SHM PUBLISHER
SURYA HIJAU MANFAAT

Journal of Soft Computing Exploration

Homepage: https://shmpublisher.com/index.php/joscex

JOSCEX

p-ISSN: 2746-7686
e-ISSN: 2746-0991

.

A text security evaluation based on the advanced encryption standard algorithm

Aristides Bima[1], Candra Irawan[2], Deddy Award Widya Laksana[3], Andi Danang Krismawan[4], Folasade Olubusola Isinkaye[5]
[1, 2, 3, 4]Department of Informatics, Universitas Dian Nuswantoro, Indonesia
[5]Department of Computer Science, Ekiti State University, Nigeria

Article Info

Article history:

Received December 11, 2023
Revised December 20, 2023
Accepted December 27, 2023

Keywords:

Security
AES
BER
CER
Avalanche effect

ABSTRACT

This research approach analyzes a number of advanced encryption standard (AES) performance factors, including encryption and decryption speed, processing resource, consumption, and resilience to cryptanalysis attacks. The study findings demonstrate that AES is successful in providing high-level data security, particularly when used in the CBC (Cipher Block Chaining) operating mode. Performance is dependent on the length of the key that is utilized. Increasing the level of security through the use of longer keys may result in an increase in the amount of time needed for encryption. The experimental results show that the highest results from the data are as follows: the length of the encryption time is 0.00005317 seconds, the length of the decryption time is 0.00000882 seconds, the results of BER and CER are 0, the results of entropy are 7.44237, and the results of avalanche influence are 54.86%.

This is an open-access article under the CC BY-SA license.

Corresponding Author:

Aristides Bima,
Study Program in Informatics Engineering
University of Dian Nuswantoro
Imam Bonjol 207, Semarang, 50131, Indonesia
Email: aristidesbima@gmail.com
https://doi.org/10.52465/joscex.v4i4.274

1. INTRODUCTION

Data security is more critical than ever amid the advanced that is tenaciously shaping society and business. While the progressively broad and frequent business of data information through advanced channels has numerous benefits, it also has noteworthy security protections. In spite of the fact that advances in information technology continue to increase the complexity of managing with progressively advanced cyber dangers, they open up many opportunities for advancement [1], [2]. Cryptography [3]–[6] could be a field that examines scientific strategies related to the angles of information and data security, such as information reliability and authentication. Cryptography studies how communication is carried out so that individuals as it are have the correct to use or access the data sent and depends intensely on numerical theory and computer applications [7]. Cryptographic calculations are built with computational vigor suspicions that make it especially difficult for enemies to crack them. Breaking a cryptographic framework is exceptionally difficult, some theoretically, which makes it not doable to do it in a simple way. This conspiracy is considered exceptionally computationally secure since there are openings for theoretical propels to progress numbers factorization calculations and computational innovations that require nonstop alterations to these arrangements. There are a few information security plans that are indeed invulnerable to boundless computing, but they are exceptionally difficult to realize.

Encryption innovation has developed as the key to maintaining information privacy and passion in the midst of expanding data security and privacy issues. Progressed Encryption Standard (AES) [8], [9] is the most commonly utilized and dependable symmetric key algorithm which could be a continuation of the Information Encryption Standard (DES) calculation. It comprises three pieces of encodings that AES employs, specifically AES-128, AES-192, and AES-256, which are taken from a larger collection to begin with distributed as Rijndael [4]. Each cipher includes a measure of 128 bits, and the key sizes are 128, 192, and 256 bits, individually [10]. AES was founded by the National Institute of Guidelines and Innovation (NIST) in 2001 making it a top choice in numerous businesses, such as communications, money management, and healthcare. AES' victory as a standard encryption standard for more than twenty years illustrates its control and capacity to respond to advancing cyber dangers.

The most important point of this inquiry is to detail and completely analyze the AES calculation within the setting of data security and give a more profound understanding of AES. This understanding is used as a premise for executing more reliable innovation and superior data security arrangements. More than fair a theoretical investigation, but it makes a difference to get it the security issues that worldwide society faces whereas defending individual information within the digital time. The internal mechanisms that guarantee information security incited the determination of AES as the main subject of this inquiry. Unraveling the nuts and bolts of AES security, one can learn its complex and in-depth security foundations.

2. METHOD

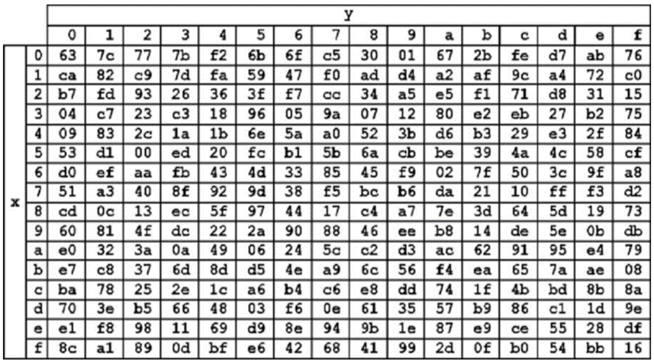
Data Collection

The information collection procedure utilized is the writing audit. The writing survey or writing audit could be an information collection procedure through articles or diaries that involves investigation and inquiry related to the subject. The point of this strategy is to get a conceptual system, theoretical premise, and setting and to discover past investigation that back or shape the premise of this investigation. Different references related to articles and diaries containing cryptography and AES calculation. Analysis is carried out to decide the level of security through the design of characters that shape into irregular composing when encrypting and the time required when encrypting and decrypting. Not as it were encryption, this investigate too looks for how regularly there are errors at the bit level called Bit Error Rate (BER), how frequently the error rate at the character level is called Character Error Rate (CER), how certain the level of randomness in data is called Entropy, and changes to one bit within the input can cause changes within the yield preparation, which is called the Avalanche Effect.

AES Algorithm Encryption Procedure

AES is celebrated for its speed and security which comes from information that is at that point encrypted utilizing complex block cipher strategies. Furthermore, since it employs less processing power, AES is competent in implementing faster encryption strategies compared to others. The change cipher employs a round of encryption as do numerous other square ciphers [11]–[13]. Separate the entered plain content into pieces consisting of four rows and four columns. Each box has one byte, so the piece consists of 16 bytes. Each loop has different development steps that work together to create a function that is rehashed over and over again. The key length in AES decides the number of rounds that can be performed. The AES-128 bit performs 10 turns, the AES-192 bit performs 12 turns, and the AES-256 bit performs 14 turns. FIPS 197 states that round keys are made from cryptographic keys using the Rijndael key schedule. After the final AES period, the state makes a ciphertext that is diverse from the normal text. Decryption of the ciphertext uses the same symmetric key utilized during encryption [12], [14]–[17]. AES uses a substitution-permutation arrangement, in which the cryptographic key from the input column, as shown in Figure 1, which is ordinarily called plaintext, is prepared in a mathematical arrangement employing a substitution box or permutation box.

Data are moved from their original storage location and diffusion occurs. The aim is to design a wrapper that inserts the data location into the appropriate row. Do not change the first line. However, the second line shifts the bytes one step to the left. The third row shifts two steps to the left and the fourth row shifts three steps to the left, as shown in Figure 2. During this phase, AES combines the bytes for a process of mathematical operations known as linear transformations. Each column performs the function of the XOR algorithm in a complex and mathematical way by combining all four bytes values to produce four new bytes as output as shown in Figure 3. When encryption is performed, this is the first and last operation applied to the status array. This step is the most important step in the algorithm. By using a special key schedule of the Rijndael algorithm that forms the basis of AES, a new key is presented in each round. The result of the merger column for the keys in the first round. Then we go back to SubBytes, and the whole process starts over, as shown in Figure 4.



		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Figure 1. SubBytes s-box [18]

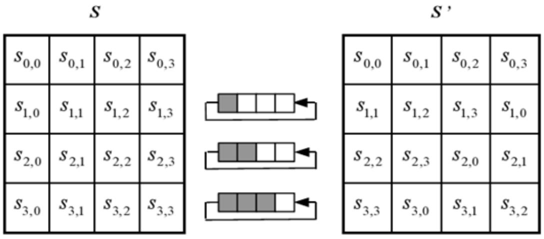

S
S '
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{1,1}$ $s_{1,2}$ $s_{1,3}$
$s_{2,0}$ $s_{2,1}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,1}$ $s_{3,2}$ $s_{3,3}$
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,1}$ $s_{1,2}$ $s_{1,3}$ $s_{1,0}$
$s_{2,2}$ $s_{2,3}$ $s_{2,0}$ $s_{2,1}$
$s_{3,3}$ $s_{3,0}$ $s_{3,1}$ $s_{3,2}$

Figure 2 Shiftrows process [18]

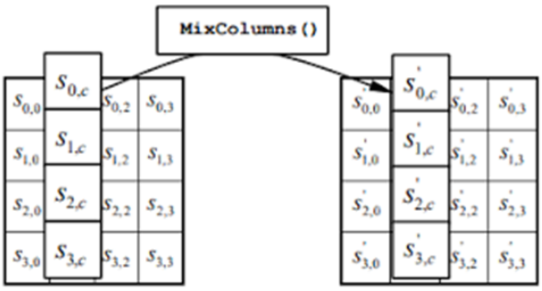

MixColumns()
$s_{0,0}$ $s_{0,c}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{1,c}$ $s_{1,2}$ $s_{1,3}$
$s_{2,0}$ $s_{2,c}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,c}$ $s_{3,2}$ $s_{3,3}$
$s'_{0,0}$ $s'_{0,c}$ $s'_{0,2}$ $s'_{0,3}$
$s'_{1,0}$ $s'_{1,c}$ $s'_{1,2}$ $s'_{1,3}$
$s'_{2,0}$ $s'_{2,c}$ $s'_{2,2}$ $s'_{2,3}$
$s'_{3,0}$ $s'_{3,c}$ $s'_{3,2}$ $s'_{3,3}$

Figure 3 Mix column process [18]

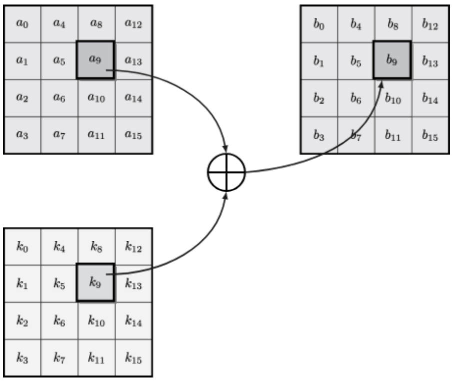

a_0 a_4 a_8 a_{12}
a_1 a_5 a_9 a_{13}
a_2 a_6 a_{10} a_{14}
a_3 a_7 a_{11} a_{15}
b_0 b_4 b_8 b_{12}
b_1 b_5 b_9 b_{13}
b_2 b_6 b_{10} b_{14}
b_3 b_7 b_{11} b_{15}
k_0 k_4 k_8 k_{12}
k_1 k_5 k_9 k_{13}
k_2 k_6 k_{10} k_{14}
k_3 k_7 k_{11} k_{15}

Figure 4 Add round key process [18]

.

Bit Error Rate (BER)

Bit error rate (BER) is a metric that measures how often errors occur in bits. BER is calculated as the ratio of bad bits to the total number of bits transmitted and received [19], [20]. BER is typically expressed using percentage notation or exponential notation, such as 10^(-6). There is no direct relationship between BER and AES-256 encryption and decryption because AES includes a symmetric cryptography algorithm to correct bit errors to maintain data confidentiality. However, bit errors can occur when transmitting or storing data in systems that use AES-256 as a communication component. Formula for Bit Error Rate (BER) as illustrated in (1). For example, if 50 bits are sent and it turns out that there are 5 wrong bits, the BER result will be BER $= \frac{5}{50} = 0.1.$

$$BER = \frac{Number\ of\ error\ bits}{Total\ of\ bits} \tag{1}$$

Character Error Rate

This is almost the same as BER, only the word character is different. In other words, the character error rate (CER) is a metric that measures how often characters between two text strings are erroneous [21]. This metric is often used in the context of optical character recognition (OCR), handwriting recognition, or natural language processing, where character relationships between texts are very important. Formula for Character Error Rate (CER) as illustrated in (2).

$$CER = \frac{Number\ of\ error\ characters}{Total\ of\ characters} \tag{2}$$

For example, if you have two text strings where the starting string is "@YunD4" and the result is "Ru5", the CER result will be if:

a) Detected characters. The characters that should not be in the results are none. The number of characters detected is 0.
b) Deleted characters. The character that should be in the result string but is not is “@YnD4”. The number of characters deleted was 5.
c) Entered characters. Characters that should not be present in the results, but are detected are none. The number of characters entered is 0. The number of characters in the string is 6.

then the $CER\ is\ \frac{5}{6} = 0.8333.$

Entropy

Entropy is an ambiguous measurement in a system. In cryptography, entropy is typically used to evaluate the degree of randomness of a key or source [22]. The higher the entropy value, the more difficult it is for third parties to deduce or access the encrypted data or keys. Formula for entropy as illustrated in (3).

$$H(X) = -\sum_{i=1}^{n} P\,(x_i) log_2(P(xi)) \tag{3}$$

For example, if there is a key "11110100", then the entropy result is :

a) Total bits are 8
b) The number of bits with a value of "1" is 5, then $P(1) = \frac{5}{8}$
c) The number of bits with the value "0" is 3, then $P(0) = \frac{3}{8}$
d) $H(X) = -(P(1).log_2\left(P(1)\right) + P(0).log_2(P(0)))$ = $H(X) = -\left(\frac{5}{8}.log_2\left(\frac{5}{8}\right) + \frac{3}{8}.log_2\left(\frac{3}{8}\right)\right)$ = $H(X) = -\left(\frac{5}{8}.(-0.678) + \frac{3}{8}.(-1.415)\right)$ = $H(X) = -(-0.423) + (-0.531) = 0.954$

Avalanche Effect

The avalanche effect in cryptography is the idea that small changes in the input to an algorithm can cause large changes in the final result. More specifically, the avalanche effect describes all bits that change in the input that affect the final result, such that all bits of the final result are sensitive to each bit of the input [23]–[25]. The properties of the avalanche effect are highly desirable for cryptographic algorithms, especially for creating ciphers that are resistant to cryptanalysis attacks. If an encryption algorithm has a strong avalanche effect, a slight change in the plaintext or key can produce a different ciphertext. Formula for the avalanche effect as illustrated in (5). For example, if there are 6 bit errors with a bit length of 30, then the percentage of $Avalanche\ Effect = \frac{6}{30}\ X\ 100\% = 20\%$.

Research Scheme

Processes are created using activity diagrams and flow chart representations. Two graphical representations are used to represent the processes or flows of a system. Both serve the same purpose, but the main difference is how the information is displayed. The representations of the process flow diagram are shown in Figures 5 and Figure 6. This is a representation of the process flow to perform encryption and decryption with the AES-256 algorithm. In Figure 4, the encryption process begins with plaintext and key input, then performs Key Expansion, and enters the AddRoundKey phase.

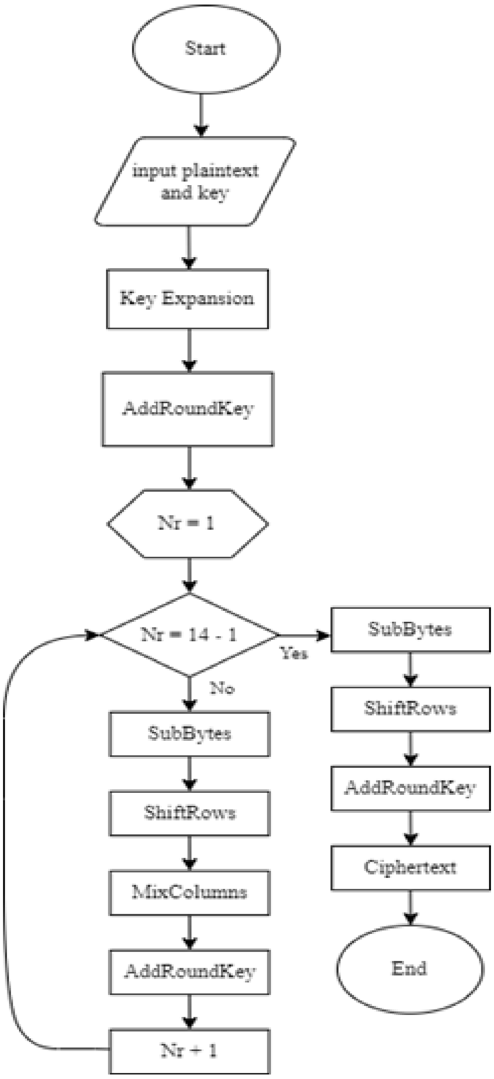


Figure 5. Flow chart of the AES encryption process

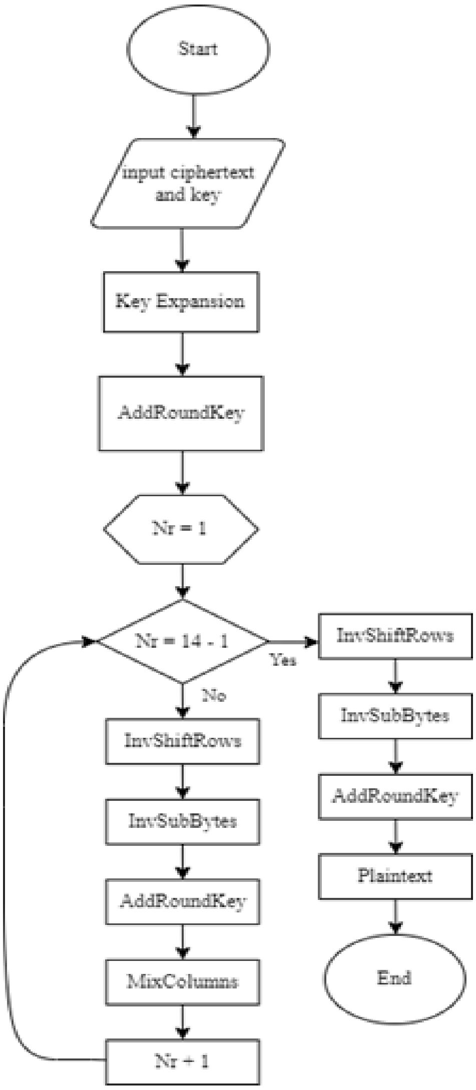


Figure 6 Flowchart of the AES decryption process

The process begins with a round count phase, starting at 1. If the rotation has not reached 13 times, it enters the SubBytes, ShiftRows, MixColumns, and AddRoundKey stages. The number of revolutions is then increased by 1 until the number of revolutions increases by a factor of 13. Once 13 rounds are reached, the next stage consists of SubBytes, ShiftRows, AddRoundKey, and ciphertext generation. In Figure 5, the decryption process begins by entering the plaintext and key, performs key expansion, and enters the AddRoundKey phase. The decryption process starts with a round-number phase starting from 1. If the rotation has not reached 13x, the InvShiftRows, InvSubBytes, AddRoundKey, and MixColumns levels are entered, and the number of rotations is increased by one. This process is repeated until 13 rotations are reached. After 13 rotations, the next stage is to generate InvShiftRows, InvSubBytes, AddRoundKey, and plain text. When the program is run from the website, the activity diagram is illustrated in Figure 6 and the flowchart was illustrated in Figure 7.

In Figure 6, the process of using image images from activity diagrams in a program begins with the user entering the program. The processing system displays the program page to the user. The user enters the plaintext and key provided by the system. If the plaintext is not entered, the system forces the user to enter it. The same applies to keys. If these are not entered, the system will force the user to enter a key. Once the plaintext and keys are entered, the system processes them and displays the results sent by the system once the process is complete. In Figure 7, the program starts processing from the flowchart image within the program. Then enter the plain text and the key. If the plaintext is not entered, the system forces the user to enter it. Similarly, if a key is not entered, it will be forced to be entered. If you enter both plain text and a key, the results will be displayed in the program.

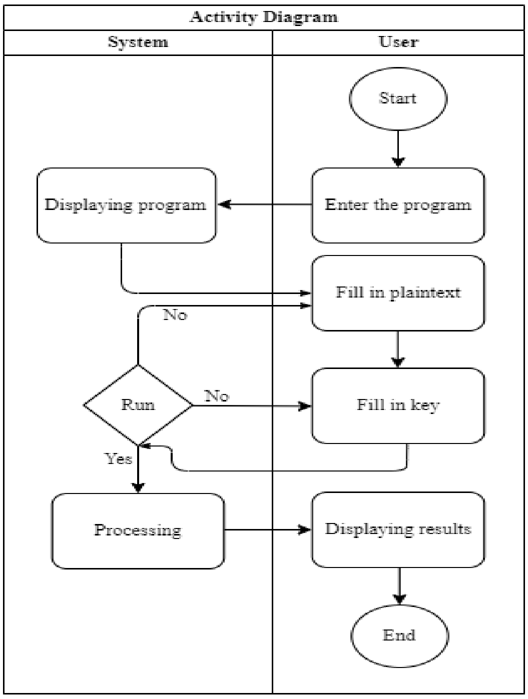


Figure 6. Activity diagram of the proposed interface

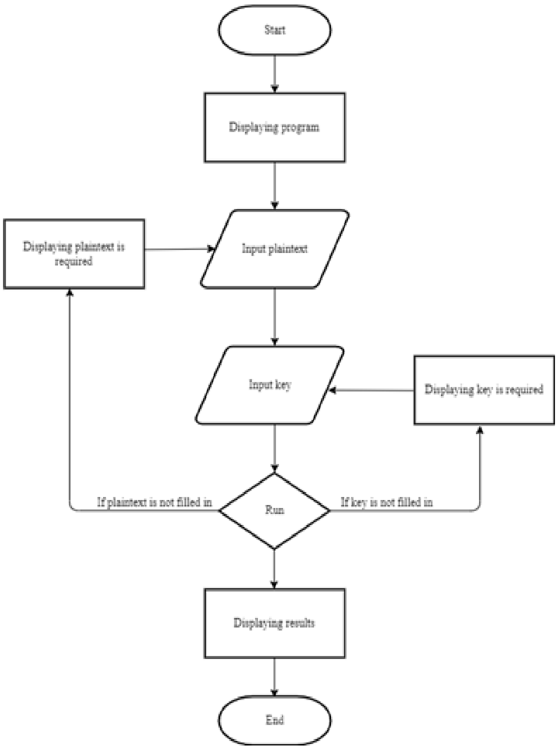
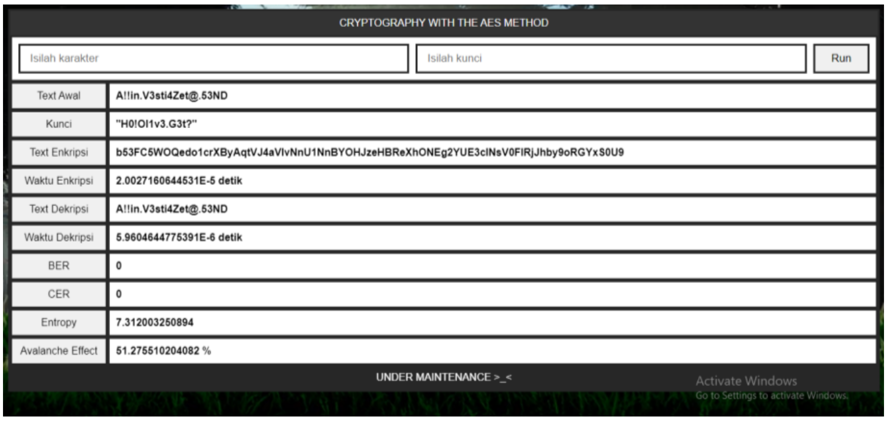


Figure 7. The flow chart of interface

.

3. RESULTS AND DISCUSSIONS

This study was created using the PHP programming language and xampp to retrieve output results from a localhost website. This data security implementation takes the form of a plain text input and a key, and when executed, displays the results shown in Figure 8. Figure 8 of the above results shows the output when the plain text and key are input. The result is the specified initial text, including the key, ciphertext, encryption time length (seconds), decrypted text and decryption time length (seconds), BER, CER, entropy, and avalanche effects. If the plaintext or key is empty, an image similar to Figure 9 is displayed and the user must enter the plaintext or empty key. Some of the data results that have been obtained when filling in, for example, the plaintext is A!!in.V3sti4Zet@.53ND and the key is "H0!Ol1v3.G3t?" by not changing the plaintext and key during the process, The results are used to determine encryption time, decryption time, Bit Error Rate (BER), Character Error Rate (CER), entropy, and avalanche effects.



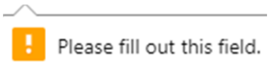
CRYPTOGRAPHY WITH THE AES METHOD

Text Awal	A!!in.V3sti4Zet@.53ND
Kunci	"H0!Ol1v3.G3t?"
Text Enkripsi	b53FC5WOQedo1crXByAqtVJ4aVlvNnU1NnBYOHJzeHBReXhONEg2YUE3clNsV0FIRjJhby9oRGYxS0U9
Waktu Enkripsi	2.0027160644531E-5 detik
Text Dekripsi	A!!in.V3sti4Zet@.53ND
Waktu Dekripsi	5.9604644775391E-6 detik
BER	0
CER	0
Entropy	7.312003250894
Avalanche Effect	51.275510204082 %

UNDER MAINTENANCE >_<

Activate Windows
Go to Settings to activate Windows.

Figure 8. Output results



Please fill out this field.

Figure 9. Pop-up information when plaintext and key are empty.

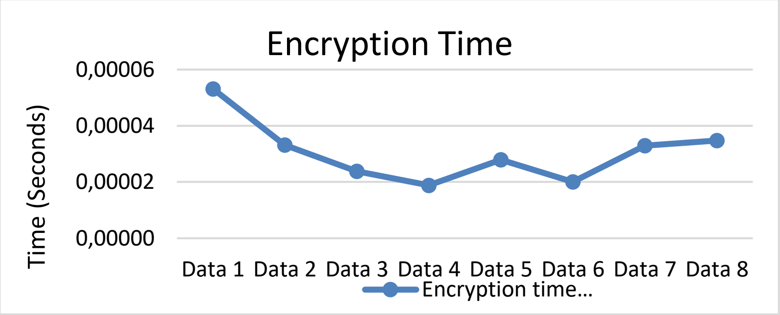


Figure 10. Encryption-time results

Based on the results of the above data, we will explain the results of the encryption time when the same plaintext and key are used eight times. The longest encryption time for data 1 was 0.00005317 seconds or 5317 ×10^(-8) seconds. The encryption time for data 2 is 0.00003314 seconds or 3314 ×10^(-8) seconds, which is shorter than data 1. Data 3 is reduced in number and receives an encryption time of 0.00002384 seconds or 2384 ×10^(-8) seconds. The length of the encryption time result for data 4 decreased and reached the minimum result of 0.00001884 seconds or 1884 ×10^(-8) seconds. The length of encryption time for data 5 has increased to 0.00002789 seconds 2789 ×10^(-8) seconds. For data 6, the encryption time is 0.00002003 seconds or 2003 ×10^(-8) seconds and the callbacks are reduced. Data 7 result in a duration of 0.0000329 seconds or 3290 ×10^(-8) seconds, which significantly exceeds Data 6. The result for Data 8 increases the number by 0.00003481 seconds or 3481 ×10^(-8) seconds. The average of all the data that can be collected in Figure 10 is 0.0000305625 seconds or 305625 ×10^(-10) seconds. If the plaintext and key parts do not change, one can conclude that the result changes over the long period of encryption and is not constant.

According to the results of the data above, using the same plaintext and key 8 times can explain the long decryption time results. The decryption time for data 1 is 0.00000810 seconds, or 810 ×10^(-8) seconds. Data 2 increased over Data 1 and became the highest data with a decryption time of 0.00000882 seconds, or 882 ×10^(-8) seconds. Data 3 reduced the decryption time to 0.00000691 seconds or 691 ×10^(-8) seconds.

The results for data 4 were also reduced by the long decryption time of 0.00000596 seconds or 596 ×10^(-8) seconds. Data 5 was successfully acquired with a decryption time of 0.00000619 seconds or 619 ×10^(-8) seconds. Data 6 showed an increase in decryption times as long as 0.00000786 or 786×10^(-8) seconds. Data 7 shows a decrease, with a decryption time of 0.00000691 seconds or 692×10^(-8). The result for data 8 is the lowest data with a long decryption time of 0.000005 or 500×10^(-8). The average of the eight data points in Figure 11 is 0.00000697 seconds or 697×10^(-8) seconds. We can conclude that if the provided plaintext and key remain unchanged, the decryption time results will change and remain unchanged. Based on the results for the data in Figure 12, let us discuss the BER and CER results obtained when using the same plaintext and key 8 times. The results of data 1 to data 8 are numbered 0. The average value obtained from the eight data points in Figure 12 is 0. We can conclude that the BER and CER results are constant and do not change.

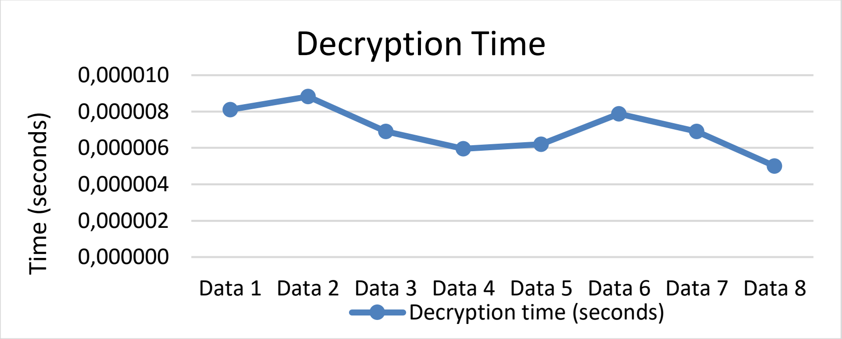


Figure 11. Results of decryption time

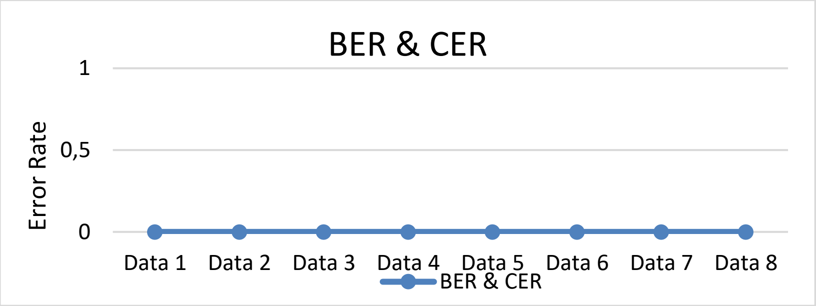


Figure 12. BER and CER results

Consistent with the results in the data above, it describes the entropy results found when using the same plaintext and key eight times. The entropy result for data 1 is 7.27019. For data 2, the entropy result decreased to 7.20132. For data 3, the entropy result also decreased by 7.10132. Data 4 increased rapidly and reached the highest result of 7.44237. After the increase, the entropy result decreased by 7.23451 in data 5. Data 6 recorded a slight increase in the number of 7.25257. Data 7 clearly recorded a decrease in the entropy result of 7.03906. Data 8 is the result of reducing data 7, and data 8 results in the lowest data of 7.0312. The average of all data in Figure 13 is 7.196567. We can conclude that the entropy result does not change the plaintext and the key, but it does change the result. Based on the data in Figure 13, describe the results of the avalanche effect results that occur when the same plaintext and key are used eight times. The result of the avalanche effect from data 1 was found to be 53.54%. Data 2 show that the avalanche effect was reduced by 53.13%. In Data 3, the percentage also decreased to 52.49%. Data 4 showed a rapid increase and reached the highest result of 54.86%. Data 5 immediately experienced a sharp drop and became the lowest data result with a rate of 51.63%. Data 6 show a further increase, with an avalanche effect of 53.99%. The decrease occurred in data 7, reaching 52.46%. Data 8 are the latest data with a maximum decline of 51.75%. The average percentage of numbers that can be generated from the eight data in Figure 14 is 52.98%. We can conclude that even for the same plaintext or key, the results can vary due to the avalanche effect. A total of eight ciphertext variations can be created if using plaintext like A!!in.V3sti4Zet@.53ND and a key like "H0!Ol1v3.G3t?".

Based on the data in Figure 13, describe the results of the avalanche effect results that occur when the same plaintext and key are used eight times. The result of the avalanche effect from data 1 was found to be

.

53.54%. Data 2 show that the avalanche effect was reduced by 53.13%. In Data 3, the percentage also decreased to 52.49%. Data 4 showed a rapid increase and reached the highest result of 54.86%. Data 5 immediately experienced a sharp drop and became the lowest data result with a rate of 51.63%. Data 6 show a further increase, with an avalanche effect of 53.99%. The decrease occurred in data 7, reaching 52.46%. Data 8 are the latest data with a maximum decline of 51.75%. The average percentage of numbers that can be generated from the eight data in Figure 14 is 52.98%. We can conclude that even for the same plaintext or key, the results can vary due to the avalanche effect. A total of eight ciphertext variations can be created if using plaintext like A!!in.V3sti4Zet@.53ND and a key like "H0!Ol1v3.G3t?".

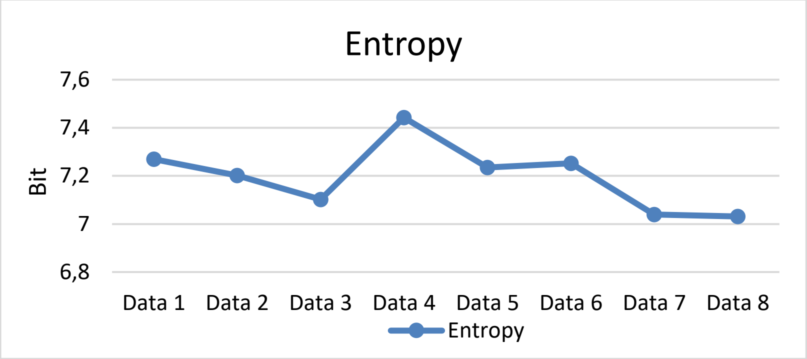


Figure 13. Entropy results

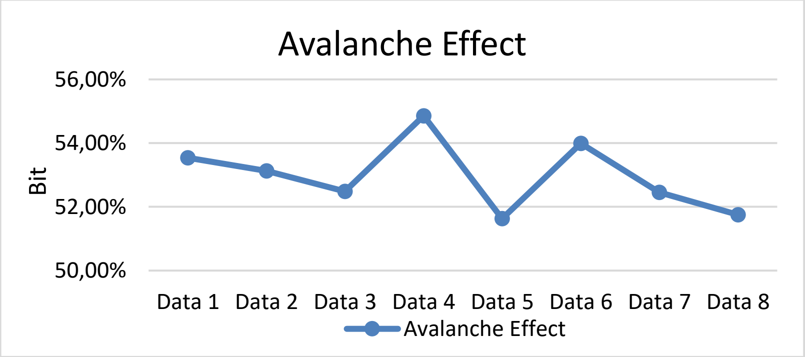


Figure 14. Avalanche effect results

Table 2. Ciphertext variations

No	Plaintext	Key	Ciphertext
1	A!!in.V3sti4Zet@.53ND	"H0!Ol1v3.G3t?"	PSmJyfxXRoM1gKxvwHV38TZPdjlVeEwxNmd3NEtKVGlKbnRRRjZQRUQ1QWVxUWVqS01RazlnRmE2N3M9
2	A!!in.V3sti4Zet@.53ND	"H0!Ol1v3.G3t?"	Bp2R/1lzMCVywavmhhc9zEZsMFdwTnNHU1o0WDZsd3Rza1p6eG9WT1Fib2lWM2I3MERWK2tVeUtXZEE9
3	A!!in.V3sti4Zet@.53ND	"H0!Ol1v3.G3t?"	RAJY4Xw1Lkp9xXFb2giUnzVmK1JYemx2aGgyNDJTZjNzUDBjeGErWkVZVDkzSUVTMGlncFE4cXRqWkU9
4	A!!in.V3sti4Zet@.53ND	"H0!Ol1v3.G3t?"	eMx0G3g6U91qzonavJHqiWliUlBCREFMUCtHeGtneDl1bVEzNTY2SXpYZzJQUXdZR1BmOGs0WkJJbms9
5	A!!in.V3sti4Zet@.53ND	"H0!Ol1v3.G3t?"	zfBt+ic5GTCvD48ZF9mILjNqM3JUWXhnUXZoTWpZT2NTbWZZWitVQzZCZytQbkxiWjVDc1BwK2x4OEk9
6	A!!in.V3sti4Zet@.53ND	"H0!Ol1v3.G3t?"	Vr54wLe8leJPuBLzSkMgmXNHNnpZamZ4WS9IYTZ3RUJDNFBpcGRWTCtBYjNUTkFmNFRyYmdZQTdCZEU9
7	A!!in.V3sti4Zet@.53ND	"H0!Ol1v3.G3t?"	Stv++xlAUAFvVlksRGhsPE1wYTdYRzVidTJ2aDhqSU41QWwvbFdleVNKaDh4NFMwbEIreE95RllrZXM9
8	A!!in.V3sti4Zet@.53ND	"H0!Ol1v3.G3t?"	94LTRAhR4IrszeCVNHPsKnNiQ1JkSlVwamZ2ejhlWjlDNlRKTjdrZUJkNXlsMEZYT2Exb1RFQUQvQ3c9

Based on the ciphertext variation in Table 2 above, this is due to the ciphertext, which may differ depending on the key length even if the plaintext and key are the same. AES-256 has a 256-bit long key and is compatible with large bit sizes that are difficult to crack. The combinations you can create depend on the size of the key. There are more, such as the number of rounds of AES encryption. AES-128 uses 10 rounds, AES-192 uses 12 rounds, and AES-256 uses 14 rounds. Doing many rounds can make the resulting ciphertext more complex. There are additional elements in the encryption process, such as an initialization vector (IV). Even the same plaintext and key can produce different ciphertexts. The properties represented in Table 2 are probabilistic encryption. An encryption model known as probabilistic encryption allows the same message to be encrypted into a different ciphertext each time the encryption process is performed. That is, if a message containing the same plaintext and key is used repeatedly, the ciphertext will be different from the others. The goal of probabilistic encryption is to prevent attacks on the encryption. In this attack, it is possible to obtain information by analyzing the difference between the same plaintext and the ciphertext obtained by encryption of the key.

Table 3. Combinations of keys

Key Size	Possible Combinations
1 bit	2
2 bit	4
4 bit	16
8 bit	256
16 bit	65536
32 bit	X 4.2 10^{9}
56 bit	X 7.2 10^{16}
64 bit	X 1.8 10^{19}
128 bit	X 3.4 10^{38}
192 bit	X 6.2 10^{57}
256 bit	X 1.1 10^{77}

4. CONCLUSION

The conclusion drawn from this study based on the abstract, introduction, method, research scheme, results, and discussion is that AES is a reliable symmetric key algorithm and is a continuation of the DES algorithm, and there are three AES block encodings about it, namely AES-128, AES-192, and AES-256. How it works depends on the keys you use. The longer the encryption takes, the more secure the security level. The AES key length also determines the number of rounds performed, example, AES-128 bit (10 rounds), and AES-192 bit (12 rounds), AES-256 bit (14 rounds). Each AES encryption process starts with a SubBytes. Next, we have ShiftRows, then MixColumns, and finally AddRoundKey. The AES encryption process is conveyed through a representation using activity diagrams and flow charts that show the flow of the system. The time required to view the results of the encryption and decryption processes is obviously not the same. The data results show that the decryption time is fast compared to the encryption time. It turns out that by using the same initial text and key, we can change the ciphertext.

REFERENCES

[1] A. K. Agrahari, M. Sheth, and N. Praveen, "Comprehensive Survey on Image Stegnography Using LSB With AES," *Int. J. Appl. Eng. Res.*, vol. 13, no. 8, pp. 5841–5844, 2018.

[2] D. M. Kuryazov, "Development of electronic digital signature algorithms with compound modules and their cryptanalysis," *J. Discret. Math. Sci. Cryptogr.*, vol. 24, no. 4, pp. 1085–1099, May 2021, doi: 10.1080/09720529.2021.1878628.

[3] R. Marqas, S. M. Almufti, and R. Rebar, "Comparing Symmetric and Asymmetric cryptography in message encryption and decryption by using AES and RSA algorithms," *J. XI'AN Univ. Archit. Technol.*, vol. XII, no. III, Mar. 2020, doi: 10.37896/JXAT12.03/262.

[4] T. Kumar, K. Reddy, S. Rinaldi, B. Parameshachari, and K. Arunachalam, "A Low Area High Speed FPGA Implementation of AES Architecture for Cryptography Application," *Electronics*, vol. 10, no. 16, p. 2023, Aug. 2021, doi: 10.3390/electronics10162023.

[5] K. Patel, "Performance analysis of AES, DES and Blowfish cryptographic algorithms on small and large data files," *Int. J. Inf. Technol.*, vol. 11, no. 4, pp. 813–819, Dec. 2019, doi: 10.1007/s41870-018-0271-4.

[6] A. U. Rohmah and D. Djuniadi, "Tiny encryption algorithm (TEA) for analysis and implementation of cryptool2-based text message encryption and decryption processes," *J. Student Res. Explor.*, vol. 1, no. 1, pp. 33–40, Dec. 2022, doi: 10.52465/josre.v1i1.111.

[7] E. Y. Purba, S. Efendi, P. Sirait, and P. Sihombing, "Collaboration of RSA Algorithm Using EM2B Key with Word Auto Key Encryption Cryptography Method in Encryption of SQL Plaintext Database," *J. Phys. Conf. Ser.*, vol. 1230, p. 012009, Jul. 2019, doi: 10.1088/1742-6596/1230/1/012009.

[8] E. H. Rachmawanto, R. S. Amin, D. R. I. M. Setiadi, and C. A. Sari, "A performance analysis StegoCrypt algorithm based on LSB-AES 128 bit in various image size," in *2017 International Seminar on Application for Technology of Information and Communication (iSemantic)*, IEEE, Oct. 2017, pp. 16–21. doi: 10.1109/ISEMANTIC.2017.8251836.

[9] A. Alamsyah, B. Prasetiyo, and Y. Muhammad, "S-box Construction on AES Algorithm using Affine Matrix Modification to Improve Image Encryption Security," *Sci. J. Informatics*, vol. 10, no. 2, pp. 69–82, Apr. 2023, doi: 10.15294/sji.v10i2.42305.

[10] N. Sharma, Prabhjot, and H. Kaur, "A Review of Information Security using Cryptography Technique," *Int. J. Adv. Res. Comput.*

Sci., vol. 8, no. 4, pp. 323–326, 2017, doi: https://doi.org/10.26483/ijarcs.v8i4.3760.

[11] A. Ajmera, S. S. Ghosh, and T. Vijayetha, "Secure LSB Steganography over Modified Vigenère-AES Cipher and Modified Interrupt Key-AES Cipher," in *2018 IEEE Punecon*, IEEE, Nov. 2018, pp. 1–7. doi: 10.1109/PUNECON.2018.8745393.

[12] P. P. Bandekar and G. C. Suguna, "LSB Based Text and Image Steganography Using AES Algorithm," in *2018 3rd International Conference on Communication and Electronics Systems (ICCES)*, IEEE, Oct. 2018, pp. 782–788. doi: 10.1109/CESYS.2018.8724069.

[13] G. N. Salmi and F. Siagian, "Implementation of the data encryption using caesar cipher and vernam cipher methods based on CrypTool2," *J. Soft Comput. Explor.*, vol. 3, no. 2, Sep. 2022, doi: 10.52465/joscex.v3i2.86.

[14] C. A. Sari, G. Ardiansyah, D. R. I. M. Setiadi, and E. H. Rachmawanto, "An improved security and message capacity using AES and Huffman coding on image steganography," *TELKOMNIKA (Telecommunication Comput. Electron. Control.*, vol. 17, no. 5, p. 2400, Oct. 2019, doi: 10.12928/telkomnika.v17i5.9570.

[15] Y. Alemami, M. A. Mohamed, and S. Atiewi, "Advanced approach for encryption using advanced encryption standard with chaotic map," *Int. J. Electr. Comput. Eng.*, vol. 13, no. 2, p. 1708, Apr. 2023, doi: 10.11591/ijece.v13i2.pp1708-1723.

[16] U. Banerjee, S. Das, and A. P. Chandrakasan, "Accelerating Post-Quantum Cryptography using an Energy-Efficient TLS Crypto-Processor," in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, IEEE, Oct. 2020, pp. 1–5. doi: 10.1109/ISCAS45731.2020.9180550.

[17] K. Muttaqin and J. Rahmadoni, "Analysis And Design of File Security System AES (Advanced Encryption Standard) Cryptography Based," *J. Appl. Eng. Technol. Sci.*, vol. 1, no. 2, pp. 113–123, May 2020, doi: 10.37385/jaets.v1i2.78.

[18] M. Alwazzeh, S. Karaman, and M. N. Shamma, "Man in The Middle Attacks Against SSL/TLS: Mitigation and Defeat," *J. Cyber Secur. Mobil.*, Jul. 2020, doi: 10.13052/jcsm2245-1439.933.

[19] D. Kuswanto and A. Rachmad, "COMBINATION SCHEME OF AES ENCRYPTION AND ERROR CORRECTION TURBO CODE FOR CRYPTOGRAPHY OF CLOUD STORAGE," in *Proceedings of the International Conference on Science and Technology (ICST 2018)*, Paris, France: Atlantis Press, 2018. doi: 10.2991/icst-18.2018.146.

[20] D. Kuswanto, "Performances Combination Schemes AES-Turbo Code Based-on Keys Length," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1125, no. 1, p. 012047, May 2021, doi: 10.1088/1757-899X/1125/1/012047.

[21] H. K. Ronaldo Cahyono, C. Atika Sari, D. R. Ignatius Moses Setiadi, and E. Hari Rachmawanto, "Dual Protection on Message Transmission based on Chinese Remainder Theorem and Rivest Cipher 4," in *2019 International Conference on Information and Communications Technology (ICOIACT)*, IEEE, Jul. 2019, pp. 74–78. doi: 10.1109/ICOIACT46704.2019.8938568.

[22] M. T. Elkandoz, W. Alexan, and H. H. Hussein, "Double-Layer Image Security Scheme with Aggregated Mathematical Sequences," in *2019 International Conference on Advanced Communication Technologies and Networking (CommNet)*, IEEE, Apr. 2019, pp. 1–7. doi: 10.1109/COMMNET.2019.8742370.

[23] M. Essaid, I. Akharraz, A. Saaidi, and et A. Mouhib, "Image encryption scheme based on a new secure variant of Hill cipher and 1D chaotic maps," *J. Inf. Secur. Appl.*, vol. 47, pp. 173–187, Aug. 2019, doi: 10.1016/j.jisa.2019.05.006.

[24] R. G. Barrieta, A. S. Canlas, D. M. A. Cortez, and K. E. Mata, "Modified Hill Cipher Algorithm using Myszkowski Transposition to address Known-Plaintext attack," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 10, no. 4, pp. 3242–3249, Apr. 2022, doi: 10.22214/ijraset.2022.41970.

[25] H. V Gamido, "Implementation of a bit permutation-based advanced encryption standard for securing text and image files," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 19, no. 3, p. 1596, Sep. 2020, doi: 10.11591/ijeecs.v19.i3.pp1596-1601.