


Research Article

An Enhanced and Adaptive Algorithm for Secure Encryption of Data using Advanced Encryption Standard (AES)

Mohand S. Taha 
University of Technology

Baghdad, Iraq

Mohanad.S.Taha@uotechnology.edu.iq

ARTICLE INFO

Article History

Received: 01/05/2025

Accepted: 11/06/2025

Published: 15/08/2025

This is an open-access article under the CC BY 4.0 license:

<http://creativecommons.org/licenses/by/4.0/>



ABSTRACT

Because of new multimedia types and advances in technology, protecting data has become extremely valuable. Because of the changes happening around us, cryptography continues to protect information that is crucial for security. Security methods and systems are continually being improved, but we should continue to review data protection methods while data travel through different services. One of the reasons data systems are secure today is because of encryption which makes plaintext into ciphertext. It examines the strengths and weaknesses of various cryptographic algorithms using what has been written by other authors. This paper reviews the use of cryptography in several literatures to make data security better nowadays. Data is protected using encryption during online and other transfers, yet the information may be obtained from an unauthorized attacker who is persistent enough. Two or more approaches to security should be combined, according to what is highlighted in the article. AES encryption and decryption can be improved by adding other algorithms such as the replacement algorithm. AES is used in this study to safeguard a message before it is presented. In 1998, Rijmen and Daemen designed the AES algorithm and named it Rijndael. Many people have started using it because its security is strong and it is not easily broken using brute force. The technique is applied on the plaintext of the AES algorithm. Thanks to this approach, breaking the encryption is extremely difficult, as you need the right key, though AES itself remains a simple system. The performances of both Advanced Encryption Standard and its modified option were tested after they were put into practice. There was an avalanche of 54.69% with the new version of Advanced Encryption Standard compared to the 50.78% observed in the standard AES. The strong encryption

Keywords: *Rijndael algorithm; Advanced Encryption Standard (AES); Polyalphabetic substitution; Avalanche effect.*

1. INTRODUCTION

All relevant communication systems require secure data transmission to preserve the privacy of the message being sent; this message could be a simple e-mail or a billion euro bank transaction. There are several encryption standards and algorithms, such as public key ciphers, symmetric ciphers, and hash functions that protect communications networks. Although public algorithms offer greater reliability, mainly because of their key size, computational demands do not allow them to be effectively used to encrypt most data. Symmetric encryption algorithms are used to encrypt large amounts of data. One of the emerging new of the standards established by the National Institute of Standards and Technology (NIST) is the Advanced Encryption Standard (AES) [1, 2]. The progression of encryption standards in reaction to the increasing complexity of cyber threats represents a crucial phase in the annals of information security. As we approached the end of the twentieth century, encryption systems, including the famous DES, had weaknesses that left them at risk from new cyber dangers. There was once belief that DES could not be penetrated, yet with new threats to encryption, a new method was needed. In view of how quickly events

were moving, in 1997, the NIST set about choosing a new way to secure data that would ultimately result in the adoption of AES. In October 2000, officials chose the Rijndael algorithm from Belgian cryptographers Vincent Rijmen and Joan Daemen after three comparison rounds and advice from leading international experts. There were many reasons for the move from DES to AES. It was mainly the short key and block sizes of DES that left it subject to a range of current attack techniques. Unlike the DES, the AES made improvements both in efficiency and in saving memory, so it works better for today's computer systems. Researchers have shown that the AES algorithm can withstand most types of attacks. AES is highly protected and has not been handled; it keeps its data safe from intruders so far. AES's stability shows why it is now a vital part of modern encryption and marks a big step ahead of DES. This step toward AES introduces a major change in cryptography which is likely to make networks safer and their information better protected. Encryption turns regular information into an unreadable organization called (encrypted-text). Following decoding, the information restores to its unique frame which is named normalized. The algorithm takes encryption keys measuring 128, 192 or 256 bits to process data in blocks with the same size [3,4]. In the last few years, many articles have appeared looking to address AES security issues and improve the Standard AES. Authors Abeer T. Maolood and Yasser A. Yasser claimed in 2017 that making a few updates to the AES algorithm can reinforce the security of Standard AES [5]. An important part of the first alteration involves key-dependent techniques. The AES-NI design replaces the single S-box in the Standard AES with 10 S-boxes which improves the mixing of data in the Byte Substitution layer. Improvements to how rows are moved in the ShiftRows layer are made by using key-derived variables rather than the fixed positions in Standard AES. The third update is to use two keys instead of one like in Standard AES. Both are used for both encryption and decryption which improves how AES is structured and how its keys are made. In the year 2018, Sumathy Kingslin and others [6] provided a detailed introduction to five encryption techniques involving substitution and transposition and this research assessed these schemes based on criteria such as encryption and decryption time, memory usage and their avalanche effect. As reported in [7], the proposed strategy raised the complexity of key production which helped maintain their security against hackers. This proposed method creates keys using the 3DKGM procedure, not through the usual two-dimensional S-box system. The system's resilience and security depend on the use of chaos cryptography and Logistic Map. The continuing effectiveness regarding critical Internet of Things data is analyzed by looking at a scenario based on a smart house protected by the Internet of Things. For compatibility with the smart-home system, a sensor must be capable of enough computing. Eltayeb Ahmed proposed in 2019 a straightforward approach for modifying the AES S-Box design in non-linear ways. This way, we not only find the inverse easily, but the method also does the matrix multiplication needed for this transformation to work. As a result, there is no need for the characteristic matrix in this approach, so the updated method is easier to use in building the S-Box [8]. Runju S Karthal and Varghese Paul announced a one-of-a-kind cipher in 2019 that uses several random tables to do its encryption. We suggest that each letter in the plaintext be encrypted with a random table of 26×26 size. For this situation, instead of using one Vigenere table, we introduce an unlimited number of alphabetical tables, the amount decided by the length of the text. Every stage of the development process will produce a new and different table. Experts say crackers cannot hack this code because the keyword appears twice everywhere in the ciphertext [9].

2. AES ALGORITHM

2.1 Structure of AES

Another paragraph describes the general structure of AES. Each 128-bit input to calculate encryption and decoding is considered a single original bit. This entrance is described by a square network of byte. This part is imitated in the field of state, amended into encryption or interpretation. The state is copied in productivity network by performing after the final organization. In the middle of the initialization, the 128-bit lock is considered a random byte network in the same way. Refer to the words of the locking plan at that time to amplify the lock; for 128-bit lock, each word is four bytes and global lock plans include words. In the network, the byte is organized in the way of arranging columns. The main column of the frame contains four main bytes of the input in the text in 128 bits clear to encrypt encoding, the time column contains four other bytes and this design takes place. In addition, the four keys are expanded and the frame is separated in the column of the network W. Some characteristics of AES complete technique [10]:

1. An extension handle is applied to the input key, resulting in a group of forty-four 32-bit words, referred to as $w[i]$. As shown in Figure 1a, four specific words (equivalent to 128 bits) are used as the circular key for each circular of the encryption handle.

2. The encryption handle consists of four distinct phases, including three stages of substitution and one arrangement of stages:

- Replace Bytes: In this step, the information square is replaced byte by byte using an S-box.

- Reposition Columns: This arrangement gives the information a clear stage.

- Blend Columns: This arrangement substitutes using number-crunching procedures over $GF(2^8)$.

- Include Circular Key: In this step, the current data piece and a portion of the extended key are subjected to a simple bitwise XOR operation.

3. The design of the Progressed Encryption Standard (AES) is generally direct. Both the encryption and decoding forms commence with the Include Circular Key arrange, taken after by nine rounds that join all four stages, concluding with a tenth circular that comprises of three stages. Figure (2) outlines the structure of a total encryption circular [10].

4. The Add Round Key stage is the only stage that utilizes the key. Consequently, the cipher initiates and terminates with an Add Round Key stage. Additional arrangements would be reversible without the key, thereby contributing no extra security.

5. The Add Round Key organize capacities additionally to a Vernam cipher and, in separation, would not give considerable security. The remaining three stages collectively introduce confusion, diffusion, and nonlinearity; however, they would not offer security on their own due to their lack of key utilization. The cipher operates through alternating processes of XOR encryption (Add Round Key) of a data block, followed by the scrambling of the block (the other three stages), and then another round of XOR encryption, and so forth. This methodology is both efficient and highly secure.

6. Each stage is readily reversible. In the decryption algorithm, inverse functions are employed for the Substitute Bytes, Shift Rows, and Mix Columns stages. For the Add In the Round Key stage, the inverse operation is performed by XORing the data block with the same round key, taking advantage of the property that $A \oplus A \oplus B = B$.

7. The decryption algorithm uses the expanded key in reverse order, as is typical for most block ciphers. It is, however, crucial to point out that the encryption and decryption algorithms are not the same; this is a feature of AES's particular structure.

8. When all four stages are determined to be reversed, it is only a matter of confirming that unscrambling provides an effective recovery of the first plain text. According to Figure 1, encryption and decoding move in inverse vertical directions. At each even intersection (such as the dashed line within the outline), the state is uniform for both encryption and decoding.

9. In both the encryption and unscrambling forms, the ultimate circular comprises as it were three stages. This plan choice stems specifically from AES's one of a kind structure and is vital for ensuring the cipher's reversibility.

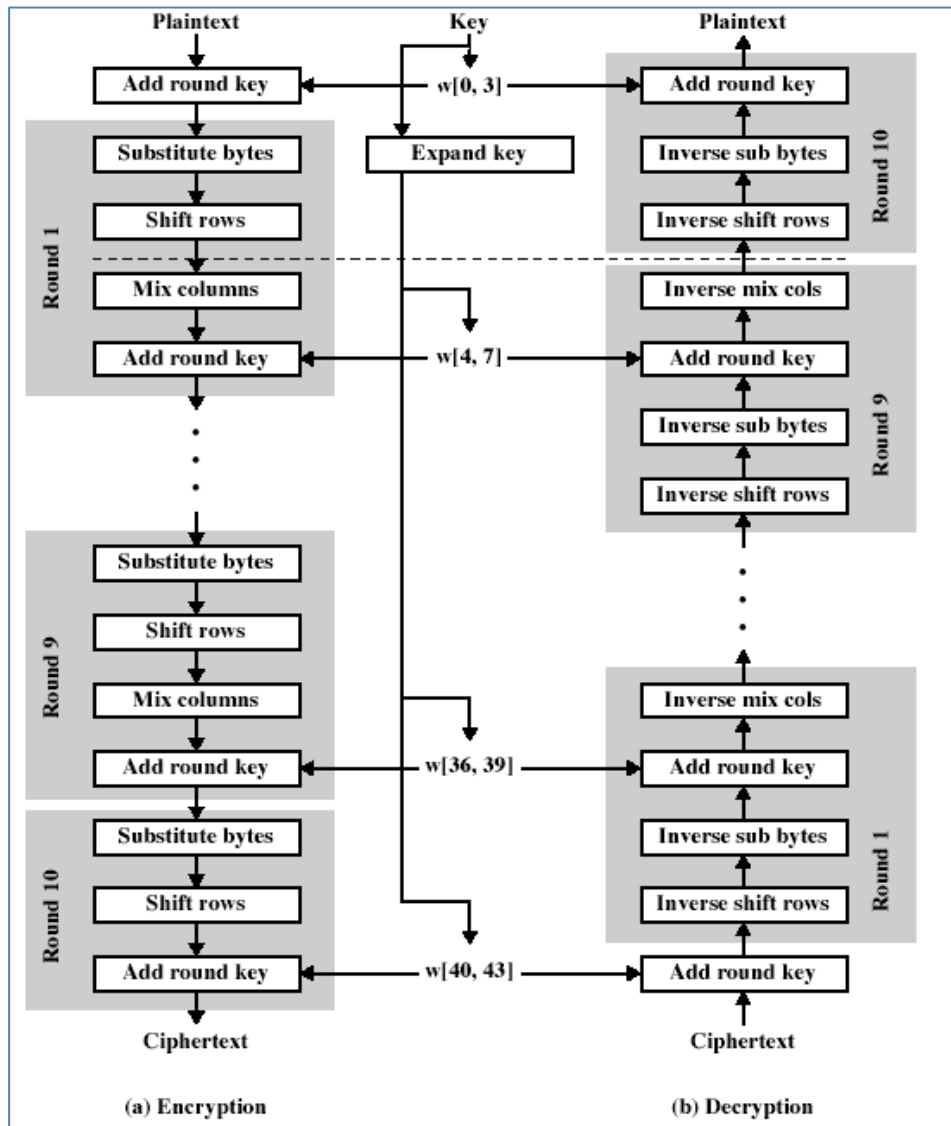


Fig. 1. The process of AES-128 (a) Encryption, (b) Decryption [10].

2.2 AES Encryption

2.2.1 Substitute Bytes Transformation

The bytes substitution transformation, as described in reference [11], constitutes a non-linear substitution process that functions independently on each byte of the state, utilizing a substitution table (S-box) illustrated in Figure 2. The S-box employed in the Sub Bytes transformation is represented in hexadecimal format in the same figure. For instance, if $S_{1,1}$ is defined as $\{53\}$, the corresponding substitution value is ascertained by locating the intersection of the row indexed by '5' and the column indexed by '3' in Figure 3. Consequently, this results in $S'_{1,1}$ being assigned a value of $\{edh\}$ [12].

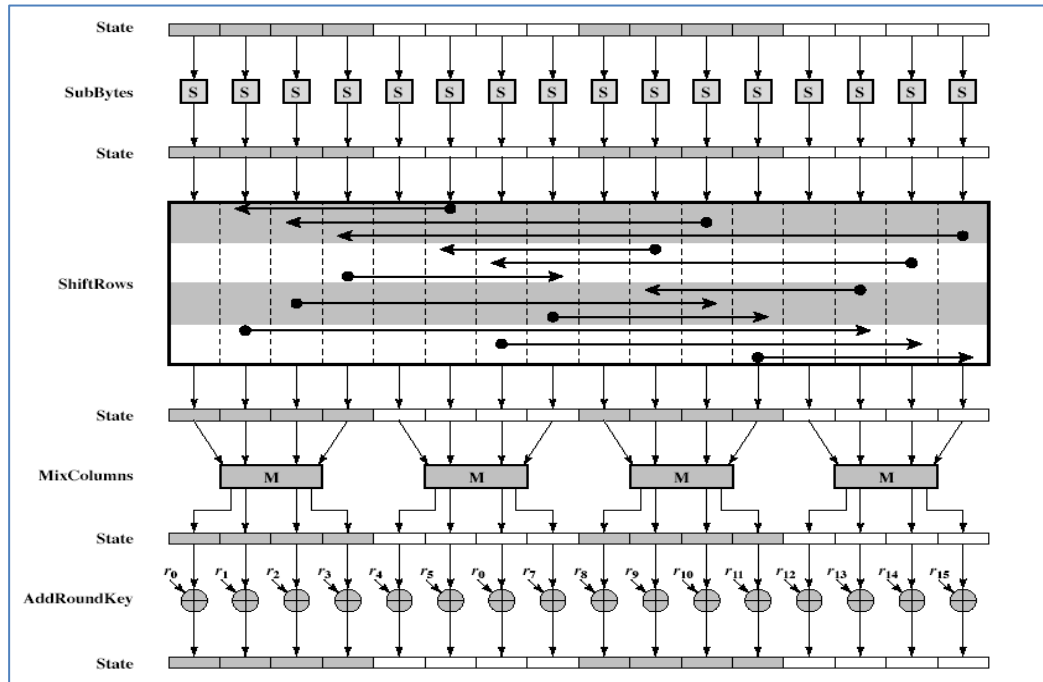


Fig. 2. AES encryption algorithm [10].

		0	1	2	3	4	5	6	y	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	
x	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	2c	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	

Fig. 3. S-Box [13].

2.2.2 Transformation of Shift Rows

The bytes within the last three lines of the State are consistently moved by different byte offsets within the Move Columns change, which is spoken to by ShiftRows(). When this operation is performed, the bytes involved in the most reduced areas are pushed to the top of the push (i.e., higher values of c within the same push), while the bytes involved in lower positions are moved to lesser centrality within the push (i.e., lower values of c within a characterized push). In Figure 4 [12,14], ShiftRows () appears to have been changed.

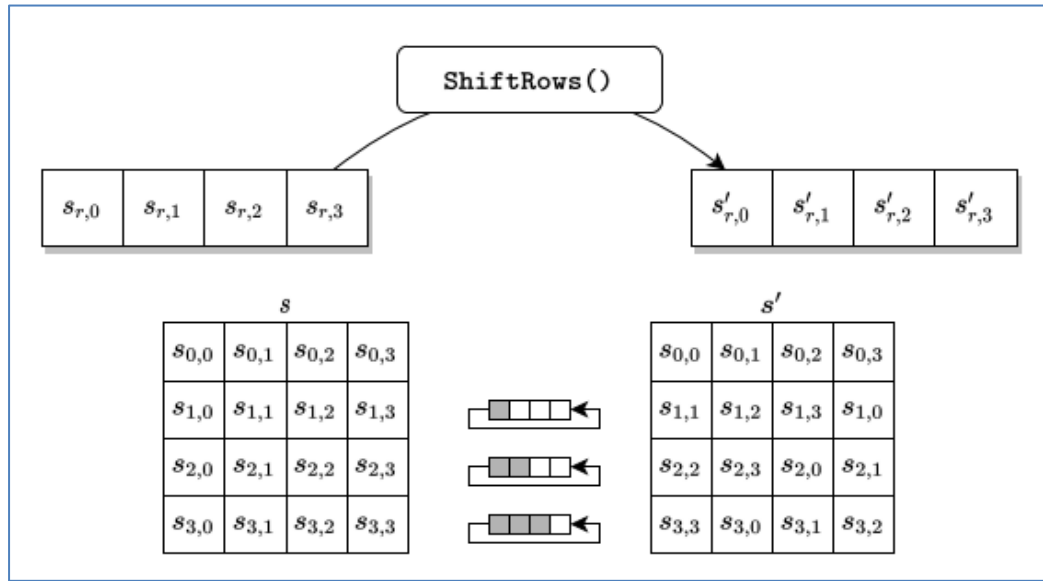


Fig. 4. Shift Row [12].

2.2.3 Mixing of Columns Transformation

The Galois Field increase guidelines serve as the foundation for this modification. As shown in Figure 5, each byte inside a column is replaced with a contemporary value determined by a calculation of the four bytes displayed in that specific column (as shown in Equation 1) [15 - 18].

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \quad (1)$$

This multiplication results in the following (equation 2) [12] replacing the four bytes in a column:

$$\begin{aligned} s'_{0,c} &= (02 \bullet s_{0,c}) \oplus (03 \bullet s_{1,c}) \oplus s_{2,c} \oplus s_{3,c} \\ s'_{1,c} &= s_{0,c} \oplus (02 \bullet s_{1,c}) \oplus (03 \bullet s_{2,c}) \oplus s_{3,c} \\ s'_{2,c} &= s_{0,c} \oplus s_{1,c} \oplus (02 \bullet s_{2,c}) \oplus (03 \bullet s_{3,c}) \\ s'_{3,c} &= (03 \bullet s_{0,c}) \oplus s_{1,c} \oplus s_{2,c} \oplus (02 \bullet s_{3,c}) \end{aligned} \quad (2)$$

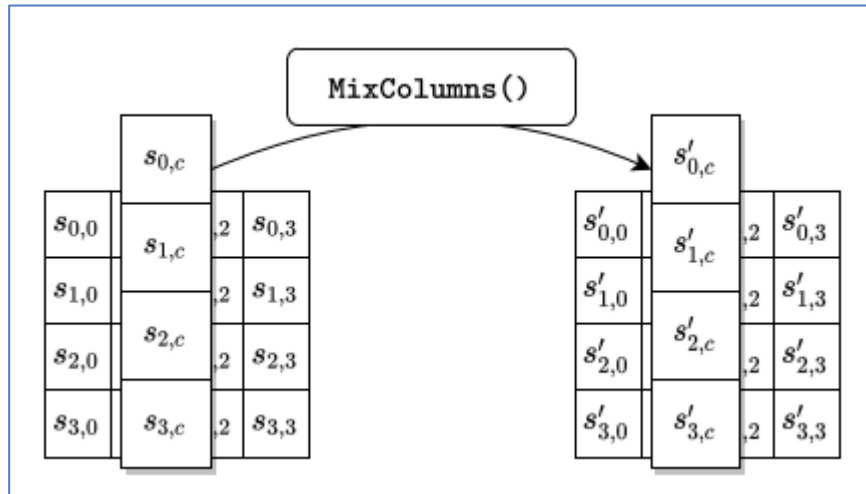


Fig. 5. Mixing of Columns Transformation [18].

2.2.4 Add Round Key Transformation

As seen in the writing [12], the AddRoundKey change is the bitwise XOR operation applied to the 128 bits of the State and the 128 bits of the circular key. This operation can be thought of as a column-wise interaction between a single word of the round key and the four bytes of a State column, as shown in Figure 6; it can also be thought of as a byte-level operation. Take the first matrix, which represents the State, and the second matrix, which represents the round key, for example. Notably, the inverse of the AddRoundKey transformation is equivalent to the forward transformation, given that the XOR operation is self-inverse. The simplicity of the AddRoundKey transformation ensures that it influences every bit of the State. Furthermore, the complexity associated with the round key expansion, in conjunction with the intricacies of the other stages of the Advanced Encryption Standard (AES), contributes to the overall security of the encryption process [19].

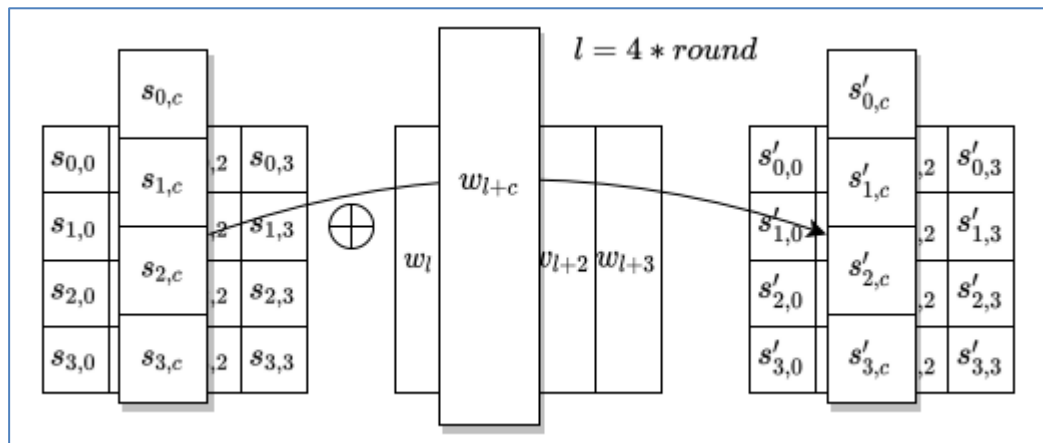


Fig. 6. Add Round Key Transformation [12].

2.2.5 AES Key Expansion

The AES key extension computation creates a straight cluster of 44 words (156 bytes) using a 4-word (16-byte) key as input. This result is sufficient to provide a 4-word circular key for each of the ten rounds of the encryption procedure as well as the main Add Round Key phase. The pseudocode that follows shows the most development preparation: To start with four words, the key is first replicated into the expanded key.

Four-word increases are then used to fill in the leftover space in the extended key. The word that arrives right some time recently, $w[i-1]$, and the word that comes four positions some time lately, $w[i-4]$, are dependent upon each newly implanted word, spoken to by $w[i]$. A straightforward XOR operation is used in three of the four situations. For words that are products of four within the w cluster, however, a more intricate process is used. Figure 7 shows the key's expansion primary eight words, and image g represents this intricate task.

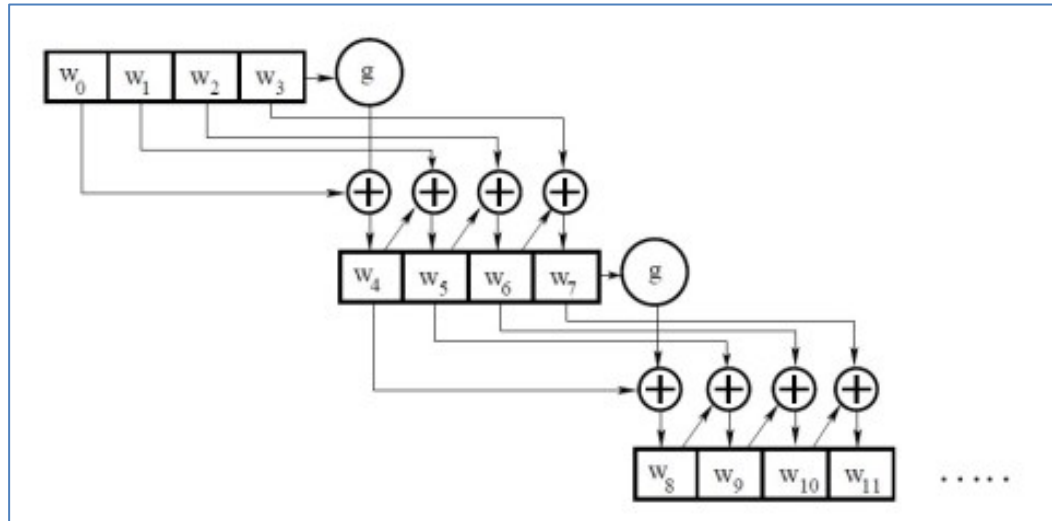


Fig. 7. Key expansion [20].

The following sub-functions are included in the function g :

1. RotWord: This function performs a one-byte circular left shift on an input word $[b_0, b_1, b_2, b_3]$ to transform it into $[b_1, b_2, b_3, b_0]$.
2. SubWord: This function replaces each byte of its input word with a byte using the S-box.
3. The outcomes of steps 1 and 2 are then subjected to an XOR operation using a round constant, $Rcon[j]$. A word is referred to as a round constant if its three rightmost bytes are always zero. Consequently, when a word and $Rcon$ are put through an XOR operation, just the word's leftmost byte will be impacted. Figure (8) illustrates the number of rounds required for three distinct key lengths [21].

Algorith m	Cipher Key size	No. of Rounds (Nr)	Round Key size
AES-128	128 bits	10	128 bits
AES-192	192 bits	12	128 bits
AES-256	256 bits	14	128 bits

Fig. 8. Key-Block- Round Combinations [22].

2.3 AES Decryption

To recover data that has been previously encrypted, it is necessary to execute the inverse of the Advanced Encryption Standard (AES) cipher. This decryption process mirrors the encryption procedure; however, the operations conducted within each subroutine differ. The architecture of the inverse AES cipher is illustrated in Figure 1a. Each operation utilized during encryption must be inverted to facilitate the decryption of a message. The following are descriptions of each of these inverse operations [1,12, 23]:

1. InvShiftRows: The InvShiftRows operation functions similarly to the ShiftRows operation, with the distinction that the rotation is executed to the right rather than to the left.
2. InvSubBytes: The InvSubBytes operation parallels the SubBytes operation, but employs the inverse of the S-box utilized during encryption, as depicted in Figure 9.
3. InvMixColumns: The InvMixColumns operation serves as the inverse of the MixColumns transformation that was previously described for encryption.

								y									
		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
x	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

Fig. 9. Inverse S-Box [24].

3. SUBSTITUTION ALGORITHM

In the field of cryptography, a substitution algorithm refers to a technique employed to obscure a message by systematically replacing each character or symbol in the original text with an alternative character or symbol. This method, commonly referred to as a substitution cipher, constitutes a form of encryption in which each component of the plaintext most often a letter is substituted with a different component, which may be another letter, symbol, or a combination of symbols. The fundamental concept is to replace one set of symbols with another in such a way that the original message becomes difficult to interpret without knowledge of the specific substitution scheme. The aim is to ensure that anyone who isn't aware of the substitution rule can't read the message [25].

4. PROPOSED ALGORITHM

The basic idea behind the suggested approach is to encrypt the plaintext message using a substitution cipher algorithm before using it as an input for the AES algorithm. A substitution cipher is one in which the plaintext's letters are swapped out for different letters, digits, or symbols. To encrypt a message, the suggested approach combines two distinct keywords with a table, as illustrated in figure 10. To encrypt the following plaintext message, for instance:

"TO BE OR NOT TO BE"

The encryption process starts by writing the keywords "ALGORITHM and PLUS" above the plaintext message as many times as needed. For each letter in the plaintext, the ciphertext is derived using the tableau by identifying the intersection of the column provided by the plaintext letter and the row provided by the matching keyword letter.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Fig. 10. Table of polyalphabetic substitution [26].

Keywords:

AL GO RI THM PL US

Plaintext:

TO BE OR NOT TO BE

1st Ciphertext:

TZ HS FZ GVF IZ JW

From the table, as shown in figure 11, the letter (**A**) in keyword and (**T**) in plaintext, result in letter (**T**) in ciphertext. An encrypted message can be decrypted just as easily. Above the message, the keyword is written several times:

Keywords:

AL GO RI THM PL US

1st Ciphertext:

TZ HS FZ GVF IZ JW

Plaintext:

TO BE OR NOT TO BE

Decryption process uses the keyword letter to pick a column of the table and then traces down the column to the row containing the ciphertext letter. The index of that row is the plaintext letter. There are 4 '**O**'s in the plaintext message and that they have been encrypted by '**Z**,' '**F**,' '**V**,' and '**Z**' respectively. This successfully masks the frequency characteristics of the letter '**O**'. One way of looking at this is to notice that each letter of the keyword **ALGORITHM** and **PLUS** picks out 1 of the 17 possible substitution

alphabets given in the table. Also, the two different keywords increase the security of encryption process. Every message encrypted with a table cipher is a collection of simple substitution ciphers, since the number of these ciphers is equal to the number of letters in the keywords. The flow chart for the proposed algorithm is shown in Figure 11.

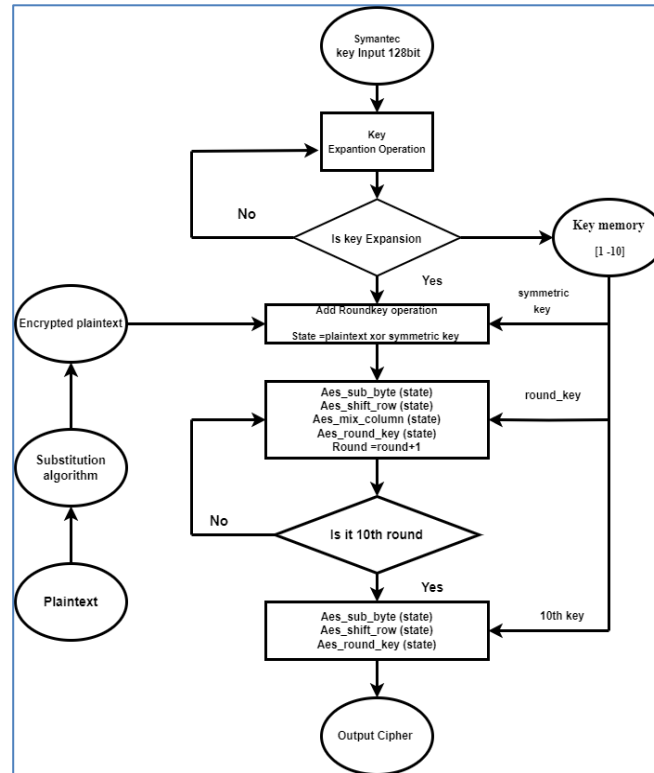


Fig. 11. The proposed AES algorithm's flow chart.

5. RESULTS AND DISCUSSION

A message is arranged in a 4x4 matrix called the State matrix (M) is shown in Figure 14. Using this structure is fundamental in the Advanced Encryption Standard (AES) since it makes encryption much easier to organize and carry out in parallel. In AES systems, most users pick a key length and data block size of 128 bits so that the algorithm is secure but doesn't require too much computer power. Improvements have been made to the recommended strategy to raise both the difficulty and security of the encryption process. Such improvements are designed to strengthen AES against differential and linear cryptanalysis without changing its usual form. By inserting an adaptive S-box or changing the transformation step, the encryption method could show new and variable features. In Figure 13, the encrypted result is displayed, followed in Figure 12 by the same data back in plaintext form. Correct implementation of the suggested changes is shown by the ability to restore the plaintext after decoding. These simulations were possible in MATLAB which is a flexible system for building and inspecting cryptographic solutions. Thanks to MATLAB, creating and analyzing the encryption scheme became much simpler, helping prove which design changes affected the algorithm's functioning. Ultimately, the algorithm changes increase AES's safety, but do not reduce its basic management and performance. The results of encrypting and decrypting are shown in Figure 12 and Figure 13.

T	O	B	E
O	R	N	O
T	T	O	B
E	Space	Space	Space

Fig. 12. Plaintext.

Plaintext	54 4f 42 45 4f 52 4e 4f 54 54 4f 42 45 20 20 20
Cipher text	54 46 46 57 5a 5a 49 20 48 47 5a 20 53 56 4a 20
Key	00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
Round 1	d6 d2 da d6 aa af a6 ab 74 72 78 76 fd fa fl fe
Round 2	b6 64 be 68 92 3d 9b 30 cf bd c5 b3 0b fl 00 fe
Round 3	b6 d2 6c 04 ff c2 59 69 74 c9 0c bf 4e bf bf 41
Round 4	47 95 f9 fd f7 35 6c 05 f7 3e 32 8d bc 03 bc fd
Round 5	3c a9 50 ad aa 9f f3 f6 a3 9d af 22 e8 eb 57 aa
Round 6	5e f7 a7 0a 39 a6 55 a3 0f 92 3d 1f 7d 96 c1 6b
Round 7	14 e3 44 4e f9 5f 0a a9 70 e2 df c0 1a 8c 4d 26
Round 8	47 a4 e0 ae 43 1c 16 bf 87 65 ba 7a 35 b9 f4 d2
Round 9	54 f0 10 be 99 85 93 2c 32 57 ed 97 d1 68 9c 4e
Round 10	13 e3 f3 4d 11 94 07 2b 1d 4a a7 30 7f 17 8b c5
Cipher text	2b 97 68 b2 9c e1 6f 57 b7 ae 29 d9 1d 71 03 2b

Fig.13. 128-bit data, 128-bit key encrypting.

Cipher text	2b 97 68 b2 9c e1 6f 57 b7 ae 29 d9 1d 71 03 2b
Key	00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
Round 1	54 f0 10 be 99 85 93 2c 32 57 ed 97 d1 68 9c 4e
Round 2	47 a4 e0 ae 43 1c 16 bf 87 65 ba 7a 35 b9 f4 d2
Round 3	14 e3 44 4e f9 5f 0a a9 70 e2 df c0 1a 8c 4d 26
Round 4	5e f7 a7 0a 39 a6 55 a3 0f 92 3d 1f 7d 96 c1 6b
Round 5	3c a9 50 ad aa 9f f3 f6 a3 9d af 22 e8 eb 57 aa
Round 6	47 95 f9 fd f7 35 6c 05 f7 3e 32 8d bc 03 bc fd
Round 7	b6 d2 6c 04 ff c2 59 69 74 c9 0c bf 4e bf bf 41
Round 8	b6 64 be 68 92 3d 9b 30 cf bd c5 b3 0b fl 00 fe
Round 9	d6 d2 da d6 aa af a6 ab 74 72 78 76 fd fa fl fe
Round 10	00 04 08 0c 01 05 09 0d 02 06 0a 0e 03 07 0b 0f
Plaintext	54 4f 42 45 4f 52 4e 4f 54 54 4f 42 45 20 20 20

Fig. 14. 128-bit data, 128-bit key decrypting.

A phenomenon found in cryptographic encryption, often termed the "avalanche effect" [27], sets encryption apart from other information systems. Cryptographic algorithms depend greatly on the avalanche effect as a major feature. Every block cipher and hash function in cryptography wants to exhibit the avalanche effect. A glitch involves a single tiny input that causes the output to be greatly affected, preferably by changing one-half of the bits produced. Since the trait makes sure there's a big difference between input and output, it greatly decreases the ability to attack the algorithm by common methods. Measuring the strength of a high-quality block cipher relies first on whether it exhibits the avalanche effect. If you want a good cipher, don't use something that ignores small changes in the message or the encryption key. If a cipher is not sensitive, its ciphertext might appear predictable which makes it much easier for attackers to guess information about the original message or the key by looking at the statistics. Because of these flaws, both privacy and integrity of encrypted messages may be at risk [27, 28]. To make this assessment, normal AES and the revised version proposed in this paper were each developed and assessed. The investigation measured how a single change in the bit of plaintext input affected the corresponding bits in the ciphertext. The testing was completed using AES-128 and the findings tell us a lot and are also impressive. Traditional AES (29–31) follows its expected avalanche effect of 50.78%, as results will be safe if the avalanche effect is above 42 percent. The newer AES algorithm was found to diffuse messages more evenly, since its avalanche effect (54.69%) was larger than the one in the previous algorithm. Such unpredictability results from the upgraded version using enhanced ways to make it harder for an opponent to recognize how the output is related to the input. To confirm the results, twenty test cases compared what happened when one bit in the plaintext was changed, but the key did not change. A review of the outputs related to the analysis

had all the ciphertext included and the amount of different bits was counted. The width of the avalanche in one case was shown by a difference of 70 bits out of 128. The outcomes appear in **Figure 15** which compares the effect of the single-bit change in the key and in the plaintext. The numerical analysis is confirmed by the graphical evidence which shows that the ciphertext responds sharply and not as a simple line. To sum up, the stronger avalanche performance of the modified AES confirms that it can deliver high security even when little efficiency is sacrificed. This property enhances the reliance on our encryption algorithm and demonstrates that the new adjustments lead to better protection.

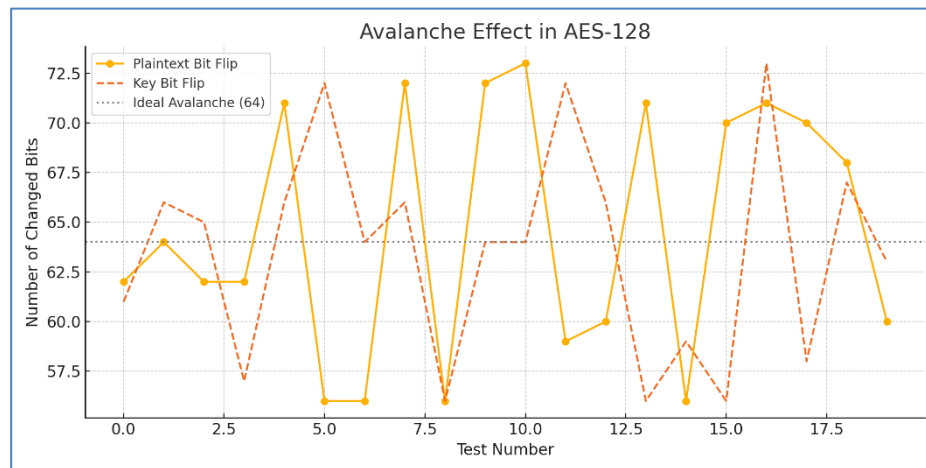


Fig. 15. The comparing change in message (plaintext) and key.

In the Figure 16, the bars represent the percentage of the avalanche effect resulting from 20 tests, where only one bit in the plaintext was changed. The red dashed line indicates the ideal avalanche effect value (50%), while the solid green line shows the actual average of the results, which exceeds 50%, indicating the strength and robustness of the encryption in the modified version of AES.

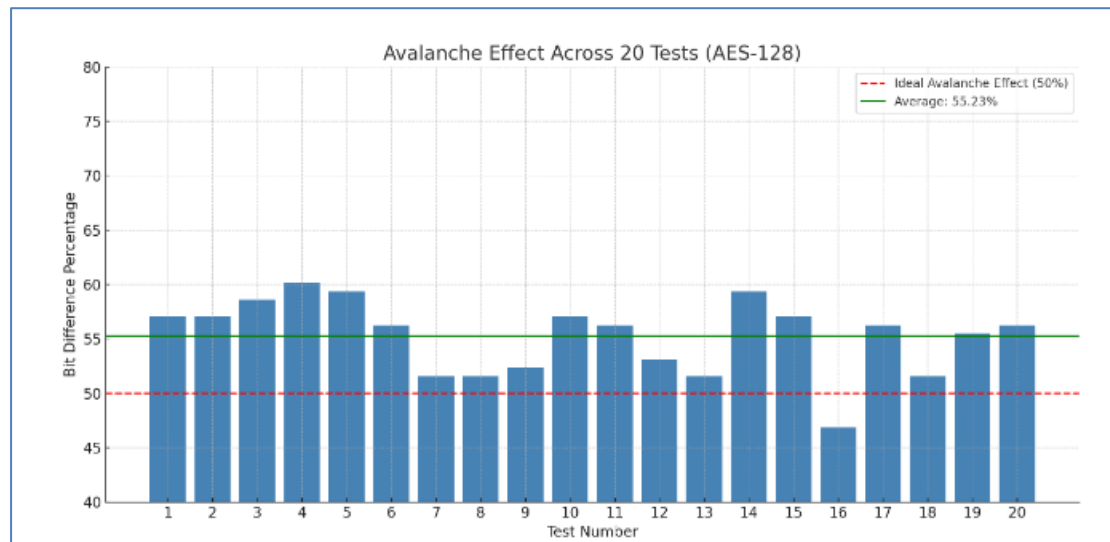


Fig. 16. Percentage of Bit Differences in Ciphertext after One-Bit Change in Plaintext (AES-128).

First, we studied the results from our AES modification and then compared them to what has been found in prior studies. Experts compare the avalanche percentage by dividing the average number of output bits that change by one change in the same place in the input. For this analysis, the modified AES had a 54.69% avalanche effect which was better than that seen in regular AES (50.78%) or in previous implementations. Greater avalanche behavior shows better diffusion and greatly boosts the defense and dependability of the

system. The table beneath this summary includes and compares the avalanche data released by various sources.

Table II The avalanche effect result.

Algorithm	Avalanche Effect (%)	References
Proposed modification with enhanced S-box	54.69%	This Work (Modified AES)
Based on original AES implementation	50.78%	Standard AES
Conventional AES with minor optimizations	49.85%	[32]
AES with custom key scheduling	51.32%	[33]
AES with lightweight block cipher integration	52.10%	[34]

6. Conclusions

The work here suggests a revised version of the Advanced Encryption Standard (AES) for improved and adjustable encryption. Thanks to substitution, the technique works together nicely with AES without raising the key or block sizes. As a result, the proposed fixes are both convenient for users and do not weaken existing protection. While AES provides a high and commonly recognized level of security, the changes introduced in this work try to make the system even stronger. The great majority of improvements result from refining diffusion and confusion methods, two important ideas introduced by Claude Shannon as essential for secure encryption. Thanks to more complicated internal changes, the upgraded AES spreads data out better and makes the pattern between what was encrypted and what is decrypted harder to decipher. In addition, the experiments verify that the new modifications enhance protection against brute-force and statistical assaults. With the modified AES algorithm, the avalanche effect is stronger than it is with normal AES. Because of this, crypto systems can avoid recognizable patterns in their responses which makes it less likely for anyone unauthorized to intercept them. According to the research, making changes to the algorithm does not reduce how well it works. Measured using MATLAB in actual simulations, the method performed with relatively strong time efficiency and better data security. The fact that the method offers powerful security and performance means it is useful in the real world, mainly in financial transactions, secure messaging and military use. In essence, the changes supported by this research update AES encryption in realistic and effective ways. The different approach introduces an important improvement to current encryption by raising diffusion and confusion levels and decreasing effort. In the future, researchers may make the system more efficient, build practical models for hardware and evaluate it along with other emerging encryption standards to widen its uses and reliability.

Declarations

Funding: Unfunded project

Ethical approval: Not needed

AI statement: AI tools were employed to aid in the literature review, data analysis, and generation of initial drafts.

Conflict of interest: The study has no conflict with any other work or any kind of interest.

Data availability statement: We can provide data upon request.

Author contributions: The authors contributed equally in the study.

References

- [1] Altigani, A., Hasan, S., Barry, B., Naserelden, S., Elsadig, M. A., & Elshoush, H. T., "Polymorphic AES Encryption Implementation," Delft University of Technology, Netherlands, 2005.

-
- [2] Aldhaibani, Jaafar A., and Nael A. Al-Shareefi. "Free space optics backhaul link for small cells of 5G cellular networks." *Journal of Engineering Science and Technology* 15.3 (2020): 1685-1697.
 - [3] Verbaunghede, I., Schaumont, P. and Kuo, H., "Design and Performance Testing of a 2.29-GB/s Rijndael Processor", *IEEE Journal of Solid-State Circuits*, Vol. 38, No. 3, March 2003.
 - [4] Rebwar Khalid Muhammed et al, "Comparative Analysis of AES, Blowfish, Twofish, Salsa20, and ChaCha20 for Image Encryption," *Kurdistan Journal of Applied Research (KJAR)*, Vol.9, No.1, June 2024.
 - [5] Abeer T. Maolood and Yasser A. Yasser, " Modifying Advanced Encryption Standard (AES) Algorithm," *Journal of Al Rafidain University College*, Issue No. , 2017.
 - [6] Sumathy Kingslin et al, " Evaluative Study on Substitution and Transposition Ciphers," *International Journal of Creative Research Thoughts (IJCRT)*, Vol. 6, No.1, January 2018.
 - [7] Rahman, Z. et al, "A. Enhancing AES Using Chaos and Logistic Map-Based Key Generation Technique for Securing IoT-Based Smart Home," *Electronics, MPDI*, Vol. 11, No.7, 2022.
 - [8] Ahmed, W.E et al, "A Modern Method for Constructing the S-Box of Advanced Encryption Standard. *Applied Mathematics*," *Applied Mathematics Journal*, Vol. 10, pp. 234-244, 2019.
 - [9] Ranju S Kartha1 et al, " An efficient algorithm for polyalphabetic substitution using random tables," *International Journal of Advanced Technology and Engineering Exploration*, Vol.5, No. 43, 2018.
 - [10] Stallings, W. "Cryptography and Network Security: Principles and Practices", Third Edition, Pearson Education, Inc., 2003.
 - [11] J. Daemen et al, "AES Proposal: Rijndael", AES Algorithm Submission, September 1999, <http://www.csrc.nist.gov/CryptoToolkit/aes/rijndael/>.
 - [12] Elaine Barker, "Announcing the Advanced Encryption Standard (AES)", *Federal Information Processing Standards Publication 197*, May 2023.
 - [13] Amandeep Singh1, et al," Analysis of Development of Dynamic S-Box Generation," *Journal of Computer Science and Information Technology*, Vol. 5, No.5, pp.154-163, 2017.
 - [14] Rizky Riyaldhi et al, " Improvement Of Advanced Encryption Standard Algorithm With Shift Row And S.Box Modification Mapping In Mix Column," *2nd International Conference on Computer Science and Computational Intelligence (ICCS CI 2017)*, *Procedia Computer Science*, Vol. 116, pp.401–407, October 2017.
 - [15] Mustafa Al-handhal et al, " Enhancing the Process of AES: A Lightweight Cryptography Algorithm AES for Ad-hoc Environments," *Journal of Advanced Research in Natural and Applied Sciences*, Vol. 8, No. 4, pp. 569-582, 2022.
 - [16] S. Sridevi sathya priya, et al, "Implementation of Efficient Mix Column Transformation for AES encryption," *4th International Conference on Devices, Circuits and Systems (ICDCS)*, IEEE publisher, pp. 95-100, 2018.
 - [17] Neethan Elizabeth Abraham et al, " FPGA Implementation of Mix and Inverse Mix Column for AES Algorithm," *International Journal for Scientific Research & Development*, Vol. 1, No.9, 2013.
 - [18] Neenu Shaji et al, "Area Optimized Architecture for AES Mix Column Operation," *International Journal of Engineering Research & Technology (IJERT)*, Vol.4, No. 9, September 2015.
 - [19] Sabbir Mahmud, "A Study on Parallel Implementation of Advanced Encryption Standard (AES)", M.Sc Thesis, Independent University, Bangladesh, May 2004.
 - [20] Amandeep Singh et al., "Analysis of Development of Dynamic S-Box Generation," *Journal of Computer Science and Information Technology*, Vol. 5, No. 5, pp. 154-163, 2017.
 - [21] Zheng, Xinxing, Han Yan et al, "Low-Delay AES Key Expansion Units Based on DDBT Structure," *Electronics MPDI*, Vol.14, no. 2, 2025.
 - [22] Joshi, A. et al., "Implementation of S-Box for Advanced Encryption Standard," *IEEE International Conference on Engineering and Technology (ICETECH)*, pp: 1-5, March 2015.
-

- [23] R. R. Rachh et al, "Efficient implementations of S-box and inverse S-box for AES algorithm," IEEE Conference (TENCON), pp. 1-6, 2009.
- [24] Jaime David Rios Arrañaga, et al, "New S-box calculation approach for Rijndael-AES based on an artificial neural network," ReCIBE, vol. 6, no. 2, pp. 49–69, Nov. 2017.
- [25] Lalit Chaurasiya et al, "A Cryptographic Technique Involving Substitution Algorithm and Logical Operator", International Journal of Innovative Science and Research Technology, Vol.10, No.3, March 2025.
- [26] Ranju S Kartha et al, "Polyalphabetic Substitution Cipher Using Multiple Random Table", International Journal of Computer Sciences and Engineering, Vol.6, Issue.5, pp.51-58, 2018.
- [27] Farshid Hossein Nejad et al., "Analysis of Avalanche Effect on Advance Encryption Standard by Using Dynamic S-Box Depends on Rounds Keys", International Conference on Computational Science and Technology (ICCST'14), 2014.
- [28] Aldhaibaini, Jaafar A., et al. "Performance Analysis of Two-Way Multi-User with Balance Transmitted Power of Relay in LTE-A Cellular Networks." Journal of Theoretical & Applied Information Technology 51.2 (2013).
- [29] Chandra Prakash Oewangan et al, " Study of Avalanche Effect in AES Using Binary Codes," IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), pp.183-187, 2012.
- [30] De Los Reyes et al, "Modified AES Cipher Round and Key Schedule," Indonesian Journal of Electrical Engineering and Informatics (IJEI), Vol. 7, No. 1, pp. 28~35, March 2019.
- [31] Abikoye OC et al, "Modified Advanced Encryption Standard Algorithm for Information Security," Symmetry, MPDI, Vol. 11, no. 12, 2019.
- [32] Selimis et al " A Low Power Design for Sbox Cryptographic Primitive of Advanced Encryption Standard for Mobile End-Users," Journal of Low Power Electronics, Vol.3, pp.327-336, 2007.
- [33] A. Kumar et al, "Performance Analysis of AES Algorithm with Varying Key Sizes", International Journal of Computer Applications, vol. 182, no. 10, 2019.
- [34] L. Zhang et al, "Enhancing Avalanche Effect in AES Through Modified Key Scheduling", Journal of Information Security Research, vol. 12, no. 3, pp. 45–52, 2021.
- [35] S. Al-Faiz et al, "Integration of Lightweight Block Ciphers with AES for Improved Diffusion", IEEE Transactions on Information Forensics and Security, vol. 18, no. 1, pp. 120–128, 2023.