3,1} & d_{3,2} & d_{3,3} \end{bmatrix} + \begin{bmatrix} K_{0,0} & K_{0,1} & K_{0,2} & K_{0,3} \\ K_{1,0} & K_{1,2} & K_{1,2} & K_{1,3} \\ K_{2,0} & K_{2,1} & K_{2,2} & K_{2,3} \\ K_{3,0} & K_{3,1} & K_{3,2} & K_{3,3} \end{bmatrix}$$

[Table 7] shows the matrix for each row for the matrix being shown on the system. All the entries are recorded in the table. This will show the values along with the matrix with a higher number of rows and columns. This is followed by the matrix with the following specification represented by matrix e.

Table.8. Multiplication

$$\begin{bmatrix} d_{0,0} & d_{0,1} & d_{0,2} & d_{0,3} \\ d_{1,0} & d_{1,2} & d_{1,2} & d_{1,3} \\ d_{2,0} & d_{2,1} & d_{2,2} & d_{2,3} \\ d_{3,0} & d_{3,1} & d_{3,2} & d_{3,3} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} x \begin{bmatrix} c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} \\ c_{1,0} & c_{1,2} & c_{1,2} & c_{1,3} \\ c_{2,0} & c_{2,1} & c_{2,2} & c_{2,3} \\ c_{3,0} & c_{3,1} & c_{3,2} & c_{3,3} \end{bmatrix}$$

$$\begin{bmatrix} c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} \\ c_{1,0} & c_{1,2} & c_{1,2} & c_{1,3} \\ c_{2,0} & c_{2,1} & c_{2,2} & c_{2,3} \\ c_{3,0} & c_{3,1} & c_{3,2} & c_{3,3} \end{bmatrix} = \begin{bmatrix} b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\ b_{1,0} & b_{1,2} & b_{1,2} & b_{1,3} \\ b_{2,0} & b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,0} & b_{3,1} & b_{3,2} & b_{3,3} \end{bmatrix}$$

It can be changed as follows

$$\begin{bmatrix} b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\ b_{1,0} & b_{1,2} & b_{1,2} & b_{1,3} \\ b_{2,0} & b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,0} & b_{3,1} & b_{3,2} & b_{3,3} \end{bmatrix} = \begin{bmatrix} S[a_{0,0}] & S[a_{0,1}] & S[a_{0,2}] & S[a_{0,3}] \\ S[a_{1,0}] & S[a1,1] & S[a_{1,2}] & S[a_{1,3}] \\ S[a_{2,0}] & S[a_{2,1}] & S[a_{2,2}] & S[a_{2,3}] \\ S[a_{3,0}] & S[a_{3,1}] & S[a_{3,2}] & S[a_{3,3}] \end{bmatrix}$$

Table 8 shows the matrix multiplication for each matrix, the results are stored into the additional matrix for subsequent usage. Matrix is followed by the specified entity after multiplication and XOR operations.

Table. 9. Operation for generating final encoding

$$\begin{bmatrix} e_{0,0} & e_{0,1} & e_{0,2} & e_{0,3} \\ e_{1,0} & e_{1,2} & e_{1,2} & e_{1,3} \\ e_{2,0} & e_{2,1} & e_{2,2} & e_{2,3} \\ e_{3,0} & e_{3,1} & e_{3,2} & e_{3,3} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} x \begin{bmatrix} S[a_{0,0}] & S[a_{0,1}] & S[a_{0,2}] & S[a_{0,3}] \\ S[a_{1,0}] & S[a1,1] & S[a_{1,2}] & S[a_{1,3}] \\ S[a_{2,0}] & S[a_{2,1}] & S[a_{2,2}] & S[a_{2,3}] \\ S[a_{3,0}] & S[a_{3,1}] & S[a_{3,2}] & S[a_{3,3}] \end{bmatrix} + \begin{bmatrix} K_{0,0} & K_{0,1} & K_{0,2} & K_{0,3} \\ K_{1,0} & K_{1,2} & K_{1,2} & K_{1,3} \\ K_{2,0} & K_{2,1} & K_{2,2} & K_{2,3} \\ K_{3,0} & K_{3,1} & K_{3,2} & K_{3,3} \end{bmatrix}$$

$$\begin{bmatrix} e_{0,j} \\ e_{1,j} \\ e_{2,j} \\ e_{3,j} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} x \begin{bmatrix} S[a_{0,j}] \\ S[a_{1,j}] \\ S[a_{2,j}] \\ S[a_{3,j}] \end{bmatrix} + \begin{bmatrix} K_{0,j} \\ K_{1,j} \\ K_{2,j} \\ K_{3,j} \end{bmatrix}$$

$$\begin{bmatrix} e_{0,j} \\ e_{1,j} \\ e_{2,j} \\ e_{3,j} \end{bmatrix} = \begin{bmatrix} 02 \bullet S[a_{0,0}] + 03 \bullet S[a_{0,1}] + 01 \bullet S[a_{0,2}] + 01 \bullet S[a_{0,3}] + K_{0,j} \\ 01 \bullet S[a_{1,0}] + 02 \bullet S[a_{1,1}] + 03 \bullet S[a_{1,2}] + 01 \bullet S[a_{1,3}] + K_{1,j} \\ 01 \bullet S[a_{2,0}] + 01 \bullet S[a_{2,1}] + 02 \bullet S[a_{2,2}] + 03 \bullet S[a_{2,3}] + K_{2,j} \\ 03 \bullet S[a_{3,0}] + 01 \bullet S[a_{3,1}] + 01 \bullet S[a_{3,2}] + 02 \bullet S[a_{3,3}] + K_{3,j} \end{bmatrix}$$

[Table 9] shows the final encoding after the operations onto the data. The data will be having addition, and XOR operations onto the data values lie into the Matrix. The whole process will move the data circularly. There will be five shifts in a circular fashion. Will generates the secured key. The security can be measured based on the parameters.

2.3 Algorithm

Algorithm 1, Mentioned in [Table 10] is for the proposed encryption technique. It input the whole data to be encrypted for security. The data will be subdivided into small sub bytes. Each sub-byte will be shifted towards the left and along with shifting there will be created hash. Later on, while decryption the data shifting identification is done from the hash table created while shifting. This will reduce the time for the search for the key to shift.

Table. 10. Encryption Algorithm

```
Algorithm 1
Encryption
Step 1 Input the plain text to the system for encryption.
Step 2 set i=1
Repeat while i<=Nr-1
Step3 Add first-round key for the first round.
Step4 Create sub bytes for the whole data.
Step5  Shift each row in the left to right manner.
      i. Generate the Hash for reducing the search time for shift identification.
      ii. Create one to one search for the hash identification.
Step6 Mix columns of the rows shifted data.
Step7 i=i+1
End loop
Step8 Create SubBytes of the data at the final stage.
Step9 Shift the rows for each sub byte.
Step10 Add round key for the final phase.
Step11 Generate the Ciphertext.
```

The algorithm mentioned for decryption in [Table 11] is having a sequence of the steps used to decrypt the ciphertext. It takes the whole bitstream of the data and then assembles the sub bytes of the data. The identified hash will be helpful for reverse shifting of the sub bytes to position the data into the original format. The time of decrypting is substantially less because the shift key is stored into the hash for each sub byte.

Table. 11. Decryption Algorithm

```
Algorithm 2
Decryption
Step1 Input the ciphertext.
Step 2 Add the round key for the first round
Step 3 set i=1
Repeat while(i<=Nr)
Step 4 Inverse shift rows
Step5 Inverse sub bytes
Step 6 AddRoundKey
Step7 Inverse mix columns
Step8 i=i+1
End loop
Step 9 Inverse shift rows
Step10 Inverse Sub Bytes
Step 11 Add round key
Step 12 Generation of Plain text
End  Algo
```

3.Results and Discussion

This section of the paper has emphasized explaining the setup adopted for the implementation of the proposed system and the results obtained in terms of enhancement of time of execution of AES.

3.1 Experimental Setup

Implementation of the improved AES for security purposes is the primary issue for the research. MATLAB as the simulation software is used for extracting the results for both AES and improved AES. Different parameters have been used for driving the experiment

Table. 12. Experimental Setup

Parameter	Value
Plain text Length	100,200 Character length
Time Measuring	Function setup
Space measurement	Function setup
Processor	I3 Processor
RAM	4 GB
Operating system	Windows 10
Simulator	Matlab 2015

3.2 Time Comparison for the base and proposed technique

Graphs in [Figure 4] and [Figure 5] show the time comparison for the existing and proposed technique. The time for the proposed system for case 1 and case 2 has shown improvement. The time for case 1 for the proposed system has shown an improvement of 25% for case 2 the proposed approach has shown an improvement of 14%. This shows using indexing and hash-based index table for the search of the used key will proportionally reduce the time. There will be a 1:1 search time for the proposed technique.

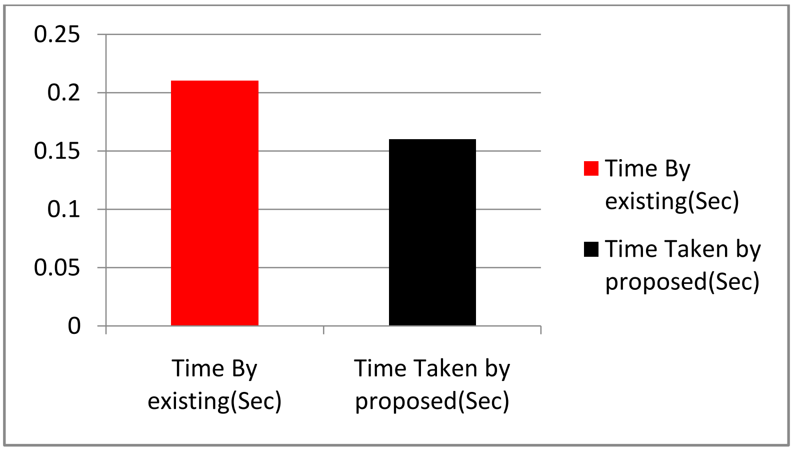


Figure. 4 Time Comparison Case 1

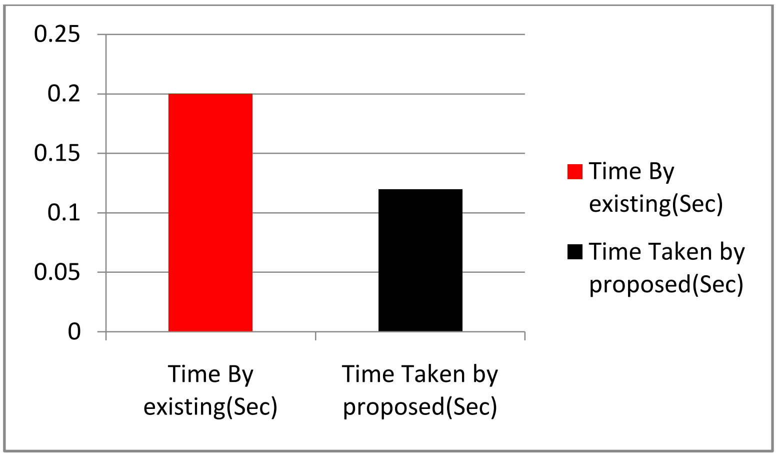


Figure.5 Time Comparison case 2

Security is achieved by having an encryption mechanism. The proposed enhanced AES is shown for the different length plain text strings. The proposed scheme can be further enhanced by reducing the memory for the table where indexing is performed. The solution to the problem will be either chaining or linear probing. This will reduce the memory space required for the proposed system.

.

4.Conclusion

AES[**Federal Information**] is the symmetric key used for data security of the data, while data is transmitted from source to destination. In the proposed enhanced algorithm, a dramatic improvement has been registered while implementing test cases. Though this technique has limitations of consuming more memory compared to the existing one, at the same time, this scheme has increased the efficiency of the algorithm by 25% and 14% respectively for the above-mentioned cases. Also, this proposed technique has emphasized mapping matrix positions which helped in attaining the results which are discussed above.

References (APA)

1. Attaullah, Sajjad Shaukat Jamal. Tariq Shah. Atta and Atta Attaullah, Sajjad Shaukat Jamal Tariq Shah, A Group Action Method for Construction of Strong.. 3D Research Center, Kwangwoon University and SpringerVerlag Berlin Heidelberg: 2017

2. Ashish Khanna, "A Java-Based Network Security on Wireless Network communication", International Journal on Recent and Innovation Trends in Computing and Communication ISSN 2321 – 8169 Volume: 1 Issue: 7,2017.

3. BawnaBhat, "DES and AES Performance Evaluation", International Conference on Computing, Communication and Automation (ICCCA2015),2015.

4. “Communication Network Security”, Book, Vol. 3, Issue 1, January 2015

5. Federal Information, Advanced Encryption Standards, Processing Standards Publication 197, November 26, 2001

6. “Introduction to Cryptography” http://www.ggu.ac.in /download/Class- Note14/ public%20key13.02.14.pdf].

7. JagbirDhillon, Krishna Prasad, Rajesh Kumar, Ashok Gill, "Secure Data in Wireless Sensor Network By Using DES", International Journal of Wireless & Mobile Networks (IJWMN) Vol. 3, No. 3, June 2015

8. Jan Sher Khan, Jawad Ahmad, Khan, Jawad Ahmad," TD-ERCS map-based confusion and diffusion of autocorrelated data" Nonlinear Dyn, Springer, 2017

9. K. Ragunath Reddy1, G.Srinivas Raju2, "A New Design of Algorithm for Enhancing Security in Bluetooth Communication with Triple DES", International Journal of Science and Research (IJSR), India Online ISSN: 2319-7064
10. Nentawe Y. Goshwe, "Data Encryption and Decryption Using RSA Algorithm in a Network Environment", IJCSNS International Journal of Computer Science and Network Security, VOL.13 No.7, July 2013

11. RizkyRiyaldhi, Rojali, Aditya Kurniawan," improvement of advanced encryption standard algorithm with shift row and s.box modification mapping in mix column", vol. 116,pp:401-407,2017.

12. Sandeep Singh, Aman Singh, "An Information Security Technique Using DES-RSA Hybrid and LSB", International Association of Scientific Innovation and Research (IASIR)

13. Somber Singh* Sunil K. MaakarDr.Sudesh Kumar, "Enhancing the Security of DES Algorithm Using Transposition Cryptography Techniques", Volume 3, Issue 6, June 2013

14. Sumartono, Isnar&Siahaan, AndysahPutera Utama. (2018). Encryption of DES Algorithm in Information Security.

15. Xiaokun Yang and Wujie Wen. Design of A PreScheduled Data Bus for Advanced Encryption Standard Encrypted System-on-Chips. IEEE: pp.506-511, 2017

.