Avoiding Lookup Table in AES Algorithm

Ragiel Hadi Prayitno
Doctoral Program in Information Technology
Gunadarma University
Jakarta, Indonesia
ragielhp@staff.gunadarma.ac.id

Sunny Arief Sudiro
Electrical Engineering
STMIK Jakarta STI&K
Jakarta, Indonesia
sunnyariefsudiro@ieee.org

Sarifuddin Madenda
Doctoral Program in Information Technology
Gunadarma University
Jakarta.Indonesia
sarif@staff.gunadarma.ac.id

2021 Sixth International Conference on Informatics and Computing (ICIC) | 978-1-6654-2155-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICIC54025.2021.9632897

***Abstract*— This article describes the AES encryption and decryption process without using lookup tables in the MixColumns transformation. The encryption process consists of transforming subbytes, shiftrows, mixcolumns and addroundkey. The process was carried out for 10 rounds, but in round 10 the mixcolumns transformation was not carried out. The decryption process consists of inverse mixcolumns, inverse shiftrows, inverse subbytes and addroundkey. In this study, the AES encryption and decryption process was carried out using two methods, namely based on the lookup table and without using the lookup table on the MC/IMC transformation. The method in this article is applied to Matlab software. The experimental results show that the encryption and decryption process using a lookup table is slower than the method without a lookup table. The encryption process without a lookup table on the MC transformation takes 0.091 seconds while using a lookup table takes 0.399 seconds. The decryption process without a lookup table on the IMC transformation takes 0.149 seconds while using a lookup table takes 0.206 seconds.**

Keywords—AES, Decryption, Encryption, Matlab, Without lookup table,

I. INTRODUCTION

The development of information and communication technology (ICT) dominates all sectors, from government departments, business, industry, to public activities. This changes made it possible to generate and transfer data on a large scale over the internet. Disseminated data can be easily controlled and duplicated, making it vulnerable to multiple data security attacks. The important thing that needs to be considered to maintain the confidentiality of information is data security. Data becomes more vulnerable to being compromised because the data has been converted into digital form [1]. The data contains sensitive information and then sends it over the internet so that it cannot be read by anyone except the intended recipient [2].

Cryptography is a security method that has been around for centuries. Cryptographic methods are used in communications from military to commercial. Advances in the Internet and e-commerce have made cryptographic methods an important part of the global economy and something that millions of people use every day. Confidential information such as bank documents, credit card statements, passwords or private messages must be encrypted and modified so that they can only be accessed by authorized persons or parties and not decrypted by others [3]. Based on the problems that have been described, various cryptographic algorithms have been developed such as the Data Encryption Standard (DES) algorithm and the Advanced Encryption Standard (AES) algorithm.

Cryptography's general method is the process of converting a text message (or plain text) into an encrypted text message (or cipher) based on an algorithm known to the sender and recipient so that the encrypted message can be returned to its original form, namely plain text. The encrypted message cannot be read by anyone except the recipient, where the recipient has the same key as the sender.

Encryption is the process of converting a plain text message into a ciphertext form. Decryption is the process of converting ciphertext into plain text messages [4]. Each encryption and decryption process has a key, where the key can be a word or phrase. The key is part of the cryptographic method used to enter data. The main difference between AES encryption technology and DES encryption technology is that AES directly uses substitution for scripts (using S-box). AES was adopted as Federal Information Processing Standards (FIPS) by the National Institute of Standards and Technology (NIST) in 2001 [5]. The AES algorithm has several processes, namely: Addroundkey, bitwise xor of the round key is performed. Subbyte, the array data matrix substitution is done using the S-Box table. Shiftrows, change the array data matrix. Mixcolumns, randomization of the array data matrix is performed.

In this paper, the researcher focuses on designing the AES algorithm using Galois for algorithm optimization. This algorithm will be applied to FPGA devices so that it requires a simple algorithm that is easy to implement on hardware with minimal resource requirements. The experiment was carried out on Asus hardware Intel Core i7 generation 8, 16GB ram using Matlab software.

II. MATERIALS AND METHODS

In cryptography, a block cipher uses symmetric key cipher. A block cipher is a data encryption method (to generate ciphertext) in which a cryptographic key and an algorithm are applied to blocks of data simultaneously as a group, rather than one bit at a time. block ciphers require a 128-bit block of plaintext as input and output the appropriate 128-bit ciphertext block when encrypting. The transformation is controlled by the second input - the secret key. The decryption algorithm uses a 128-bit block of ciphertext and a secret key that produces a 128-bit block of original text. Block ciphers can be adapted to stream ciphers, where the stream cipher operates on one digit at a time, and the transformation changes during encryption [2].

A. Data Encryption Standard (DES)

The Data Encryption Standard (DES) is a block cipher with a 64-block size and a 56-bit key. DES consists of 16 substitution and permutation rounds. In each round, data and key bits are shifted, converted, XORed, and sent through 8 s-boxes. This is a set of lookup tables that are important to the DES algorithm. The decryption process is basically the same, and the reverse is complete [2].

B. Advanced Encryption Standard (AES)

Advanced Encryption Standard (AES), also called Rijndael encryption is a block encryption standard that was recently adopted by the United States Federal Government [6, 7]. AES is projected as a replacement for DES. After several rounds of screening, AES was widely used [8]. On November 26, 2011 NIST released its latest encryption standard filtering after five years, and it came into effect in May 2002. After four years of deposition and testing, AES has become one of the most popular symmetric encryption algorithms [6].

Symmetrical AES encryption system in the group has 3 types of key length 128 bits, 192 bits and 256 bits with a packet size of 128 bits, the algorithm has good flexibility. So AES encryption system is widely used in software and hardware. In the three key lengths of the AES algorithm, the key length of 128 bits is often used. When under the key length, 10 times the iterative calculation in the internal algorithm. In addition to the final round, each round consists of five parts: SubBytes, S-box, ShiftRows, MixColumns, AddRoundKey [6].

Exclusivity between plain text and key extension blocks [9]. In AES, five units of date measurement can be used: bits, bytes, characters, groups, and states. Each round of AES includes byte substitution (SubBytes), row shift transform (ShiftRows), mixed column transform (MixColumns), key transform (AddRoundKey), etc. From one stage of data packet conversion to another, throughout the initial and final stages of encryption, the AES concept uses data clustering.

1) SubBytes

The SubBytes transform is a non-linear and reversible byte transformation. This makes AES quite strong against attacks. SubBytes transformation performs substitution on each byte of the state matrix with the reference value stored in the S-Box for encryption process. The decryption process, substitution is done using the inverse Sbox table for each state. The S-box is formed from a lookup table measuring 256 bytes [10].

2) ShiftRows

The ShiftRows transform moves rows 1, 2 and 3 of the State matrix cyclically to the left by 1, 2 and 3 positions, respectively. The offset value depends on the line number. So the first line remains unchanged. The cyclic rotation of the rows implements the diffusion property in the AES algorithm. The ShiftRows transformation is shown in figure 1.

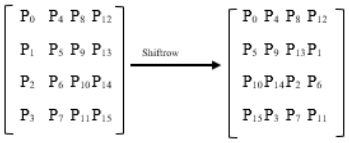

Fig. 1. Shiftrows process

3) MixColumns

MixColumns (MC) transformation, the column in the state is treated as a polynomial with the coefficient in GF(2^8).

$$[a3, a2, a1, a0] \tag{1}$$

The data in formula (1) is treated as a polynomial, so that it becomes as follows:

$$a3X^3 + a2X^2 + a1X^1 + a0 \tag{2}$$

The addition of a polynomial is performed because it is added to polynomial ring. In order that, the result remains as big as a word, the multiplication is performed modulo polynomial.

$$g(x) = X^4 + 1 \tag{3}$$

$X^4 + 1$ is the equation of decimal value 17, where X^4 represents the value 2^4 and 1 represents 2^0. It is easy to indicate that.

$$X^i \, mod \, (X^4 + 1) = X^{i \, mod \, 4} \tag{4}$$

because -1 = 1 in GF(2^8). Since $X^4 + 1$ is not irreducible polynomial in K[X] where K = GF(2^8), then K[X]/g(X)K[X] is not a field: not all polynomial has inverse. However, the polynomial.

$$a(x) = 03X^3 + 01X^2 + 01X + 02 \tag{5}$$

has Inverse MixColumns (IMC)

$$a^{-1}(x) = 0bX^3 + 0dX^2 + 09X + 0e \tag{6}$$

The values of formulas (5) and (6) are hexadecimal numbers. The MixColumns transform multiplies (modulo polynomial g(X)) each column in the state (treated as a polynomial) by the polynomial a(X). The MixColumns transformation of the state can be formulated for its effect in each column c as follows [5]:

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \tag{7}$$

S0,c, s1,c,s2,c, and s3,c are the original data. S'0, c is data change in the first row of column c which resulted from MC/ IMC transformation process between MC matrix and original data. Afterward for s'1,c, s'2,c, and s'3,c are data change in row 2 to row 4 column c.

MixColumn and InvMixColumn operations are used in Advanced Encryption Standard for the purpose of hardware implementation in restricted environments. This study was supported by mathematical analysis of both transformations and caused serial decomposition and efficient parallel [11].

The previous research has created a high-efficient architecture to perform mixed column operations, which is the main function in Advanced Encryption Standard (AES) method. This study used prehistoric Vedic Mathematics techniques which can provide more efficient results in AES [12].

4) AddRoundKey

The AddRoundKey transform is the last transformation in each round. In this transformation round keys obtained in-XOR with the state of the bitwise operation [10].

C. Mean Square Error (MSE)

Mean squared error (MSE) is the most widely used and simplest reference metric that is calculated by the difference in the intensity of the squares of the distorted and reference image pixels. MSE is the most common image quality measurement metric estimator. MSE can be used if there are outliers that need to be detected. In fact, MSE is great for attributing larger weights to such points, thanks to the L_2 norm: clearly, if the model eventually outputs a single very bad prediction, the squaring part of the function magnifies the error [11].

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(X_i - Y_i)^2 \quad (8)$$

(best value = 0; worst value = +∞)

III. PROPOSED METHOD

The developed method refers to the AES algorithm without using a reference table in the MixColumns transformation. This research was conducted using Matlab software. Figure 2 illustrates the stages of this research process.

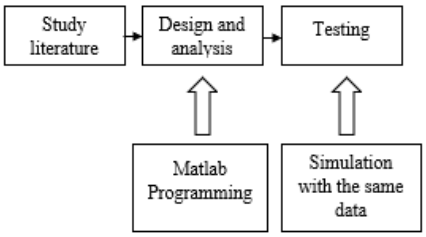

Fig. 2. Research methods

The research begins by conducting a more in-depth literature study on encryption and decryption AES algorithms, based on these results the researcher proposes a method without using a reference table on the MixColumns transformation. Fig. 3 shows the encryption process and Fig. 4 shows the decryption process in the AES algorithm.

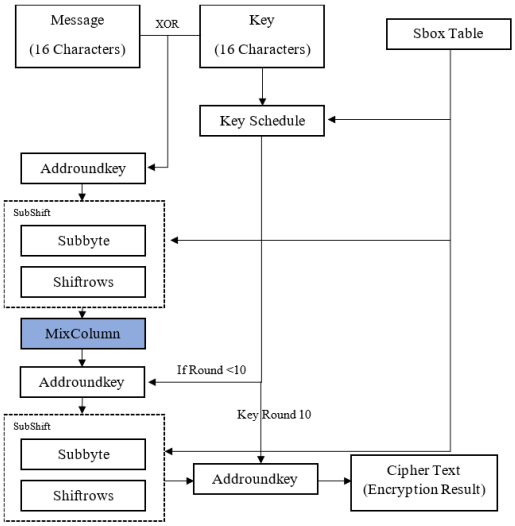

Fig. 3. Encryption process

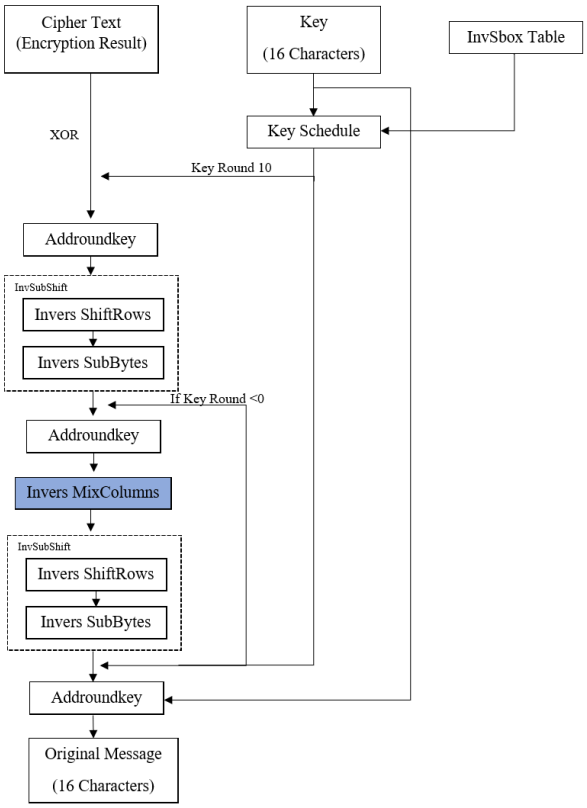

Fig. 4. Decryption process

A. Data

The data which is applied in this study is digital data in the form of 16 decimal numbers that is divided into 4x4 matrix, for example:

$$\begin{bmatrix} 170 & 249 & 69 & 159 \\ 249 & 80 & 183 & 251 \\ 143 & 80 & 80 & 239 \\ 60 & 239 & 239 & 64 \end{bmatrix}$$

B. Key

The key used in this research is digital data in the form of 16 decimal numbers which are divided into a 4x4 matrix, for example:

$$\begin{bmatrix} 236 & 167 & 246 & 156 \\ 245 & 70 & 46 & 24 \\ 19 & 34 & 121 & 96 \\ 234 & 213 & 236 & 47 \end{bmatrix}$$

C. Subbytes

Subbyte transformation is done by substitution with the S-Box table. The resulting data is a 4x4 matrix. Fig. 5 is a substitution process for the S-Box table.

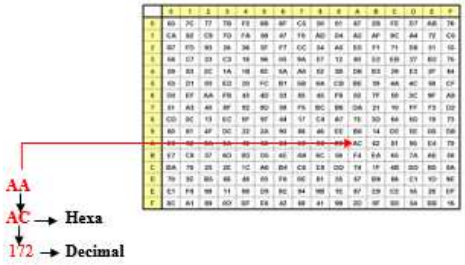

Fig. 5. Subbyte process

D. ShiftRows

The shiftrows transformation shifts the bytes in the state line with the following conditions:

1. Rows 0 is not shifted,
2. Line 1 is shifted 1 position to the left with the left byte being the right byte,
3. line 2 is shifted 2 positions to the left, so the two bytes on the left are exchanged for the two bytes on the right, and
4. Row 3 is shifted 3 positions to the left, which has the same effect as shifting 1 position to the right.

$$\begin{bmatrix} 170 & 249 & 69 & 159 \\ 249 & 80 & 183 & 251 \\ 143 & 80 & 80 & 239 \\ 60 & 239 & 239 & 64 \end{bmatrix} \xrightarrow{\text{ShiftRows}} \begin{bmatrix} 170 & 249 & 69 & 159 \\ 80 & 183 & 251 & 249 \\ 80 & 239 & 143 & 80 \\ 64 & 60 & 239 & 239 \end{bmatrix}$$

Fig. 6. ShiftRows process

$$Mix = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \quad \text{Inv}Mix = \begin{bmatrix} 14 & 11 & 13 & 9 \\ 9 & 14 & 11 & 13 \\ 13 & 9 & 14 & 11 \\ 11 & 13 & 9 & 14 \end{bmatrix}$$

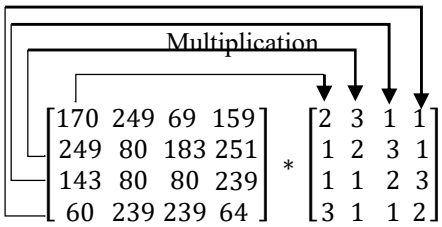

Fig. 7. MixColumns process

E. Matrix MC / IMC

MC/IMC is treated as a polynomial with coefficients in $GF2^8$, based on (5) data matrix MC and data matrix IMC, based on (6).

F. AddRoundKey

The AddRoundKey operation performs an exclusive or (XOR) between round key and the state. The stages of the AddRoundKey process are as follows:

1. The message (Plain text) to be encrypted is converted into binary numbers.
2. Key (Cipher key) used in the encryption process is converted into binary numbers.
3. XOR message with key to generate addroundkey.

IV. Result And Discussion

The first research experiment was conducted using Matlab software. In this experiment, two different methods were used:

1. Based on the table on MixColumns transformation.
2. Without lookup table on MixColumns transformation.

A. Experiment based on the table on MixColumns transformation

$$Data = \begin{bmatrix} 170 & 249 & 69 & 159 \\ 249 & 80 & 183 & 251 \\ 143 & 80 & 80 & 239 \\ 60 & 239 & 239 & 64 \end{bmatrix}$$

$$\text{Key} = \begin{bmatrix} 236 & 167 & 246 & 156 \\ 245 & 70 & 46 & 24 \\ 19 & 34 & 121 & 96 \\ 234 & 213 & 236 & 47 \end{bmatrix}$$

XOR data matrix with key matrix to generate AddRoundKey. Each byte of AddRoundKey results is then substituted against the S-Box table. After that, the shift process is carried out on rows 2, 3 and 4 of the substitution matrix.

TABLE I. L Table

0	100	125	101	150	102	126	43	175	44	127	204	151	83	68	103
0	4	194	47	143	221	110	121	88	215	12	187	178	57	17	74
25	224	29	138	219	253	72	10	168	117	246	62	135	132	146	237
1	14	181	5	189	48	195	21	80	122	111	90	144	60	217	222
50	52	249	33	54	191	163	155	244	235	23	251	97	65	35	197
2	141	185	15	208	6	182	159	234	22	196	96	190	162	32	49
26	129	39	225	206	139	30	94	214	11	73	177	220	109	46	254
198	239	106	36	148	98	66	202	116	245	236	134	252	71	137	24
75	76	77	18	19	179	58	78	79	89	216	59	188	20	180	13
199	113	228	240	92	37	107	212	174	203	67	82	149	42	124	99
27	8	166	130	210	226	40	172	233	95	31	161	207	158	184	140
104	200	114	69	241	152	84	229	213	176	45	108	205	93	38	128
51	248	154	53	64	34	250	243	231	156	164	170	55	86	119	192
238	105	201	147	70	136	133	115	230	169	118	85	63	242	153	247
223	28	9	218	131	145	61	167	173	81	123	41	91	211	227	112
3	193	120	142	56	16	186	87	232	160	183	157	209	171	165	7

TABLE II. E TABLE

1	95	229	83	76	131	181	254	251	195	159	155	252	69	18	57
3	225	52	245	212	158	196	25	22	94	186	182	31	207	54	75
5	56	92	4	103	185	87	43	58	226	213	193	33	74	90	221
15	72	228	12	169	208	249	125	78	61	100	88	99	222	238	124
17	216	55	20	224	107	16	135	210	71	172	232	165	121	41	132
51	115	89	60	59	189	48	146	109	201	239	35	244	139	123	151
85	149	235	68	77	220	80	173	183	64	42	101	7	134	141	162
255	164	38	204	215	127	240	236	194	192	126	175	9	145	140	253
26	247	106	79	98	129	11	47	93	91	130	234	27	168	143	28
46	2	190	209	166	152	29	113	231	237	157	37	45	227	138	36
114	6	217	104	241	179	39	147	50	44	188	111	119	62	133	108
150	10	112	184	8	206	105	174	86	116	223	177	153	66	148	180
161	30	144	211	24	73	187	233	250	156	122	200	176	198	167	199
248	34	171	110	40	219	214	32	21	191	142	67	203	81	242	82
19	102	230	178	120	118	97	96	63	218	137	197	70	243	13	246
53	170	49	205	136	154	163	160	65	117	128	84	202	14	23	1

The MixColumns transformation is done by multiplying the resulting ShiftRows matrix with the MixColumns matrix. Table 1 is a lookup table called table L and table 2 is lookup table called table E for MC and IMC. The resulting ShiftRows matrix and the MC matrix are multiplied based on table L and table E to change the previous matrix data. The changed matrix data will be multiplied by the IMC matrix based on table L and table E to restore the matrix data as before. The multiplication results based on table L and table E are the search results with table L then continued by adding up the results, followed by searching with table E. The search results with tables L and E are only carried out on the MC index matrix with values 2 and 3.

XOR the results of MixColumns with the first round key to generate AddRoundKey, then repeat these steps until round 9. In round 10 do the subbyte, shiftrows and addroundkey processes. Based on experiments conducted with Matlab software, Figure 8(A) shows the time required to perform the encryption process with the AES algorithm is 0.399 seconds and Figure 8(B) shows the decryption process takes 0.206 seconds.

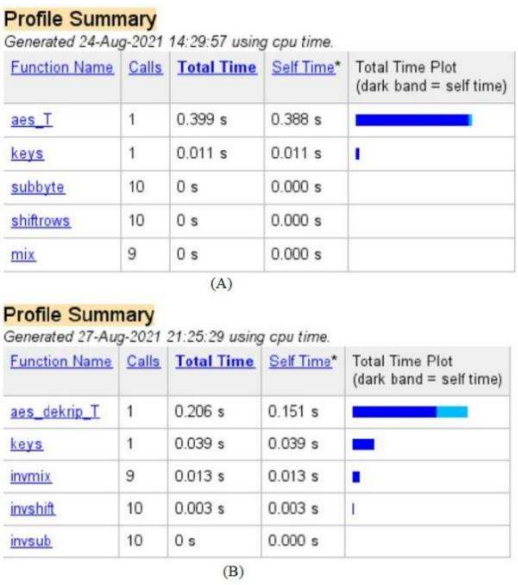

Fig. 8. Time process encryption (A) and decryption (B) based on the lookup table

B. *Experiment Without lookup table on MixColumns transformation*

The data and keys used in the second experiment were the same as the first experiment. The steps taken are also the same as in the first experiment, only the difference is in the MixColumns transformation. In the second experiment, the MixColumns transformation was performed by multiplying the resulting ShiftRows matrix with the MC matrix using the multiplication of the galois field (GF 2^8).

In the multiplication of the gallois field, if the multiplication of the gallois field exceeds 255, it is necessary to do an irreducible polynomial on the product. irreducibility needs to be applied to x8, x9, etc.

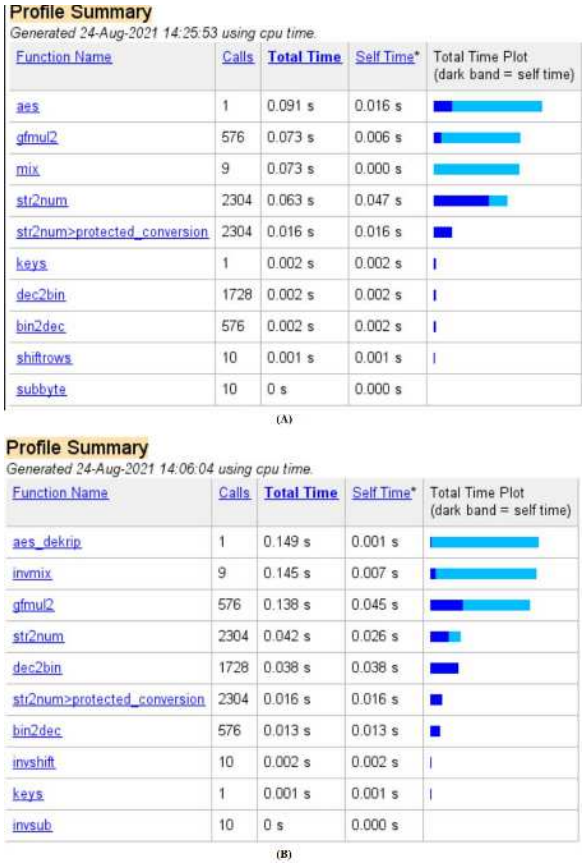

Fig. 9. Time process encryption (A) and decryption (B) without lookup table

Based on experiments conducted with Matlab software, Figure 9(A) shows the time required to perform the encryption

process with the AES algorithm is 0.091 seconds and Figure 9(B) shows the decryption process takes 0.149 seconds. This method did not reduce the data security aspect, because the results of the two methods were the same.

TABLE III. EXPERIMENT RESULTS COMPARISON

No	Messages	Key	Exp Method 1	Exp Method 2
1	$\begin{bmatrix}170 & 249 & 69 & 159\\ 249 & 80 & 183 & 251\\ 143 & 80 & 80 & 239\\ 60 & 239 & 239 & 64\end{bmatrix}$	$\begin{bmatrix}236 & 167 & 246 & 156\\ 245 & 70 & 46 & 24\\ 19 & 34 & 121 & 96\\ 234 & 213 & 236 & 47\end{bmatrix}$	$\begin{bmatrix}87 & 180 & 191 & 99\\ 55 & 96 & 146 & 163\\ 20 & 170 & 232 & 6\\ 160 & 215 & 133 & 141\end{bmatrix}$	$\begin{bmatrix}87 & 180 & 191 & 99\\ 55 & 96 & 146 & 163\\ 20 & 170 & 232 & 6\\ 160 & 215 & 133 & 141\end{bmatrix}$
2	$\begin{bmatrix}249 & 80 & 183 & 251\\ 249 & 80 & 183 & 251\\ 143 & 80 & 80 & 239\\ 60 & 239 & 239 & 64\end{bmatrix}$	$\begin{bmatrix}236 & 167 & 246 & 156\\ 245 & 70 & 46 & 24\\ 19 & 34 & 121 & 96\\ 234 & 213 & 236 & 47\end{bmatrix}$	$\begin{bmatrix}45 & 247 & 194 & 236\\ 251 & 81 & 15 & 165\\ 100 & 238 & 213 & 29\\ 156 & 6 & 169 & 253\end{bmatrix}$	$\begin{bmatrix}45 & 247 & 194 & 236\\ 251 & 81 & 15 & 165\\ 100 & 238 & 213 & 29\\ 156 & 6 & 169 & 253\end{bmatrix}$

Based on the results obtained in Table 3, the MSE between the experimental methods 1, and 2 that have been carried out for 2 experiments, the result is 0.

V. CONCLUSION

In this research, the AES algorithm was implemented using 2 methods, namely based on a reference table and without using a reference table in the MixColumns/Inverse MixColumns transformation. The experiment was carried out using the same 2 inputs, namely data and key. Based on the test results, the time required to perform the encryption process without using a reference table in the MixColumns transformation is 0.091 seconds, while the decryption process takes 0.149 seconds. While the time needed to perform the encryption process using the reference table on the MixColumns transformation is 0.399 seconds, while the decryption process takes 0.206 seconds. The experimental results show that the method using the reference table is slower because in the process there are two substitutions for the 256-byte lookup table.

ACKNOWLEDGMENT

This work is supported by Research Program Kemendikbudristek Indonesia in Hibah Penelitian Disertasi Contract Number : 064/SP2H/LT/DRPM/2021 dated 18 March 2021 and Yayasan Pendidikan Gunadarma Jakarta Indonesia, Contract Number : 05A.26/LP/UG/IV/2021, 5 April 2021.

REFERENCES

[1] Djamalilleil, A. Muslim, M. Salim, Y. Alwi and H. Herman, "Modified Transposition Cipher Algorithm for Images Encryption," The 2nd east indonesia conference on computer and information technology (Eiconcit Makassar, South Sulawesi; (pp. 1-4), Doi: 10.1109/eiconcit.2018.8878326, 2018.

[2] Shraddha Soni, Himani Agrawal and Dr. (Mrs.) Monisha Sharma, "Analysis and Comparison between AES and DES Cryptographic Algorithm," International Journal of Engineering and Innovative Technology (IJEIT), Volume 2, Issue 6, 2012.

[3] Mu. Annalakshmi & A. Padmapriya, "Zigzag Ciphers: A Novel Transposition Method," IJCA Proceedings on International Conference on Computing and information Technology 2013 IC2IT(2); pp. 8-12, Dec 2013.

[4] P. Poonia & P. Kantha, "Comparative Study of Various Substitution And Transposition Encryption Techniques," International Journal of Computer Applications (0975 - 8887; 145(10), 24-27. Doi:10.5120/ijca2016910783), 2016.

[5] Elaine Barker, "Guideline for Using Cryptographic Standards in the Federal Government:Cryptographic Mechanisms," NIST Special Publication 800-175B Revision 1, DOI: doi.org/10.6028/NIST.SP.800-175Br1, March 2020.

[6] Q. Zhang, Qunding, "Digital Image Encryption Based On Advanced Encryption Standard(AES) Algorithm," 2015 Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control: 1218-1221, 2015.

[7] Y. J. Li, W. L. Wu. "Improved Integral Attacks on Rijndael C. Journal of Information Science and Engineering," 2011, 27(6): 2031-2045.

[8] Y. W. Zhu, H. Q. Zhang, Y. B. Bao, "Study of the AES Realization Method on the Reconfigurable Hardware C," International Conference on Computer Sciences and Applications, 2013: 72-76, 2013.

[9] J. Tpldinas, V. Stuikys, R. Damasevicius, "Energy Efficiency Comparion with Cipher Strength of AES and Rijndael Cryptographic Algorithms in Moble Devices," J. Elektonika IR Elektrotechnika, 2: 11-14, 2011.

[10] H. Zodpe*, A. Sapkal, "An efficient AES implementation using FPGA with enhanced security features," Journal of King Saud University – Engineering Sciences, https://doi.org/10.1016/j.jksues.2018.07.002, 2018.

[11] Viktor Fischer, Milos Drutarovsky & Pawel Chodowiec, "InvMixcolumn Decomposition and Multilevel Resource Sharing in AES Implementation," in IEEE Trans. On VLSI systems; 13(8), pp.989-992, 2005.

[12] M. Senthil Kumar and DR. S. Rajalakshmi, "High Efficient Modified MixColumns in Advanced Encryption Standard using Vedic Multiplier," International Conference on Current Trends in Engineering and Technology, ICCTET, 2014.

[13] Berent, Adam. "Advanced Encryption Standard by Example," Document available at URL https://www.adamberent.com/wp-content/uploads/2019/02/AESbyExample.pdf Accessed: May 2021.