Enhancing Advanced Encryption Standard with Chaotic Map and Dynamic S-Box

Candra Tangguh Pradipta
Department of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
6025241035@student.its.ac.id

Riyanarto Sarno
Department of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
riyanarto@its.ac.id

Abdullah Faqih Septiyanto
Department of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
7025221022@student.its.ac.id

Dwi Sunaryono
Department of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
dwi@its.ac.id

2024 Beyond Technology Summit on Informatics International Conference (BTS-I2C) | 979-8-3315-0857-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/BTS-I2C63534.2024.10942276

***Abstract*—Advanced Encryption Standard (AES) is one of cryptography algorithms widely used for protecting data across various sectors of technology. Although widely used, there are still some vulnerabilities in AES, notably the fixed S-Box and the lack of a MixColumn step in the final encryption round, which limit its resistance against specific cryptographic attacks. This paper proposes a new enhanced AES algorithm targeting 128-bit key encryption, including the introduction of dynamic S-Box to replace the fixed S-Box, key generation based on Chebyshev chaotic map, and modified encryption process incorporating MixColumn step in the final round and reduction of number of rounds. These modifications aims to increase complexity and improve resilient against decryption against unauthorized parties, while improving the performance of the algorithm. Experimental results demonstrated that the proposed algorithm successfully enhanced both security and performance. The proposed algorithm achieves a higher average avalanche score of 53.02 %, surpassing the standard AES score of 50.00 %. For the encryption time, the proposed algorithm demonstrates faster encryption time across various file sizes, with a reduction of approximately 20% compared to the standard AES.**

Keywords—AES, chaotic map, chebyshev map, cryptography, dynamic S-Box

I. INTRODUCTION

With the rapid growth of data processing technology, data security has become one of the most pressing concerns that must be prioritized across different sectors such as financial services, healthcare, education, even the government [1]. Moreover, data can be accessed remotely by anyone with ease [2]. Data security itself refers to the process of protecting digital information from physical harm or theft, while ensuring its confidentiality and availability remain intact [3].

One of the most commonly used method to ensure data security is cryptography. The word cryptography originates from Greek, meaning "creating secrecy". It is a process of transforming original information, known as plaintext into secret information called ciphertext [4]. Cryptography can be classified into two types : symmetric key encryption and asymmetric key encryption. Symmetric key encryption uses a common key for both encryption and decryption process. While in asymmetric key encryption, two different keys are used : public key for encryption and private key for decryption [5]. In this paper, we focus on one implementation of symmetric key encryption, namely the Advanced Encryption Standard (AES).

AES encryption was developed by Joan Daem and Vincent Rijmen and introduced by National Institute of Standards and Technology (NIST) [6]. In fact, it was the first encryption algorithm by NIST that has been approved and made publicly accessible. AES algorithm has been widely used for various purposes, including secure communication, VPNs, and payment systems as demonstrated in [7], [8], [9]. However, there are certain vulnerabilities in AES system. For example, the use of fixed S-Box can be considered to provide inadequate protection. Another vulnerability includes the absence of MixColumn step in last round which lacks an effective role in security factor [10]. Previous studies [11], [12], [13] have proposed solutions for these vulnerabilites by implementing various modifications to the AES algorithm.

This paper proposes a new enhanced AES algorithm by modifying the encryption process. The proposed algorithm introduces some modifications on the standard AES, including: subtituting static S-Box with dynamic S-Box, shuffling encryption key using chaotic map, reducing the number of rounds, and adding MixColumn step on the last round of encryption.

II. RELATED WORKS

Several previous studies have explored the modifications to the AES algorithm as listed below :

A. AES Encryption

AES is a block cipher algorithm introduced by NIST in 2001 [6]. The algorithm supports three key lengths : 128,192, and 256 bit. This paper focuses on the 128-bit key length encryption. In 128 bit-key encryption, the number of rounds is 10 rounds. The flowchart of standard AES encryption process is as shown in Fig 1.

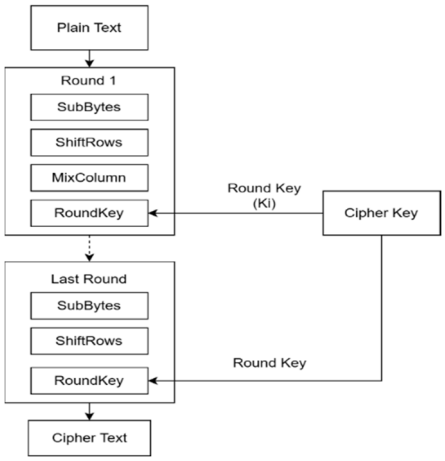

Fig. 1. AES Encryption

AES takes plaintext as input. The plaintext is then encrypted through a sequence of processes as many times as the number of rounds. Every round, with the exception of the last round, includes four steps : SubBytes, ShiftRows, MixColumn, and RoundKey. In the last round, MixColumn step is absent.

B. Dynamic S-Box

Standard AES encryption uses a predefined static Rijndael S-Box in SubByte step. SubByte is a non-linear transformation where each byte in the state undergoes a substitution based on substitution table (S-Box) independently [14]. The sequence of processes in the SubByte operation is as follows: a) 16x16 S-Box is constructed using a combination of mathematical transformations : inversion of Galois Field GF(2^8) and affine transformation. b) Each byte in the state is replaced with the corresponding value from the S-Box. Affine transformation in standard AES is shown in Equation 1, where the obtained value of b undergoes an XOR operation with a constant value equals to 0x63 in hexadecimal.

$$\begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} XOR \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \quad (1)$$

A previous study [11] proposed the substitution of static S-Box with dynamic S-Box. While the static S-Box uses constant hex value of 0x63 (99 in decimal) for XOR operation in affine transformation, the proposed method used a random generated decimal number between 20 to 255. This substitution increased the level of complexity in decryption process by a third party. However, the paper does not explain how the decryption process involving the random generated number is performed.

C. Chaotic Map

Chaotic maps are mathematical functions that produce complex, unpredictable patterns depending heavily on the starting seed value [15]. Chaotic maps exist in various dimensions, ranging from one-dimensional to multi-dimensional.

Several studies have proposed the implementation of chaotic map in the encryption process. A study [11] implemented a 1D chaotic map, called Sine Map for new key generation derived from the original key. Another research [12] combined 2D and 3D chaotic maps for key generation process.

This paper proposes implementation of a 1D chaotic map, named Chebyshev Map. Chebyshev map considered suitable for encryption due to its efficiency and its ability to offer a satisfactory level of security and randomness, thanks to its inherent chaotic properties [16]. Chebyshev Map can be written as Equation 2.

$$T_n(x) = \begin{cases} \cos(n \arccos x) & if\ |x| \le 1 \\ \cosh(n\ arcosh\ x) & if\ x \ge 1 \\ (-1)^n \cosh(n\ arcosh(-x)) & if\ x \le -1 \end{cases} \quad (2)$$

D. MixColumn Step

In a standar AES, the encryption process consists of four steps each round : SubByte, ShiftRow, MixColumn, and Add Round Key; except for the last round where MixColumn is absent. Research by [13] proposed a modification on the last round of encryption by implementing MixColumn. Another research [17] provided a completely different approach regarding this matter. Instead of using MixColumn on each round, this research replaced it with a new process called “permutation”.

III. Proposed Method

We propose a new AES algorithm with modifications on its encryption process as shown in Fig. 2.

A. Modified S-Box

In this study, we propose a dynamic S-Box as a replacement for the static S-Box used in standard AES. Instead of using 0x63 as a constant value, this method uses a value obtained from the the first byte of the key. In result, the affine transformation formula becomes as follows in Equation 3, where x_0 to x_7 is the binary value of the obtained value. For example, the given key in hexadecimal is “ 70 61 73 73 77 6F 72 64”. On this case, the value used for XOR operation in affine transformation is 0x70 (binary : 01110000).

$$\begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} XOR \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} \quad (3)$$

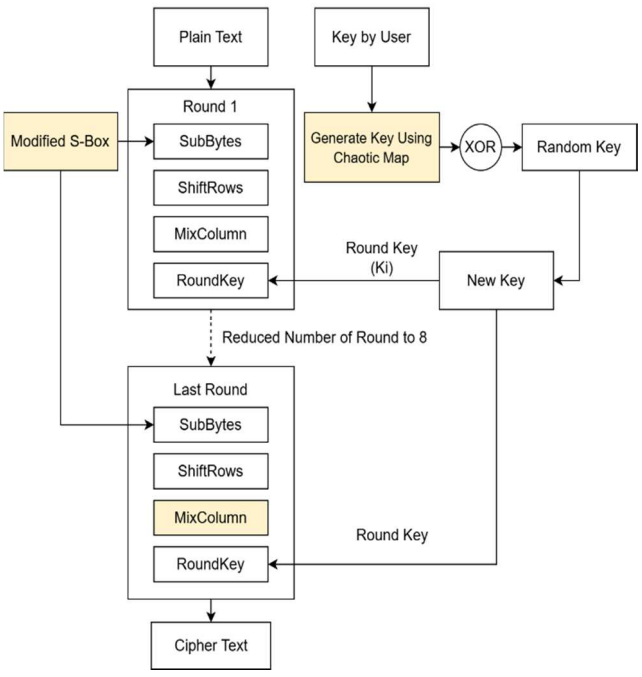

Fig. 2. Modified AES

B. New Key Generation using Chaotic Map

In this proposed method, chaotic map (Chebyshev Map) is used for new key generation. The key inputted by user is processed using an algorithm as shown in Table I. This algorithm converts the key from ASCII text value into hexadecimal value.

C. MixColumn

The last round of standard AES encryption doesn't include MixColumn step. In this proposed algorithm, we implemented the MixColumn step in the last round (8^{th} round) just as shown in Fig.3. This approach is expected to enhance the security level of encryption.

TABLE I. New Key Generation

Algorithm 1 : Key Generation using Chebyshev Map

Input : input_key
Output :
1: function generate_new_key(input_key):
2: mod_key = []
3: **for** each character in input_key:
4: mod_key.append(ASCII value of character % 9)
5: **for** each value in mod_key:
6: chaotic_sequence = []
7: repeat 15 times:
8: y = chaotic_value
9: chaotic_sequence.append(absolute value of y)
10: append chaotic_sequence to chebyshev_map_key
11: chaotic_bytes = []
12: **for** each value in the first 16 elements of chebyshev_map_key:
13: chaotic_bytes.append((value * 100) % 256)
14: random_key = Generate 16 random values between 1 and 6
15: new_key = []
16: **for** each chaotic_byte and corresponding random_key byte:
17: new_key.append(chaotic_byte XOR random_key byte)
18: final_key = Convert new_key to bytes
19: **return** final_key

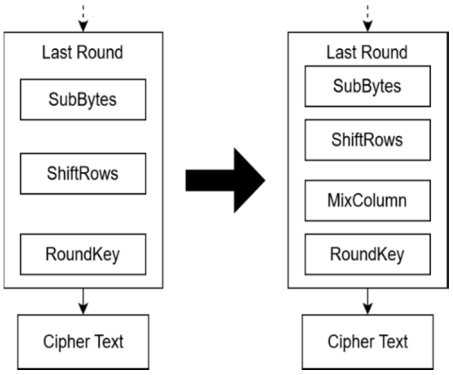

Fig. 3. Adding MixColumn to Last Round

TABLE II. Test Data

Input Key	1234567891234567
File Type	Text (.txt)
File Sizes	2 KB, 3KB, 4KB, 5KB, 10 KB

TABLE III. New Key Generation

Input Key	1234567891234567
New Key	59172e5912285c152e5e132f5e13295e

IV. Result and Discussion

For the evaluation process, we tested both proposed AES algorithm and standard AES algorithm on five text files of varying sizes, as detailed in the specifications provided in Table II.

A. New Key Generation

The input key is processed and converted into hexadecimal value using the algorithm in Table I. The result of the new key generation by the algorithm is as shown in Table III. This new key is then used for the encryption process.

B. Modified S-Box Construction

Using the new affine transformation formula defined in Equation 2, a new S-Box is constructed. Since the value of the first nibble from the new key obtained in the previous step is 0x59 (01011001 in binary), the affine transformation formula can be defined in Equation 4.

$$\begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} XOR \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \quad (4)$$

Construction of S-Box using the affine transformation in Equation 4 resulted in a completely new S-Box as shown in Table IV. This new S-Box is then used in the SubBytes process in each round of encryption replacing the Rijndael S-Box.

C. Distribution of Ciphertext

To analyze the distribution value in the ciphertext, we use a histogram diagram. A uniform distribution of values indicates that an encryption algorithm is effective. The distribution of byte value from the encryption result using proposed algorithm can be seen in Fig. 4. The byte value is in ASCII ranged 0-255. Fig. 4 shows that the ciphertext distribution is relatively balanced.

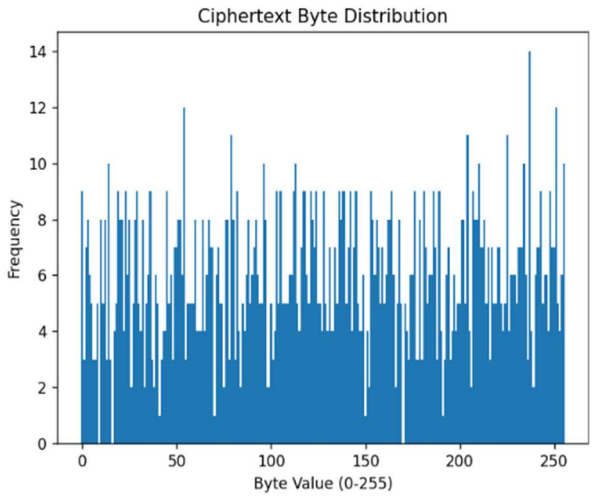

Fig. 4. Ciphertext Value Distribution

TABLE IV. New S-Box

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

D. Performance Evaluation

The performance of proposed modified AES algorithm was evaluated, including encryption time and avalanche score compared to the standard AES algorithm. For the encryption time, the evaluation process was done by testing the algorithm in encrypting five text files of different sizes (2 KB, 3 KB, 4 KB, 5 KB, and 10 KB). The result of the test can be observed in the Table V and Fig 5.

TABLE V. Encryption Time Comparison

File Size	Encryption Time (s)	
	Proposed AES	Standard AES
2 KB	20.37	24.29
3 KB	31.32	40.24
4 KB	54.05	66.03
5 KB	64.33	77.49
10 KB	134.03	166.73

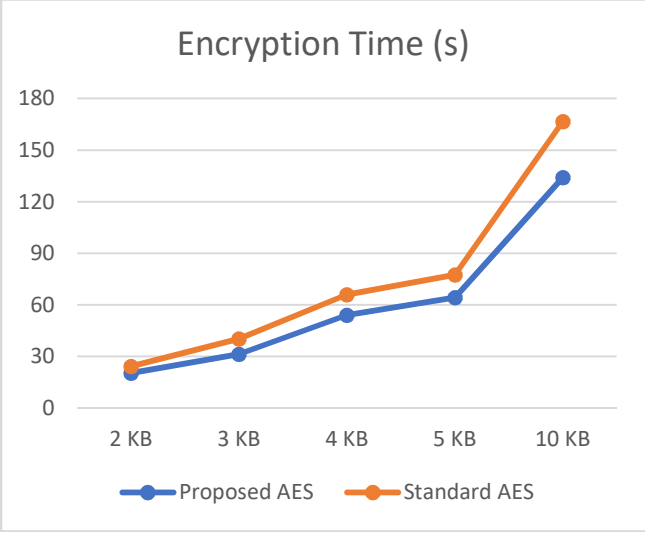

Fig. 5. Encryption Time Comparison

Based on the result in Table V and Fig 5, it can be observed that the proposed algorithm demonstrated faster encryption time, approximately 20% faster than standard AES, across various file sizes. The encryption time difference between the two increases gradually as the file sizes got larger.

To evaluate the security of the proposed algorithm, avalanche effect test was done. Avalanche effect refers to a property where a minor change, even only one bit, in the plaintext will result in significant change in ciphertext [18]. A good avalanche score for an encryption algorithm is ≥ 50%. Avalanche score can be calculated using Equation 5.

$$avalanche\ score = \frac{number\ of\ flipped\ bits}{total\ bits} \times 100\% \quad (5)$$

Avalanche effect test was conducted on three different plaintexts in hexadecimal value listed on Table VI. The avalanche score is obtained from the result of flipping the most significant bit in the plaintext. The result of avalanche effect test of proposed algorithm compared to standard AES algorithm is as shown in Table VII. The proposed algorithm produced a better average avalanche score than the standard AES, even achieving highest score of 58.59 % on Test 2.

TABLE VI. Avalanche Test Data

	Test Data in Hexadecimal
Test 1	70 61 73 73 77 6F 72 64 31 32 33 34 35 36 37 38
Test 2	56 71 76 50 39 4A 46 4B 55 5A 69 66 48 4A 42 6A
Test 3	67 6B 23 63 37 66 63 40 79 5E 76 5E 31 73 57 64

TABLE VII. Avalanche Score

Methods	Avalanche Score (%)			
	Test 1	Test 2	Test 3	Average
Proposed AES	53.12	58.59	47.66	53.02
AES Standard	52.34	50.00	47.66	50.00

V. Conclusions

The Advanced Encryption Standard (AES), although widely implemented, exhibits certain vulnerabilities that could affect its effectiveness against cryptographic attacks. Specifically, the use of a fixed S-Box and the absence of the MixColumn step in the final encryption round introduces potential weaknesses. In addition, as data sizes increase, the performance of the standard AES algorithm can be a limiting factor for applications requiring both high security and efficient processing.

To address these issues, this paper proposes an enhanced AES algorithm for 128-bit key by incorporating chaotic map-based key generation, dynamic S-Box, reduction in the number of rounds, and additional MixColumn step in the final round to address inherent limitations in standard AES that has been mentioned. The proposed algorithm successfully address those issues in AES algorithm, significantly enhancing security and complexity against potential cryptographic attacks.

Performance evaluation and experimental tests indicate that the proposed algorithm achieved a higher avalanche score and improved encryption time compared to standard AES, particularly when handling a larger file. These results suggest that the modified AES has potential for broader adoption in fields requiring robust data security.

Future work could focus on exploring the scalability of the proposed algorithm for different key sizes of AES algorithm, larger datasets, and further optimizing dynamic S-Box generation, as well as rigorous testing across various hardware platforms to assess performance consistency.

Acknowledgement

This research was funded by Institut Teknologi Sepuluh Nopember (ITS) under Penelitian Kolaborasi Pusat and under the Project Scheme of the Publication Writing and Intellectual Property Rights (IPR) Incentive (Penulisan Publikasi dan Hak Kekayaan Intelektual/PPHKI) Program 2024

References

[1] I. Riadi, T. Ahmad, R. Sarno, P. Purwono, and A. Ma'arif, "Developing Data Integrity in an Electronic Health Record System using Blockchain and InterPlanetary File System (Case Study: COVID-19 Data)," *Emerg. Sci. J.*, vol. 4, pp. 190–206, Feb. 2022, doi: 10.28991/esj-2021-SP1-013.

[2] F. Afandi and R. Sarno, "Android Application for Advanced Security System based on Voice Recognition, Biometric Authentication, and Internet of Things," in *2020 International Conference on Smart Technology and Applications (ICoSTA)*, Surabaya, Indonesia: IEEE, Feb. 2020, pp. 1–6. doi: 10.1109/ICoSTA48221.2020.1570615292.

[3] K. M. Rajasekharaiah, C. S. Dule, and E. Sudarshan, "Cyber Security Challenges and its Emerging Trends on Latest Technologies," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 981, no. 2, p. 022062, Dec. 2020, doi: 10.1088/1757-899X/981/2/022062.

[4] D. Kumar Sharma, N. Chidananda Singh, D. A. Noola, A. Nirmal Doss, and J. Sivakumar, "A review on various cryptographic techniques & algorithms," *Mater. Today Proc.*, vol. 51, pp. 104–109, 2022, doi: 10.1016/j.matpr.2021.04.583.

[5] S. Chandra, S. Paira, and S. S. Alam, "A comparative survey of symmetric and asymmetric key cryptography," 2014.

[6] A. Hamza and B. Kumar, "A Review Paper on DES, AES, RSA Encryption Standards," in *2020 9th International Conference System Modeling and Advancement in Research Trends (SMART)*, Moradabad, India: IEEE, Dec. 2020, pp. 333–338. doi: 10.1109/SMART50582.2020.9336800.

[7] A. Wahab, R. B. Bahaweres, M. Alaydrus, M. Muhaemin, and R. Sarno, "Performance analysis of VoIP client with integrated encryption module," in *2013 1st International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*, Sharjah: IEEE, Feb. 2013, pp. 1–6. doi: 10.1109/ICCSPA.2013.6487300.

[8] M. Ahirrao, B. Pathak, and M. Dixit, "AES and its Multiple Modifications for Various Applications," in *2023 6th International Conference on Advances in Science and Technology (ICAST)*, Mumbai, India: IEEE, Dec. 2023, pp. 476–479. doi: 10.1109/ICAST59062.2023.10454951.

[9] S. Jayanthy, S. Nageswarvijay, R. K. R. Kumar, and R. K. Kanth, "Smart Key using AES Algorithm," in *2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)*, Coimbatore, India: IEEE, Sep. 2021, pp. 467–473. doi: 10.1109/ICIRCA51532.2021.9545058.

[10] O. A. Dawood and O. I. Hammadi, "An Analytical Study for Some Drawbacks and Weakness Points of the AES Cipher (Rijndael Algorithm)," in *Proceeding of 1st International Conference on Information Technology*, Lebanese French University - LFU, Apr. 2017, pp. 129–133. doi: 10.25212/ICoIT17.013.

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

[12] M. S. Fadhil, A. K. Farhan, M. N. Fadhil, and N. M. G. Al-Saidi, "A New Lightweight AES Using a Combination of Chaotic Systems," in *2020 1st. Information Technology To Enhance e-learning and Other Application (IT-ELA*, Baghdad, Iraq: IEEE, Jul. 2020, pp. 82–88. doi: 10.1109/IT-ELA50150.2020.9253099.

[13] A. E. Adeniyi, K. M. Abiodun, J. B. Awotunde, M. Olagunju, O. S. Ojo, and N. P. Edet, "Implementation of a block cipher algorithm for medical information security on cloud environment: using modified advanced encryption standard approach," *Multimed. Tools Appl.*, vol. 82, no. 13, pp. 20537–20551, May 2023, doi: 10.1007/s11042-023-14338-9.

[14] National Institute of Standards and Technology (US), "Advanced Encryption Standard (AES)," National Institute of Standards and Technology (U.S.), Washington, D.C., NIST FIPS 197-upd1, May 2023. doi: 10.6028/NIST.FIPS.197-upd1.

[15] R. B. Naik and U. Singh, "A Review on Applications of Chaotic Maps in Pseudo-Random Number Generators and Encryption," *Ann. Data Sci.*, vol. 11, no. 1, pp. 25–50, Feb. 2024, doi: 10.1007/s40745-021-00364-7.

[16] J. Ryu, D. Kang, and D. Won, "Improved Secure and Efficient Chebyshev Chaotic Map-Based User Authentication Scheme," *IEEE Access*, vol. 10, pp. 15891–15910, 2022, doi: 10.1109/ACCESS.2022.3149315.

[17] N. A. Ariffin Mohd and A. Y. Ahmed Ashawesh, "Enhanced AES algorithm based on 14 rounds in securing data and minimizing processing time," *J. Phys. Conf. Ser.*, vol. 1793, no. 1, p. 012066, Feb. 2021, doi: 10.1088/1742-6596/1793/1/012066.

[18] R. Verma and A. K. Sharma, "Cryptography: Avalanche effect of AES and RSA," *Int. J. Sci. Res. Publ. IJSRP*, vol. 10, no. 4, p. p10013, Apr. 2020, doi: 10.29322/IJSRP.10.04.2020.p10013.