Square transposition: an approach to the transposition process in block cipher

Magdalena A. Ineke Pekereng, Alz Danny Wowor
Department of Informatic Engineering, Universitas Kristen Satya Wacana, Salatiga, Indonesia

Article Info

Article history:

Received Dec 26, 2020
Revised Mar 30, 2021
Accepted Oct 7, 2021

Keywords:

AES
DES
Input scheme
Retrieval scheme
Square transposotion

ABSTRACT

The transposition process is needed in cryptography to create a diffusion effect on data encryption standard (DES) and advanced encryption standard (AES) algorithms as standard information security algorithms by the National Institute of Standards and Technology. The problem with DES and AES algorithms is that their transposition index values form patterns and do not form random values. This condition will certainly make it easier for a cryptanalyst to look for a relationship between ciphertexts because some processes are predictable. This research designs a transposition algorithm called square transposition. Each process uses square 8×8 as a place to insert and retrieve 64-bits. The determination of the pairing of the input scheme and the retrieval scheme that have unequal flow is an important factor in producing a good transposition. The square transposition can generate random and non-pattern indices so that transposition can be done better than DES and AES.

Corresponding Author:

Magdalena A. Ineke Pekereng
Department of Informatics Engineering
Universitas Kristen Satya Wacana
Jl. Notohamidjojo 1-10, Salatiga 50718, Indonesia
Email: ineke.pakereng@uksw.edu

1. INTRODUCTION

Diffusional transposition process is useful for the spread of plaintext redundancy in a ciphertext. Modern cryptography such as data encryption standard (DES)and advanced encryption standard (AES) as information security standards used by National Institute of Standards and Technology (NIST) also contain transposition as one of the important processes in the algorithm [1], [2]. DES with initial permutation (IP) and inverse initial permutation $(\text{IP})^{-1}$ are truly essential in the transposition process [3]. Whereas, AES with shift rows capability is simpler in transposition [4]. The two algorithms use index values to determine the shift of each object. The excess diffusion in the algorithm is one of the factors that make DES and AES still attractive and feasible to use, which makes both of them are chosen by researchers as their information security methods [5]-[21].

The DES transposition index value in Figure 1 shows patterned results. 64-bit outputs in DES always form 8-bit groups. Each eight index values produce the same pattern, starting from the highest value that gradually decreases. For example a_i as the index value where $(i = 1, 2, 3, \cdots, 8)$ in the first group, the value of the same position in the next group always becomes $a_i + 1$ (mod 64).

The transposition on AES also has a patterned index value as shown in Figure 2. The AES index value forms a group of 4 characters. The first group (4-0) consists of four upward histograms and 0 different histograms. The second group: (3-1) consists of three upward histograms and 1 histogram that has different values. The same condition applies to the third group (2-2) and the fourth group (1-3).

The index value is expected as a new position to make the sequence of each character to be more irregular so that the diffusion factor will increasingly appear in the ciphertext. The moving of objects based on index values in DES and AES indicates a problem because it forms a certain pattern. The problem in DES and AES is that when the sequence or pattern of data $\{1, 2, 3, \cdots, n\}$ is known, the probability for finding the $\{n+1, n+2, \cdots\}$ data is great. This will weaken the algorithm and the patterned condition will certainly make it easier for cryptanalysts to find a plaintext-ciphertext relationship because part of the process is predictable.

A study related to the transposition process was also carried out by [22]-[25], who dismissed the shift row operation as a transposition operation in AES cryptography. Thus, XOR as an additional operation can be performed repeatedly up to three times. The research done by [26], [27] adds various processes to correct the shortcomings of transposition in the algorithm. Although the improvement of the transposition process by adding the algorithm in parallel will certainly obtain a good result, the adding of the algorithm takes more time, and space. In terms of efficiency, the algorithm is less elegant to be used as information security.

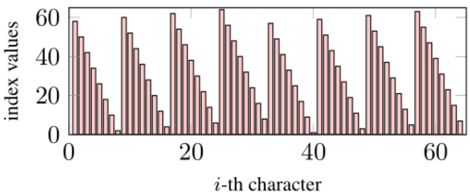

Figure 1. Index values of DES transposition

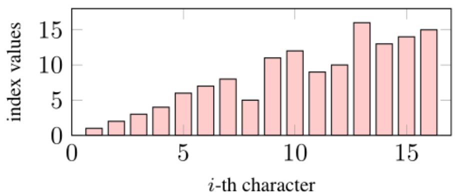

Figure 2. Index values of AES transposition

This research designs a transposition algorithm called Square Transposition. A square of $n \times n$ size is used as the medium to hold $(m = n \times n)$-bit. Each bit input is entered into a square using certain rules and taking of bit is also done with certain rules. Determination of whether the designed algorithm is good or not is seen based on its statistical testing. Statistical testing is done to determine the randomness of each index value. In addition, correlation testing is used to measure the algorithm's ability to disguise the relationship between input and output. Finally, DES and AES are compared to find out the power of Square Transposition in the algorithm for the transposition process.

2. PROPOSED RESEARCH

2.1. Square transposition

Square transposition consists of two processes namely bit-input into a square and bit-retrieval with a certain predetermined size. Suppose T = text input, t_i = i-th text character and a_i = i-th binary character, then:

$$T = \{t_1, t_2, \cdots, t_n\}; \qquad n|8, n \in \mathsf{Z}^+ \tag{1}$$

Where, $t_1 = \{a_{01}, a_{02}, a_{03}, \cdots, a_{08}\}$, $t_2 = \{a_{09}, a_{10}, a_{11}, \cdots, a_{16}\}$, $t_3 = \{a_{17}, a_{18}, a_{19}, \cdots, a_{24}\}$, $\cdots$, $t_n = \{a_{8n-7}, a_{8n-6}, a_{8n-5}, \cdots, a_{8n}\}$. If $n \not| 8$, then padding is done as many as k, so that it will result in (2). With $(n + k)|8$; $k = 1, 2, \cdots, 7$.

$$T = \{t_1, t_2, \cdots, t_n, t_{n+1}, t_{n+2}, \cdots, t_{n+k}\} \tag{2}$$

The square that is used as the transposition media can be adjusted to the bit size of the text input. This research chooses 64-bit text input, so it will be a square size of 8×8 shown in Figure 3.

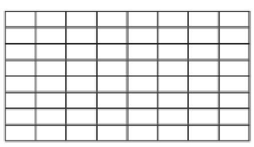

Figure 3. Square transposition 8×8

The entry scheme is a way to place each bit of a_i; $i \in \mathsf{Z}_{64}^{+}$ in the entry of square with certain rules. For example, every bit after entering into a square is the order of bits given in (3).

$$T_{sq} = \{a_1^*, a_2^*, a_3^*, \cdots, a_{64}^*\} \tag{3}$$

A retrieval scheme is a way to take every bit of $a_i^*, i \in \mathsf{Z}_{64}^{+}$ from the square with a certain rule. Notation for each bit taken from square $(a_{i\ (j)}^*)$; $\exists\, i, j \in \mathsf{Z}_{64}^{+}$ where i is the entry index and j is the retrieval index. In (4) is a schema collection dataset $L = \{l_1, l_2, l_3, \cdots, l_8\}$, where $\exists\, x \in \mathsf{Z}_{64}^{+}$.

$$\begin{aligned} l_1 &= \{a_{x\ (01)}^*, a_{x\ (02)}^*, a_{x\ (03)}^*, \cdots, a_{x\ (08)}^*\}, \\ l_2 &= \{a_{x\ (09)}^*, a_{x\ (10)}^*, a_{x\ (11)}^*, \cdots, a_{x\ (16)}^*\}, \\ &\ \vdots \qquad\qquad \vdots \\ l_8 &= \{a_{x\ (57)}^*, a_{x\ (58)}^*, a_{x\ (59)}^*, \cdots, a_{x\ (64)}^*\}. \end{aligned} \tag{4}$$

2.2. Square transposition schematic testing

Every combination of input and output schemes in the square transposition will result in a transposition method, and each combination must produce a random order of index values. All users can design their input and output schemes. Therefore, random testing needs to be done to ensure that every designed scheme will produce a good transposition method.

Figure 4 shows a testing scheme, in which if each pair of schemes has not yet reached randomness, a scheme can be replaced by another scheme. This research uses three tests of randomness (Frequency Monobit Test, Frequency Test within a Block, and Runs Test) so that if two or three methods are random, the combination of those schemes can be used as a method of transposition.

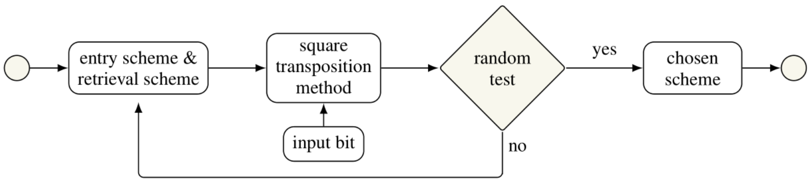

Figure 4. Testing of input and output schemes

3. RESULT AND DISCUSSION

3.1. Square transposition entry scheme

Based on (1), 64-bit is used as input and square size 8×8. Two input schemes are selected with randomly selected index values, the two schemes are given in succession in Figures 5 and 6, respectively.

a_{37}	a_{29}	a_{18}	a_{53}	a_{47}	a_{21}	a_{27}	a_{48}
a_{30}	a_{39}	a_{41}	a_{46}	a_{22}	a_{58}	a_{05}	a_{43}
a_{23}	a_{34}	a_{42}	a_{57}	a_{40}	a_{12}	a_{64}	a_{06}
a_{11}	a_{59}	a_{16}	a_{51}	a_{63}	a_{25}	a_{45}	a_{56}
a_{24}	a_{09}	a_{26}	a_{01}	a_{20}	a_{60}	a_{03}	a_{19}
a_{54}	a_{31}	a_{49}	a_{32}	a_{04}	a_{38}	a_{15}	a_{50}
a_{17}	a_{14}	a_{08}	a_{35}	a_{44}	a_{10}	a_{33}	a_{62}
a_{61}	a_{13}	a_{02}	a_{07}	a_{36}	a_{28}	a_{52}	a_{55}

Figure 5. Input scheme 1

a_{50}	a_{24}	a_{33}	a_{06}	a_{48}	a_{09}	a_{13}	a_{25}
a_{52}	a_{44}	a_{22}	a_{58}	a_{49}	a_{15}	a_{38}	a_{51}
a_{29}	a_{07}	a_{57}	a_{20}	a_{05}	a_{43}	a_{55}	a_{10}
a_{64}	a_{14}	a_{30}	a_{39}	a_{04}	a_{27}	a_{60}	a_{11}
a_{47}	a_{01}	a_{32}	a_{62}	a_{03}	a_{37}	a_{59}	a_{23}
a_{28}	a_{63}	a_{18}	a_{54}	a_{45}	a_{36}	a_{35}	a_{12}
a_{42}	a_{31}	a_{40}	a_{61}	a_{19}	a_{02}	a_{53}	a_{26}
a_{46}	a_{16}	a_{41}	a_{17}	a_{34}	a_{56}	a_{21}	a_{08}

Figure 6. Input scheme 2

3.2. Retrieval scheme design

The retrieval scheme is a rule that takes every bit from a square that previously had a bit from the bit entry process. Here are several retrieval schemes used as pairs of input schemes.

3.2.1. Horizontal retrieval scheme

This design uses the Entry-1 Scheme to insert bits into a square, as shown in Figure 7. The horizontal retrieval process is carried out from the top left corner to the right corner of the square. The order of each bit a_{8i+1} for $i = 0, 1, \cdots, 7$ is always to the left of the first entry of every line to square $(i + 1)$.

The horizontal retrieval scheme results starts from a_{37} based on the index $j = 1$ to $j = 64$ for a_{55}. Thus, square transposition output is obtained which is based on byte, as shown previously in (4), is a schema collection dataset $L = \{l_1, l_2, l_3, \cdots, l_8\}$ where $l_1 = \{a_{37}, a_{29}, \cdots, a_{48}\}$, $l_2 = \{a_{30}, a_{39}, a_{41}, \cdots, a_{43}\}$, $\cdots$, $l_8 = \{a_{61}, a_{13}, a_{02}, \cdots, a_{55}\}$. Transposition results from the retrieval-1 scheme and the horizontal entry schema can be visualized in Cartesian coordinates, where each takes index (i) is abscissa and index enter (j) as ordinate. The results of complete bit retrieval are shown in Figure 8.

a_{37}(01)	a_{29}(02)	a_{18}(03)	a_{53}(04)	a_{47}(05)	a_{21}(06)	a_{27}(07)	a_{48}(08)
a_{30}(09)	a_{39}(10)	a_{41}(11)	a_{46}(12)	a_{22}(13)	a_{58}(14)	a_{05}(15)	a_{43}(16)
a_{23}(17)	a_{34}(18)	a_{42}(19)	a_{57}(20)	a_{40}(21)	a_{12}(22)	a_{64}(23)	a_{06}(24)
a_{11}(25)	a_{59}(26)	a_{16}(27)	a_{51}(28)	a_{63}(29)	a_{25}(30)	a_{45}(31)	a_{56}(32)
a_{24}(33)	a_{09}(34)	a_{26}(35)	a_{01}(36)	a_{20}(37)	a_{60}(38)	a_{03}(39)	a_{19}(40)
a_{54}(41)	a_{31}(42)	a_{49}(43)	a_{32}(44)	a_{04}(45)	a_{38}(46)	a_{15}(47)	a_{50}(48)
a_{17}(49)	a_{14}(50)	a_{08}(51)	a_{35}(52)	a_{44}(53)	a_{10}(54)	a_{33}(55)	a_{62}(56)
a_{61}(57)	a_{13}(58)	a_{02}(59)	a_{07}(60)	a_{36}(61)	a_{28}(62)	a_{52}(63)	a_{55}(64)

Figure 7. Horizontal retrieval scheme

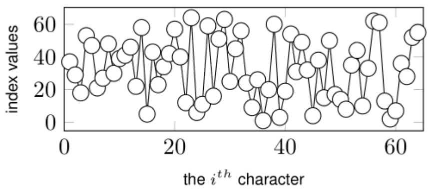

Figure 8. Graphic of the index values

3.2.2. Vertical retrieval scheme

Square transposition also uses an input-1 scheme to input each entry from the square. Retrieval is done vertically from top to bottom, starting at the top right corner entry to the bottom right of the square. In general, every bit of $a_{i\ (j)}$ and the retrieval index $j = (8z + 1)$; $z \in \{0, 1, \cdots, 7\}$. If z is even, the retrieval is done vertically top-down, and if z is odd, the retrieval will be done from the bottom up. The retrieval results are based on bits shown in Figure 9.

The vertical retrieval scheme starts from a_{48} based on the index $j = 1$ to $j = 64$ for a_{37} bits. So the square transposition output is based on bytes $L = \{l_1, l_2, l_3, \cdots, l_8\}$, where $l_1 = \{a_{48}, a_{43}, a_{06}, \cdots, a_{55}\}$, $l_2 = \{a_{52}, a_{33}, a_{15}, \cdots, a_{27}\}, \cdots, l_8 = \{a_{61}, a_{17}, a_{54}, \cdots, a_{37}\}$. The visualization of transposition index of the input-1 scheme and the vertical input scheme is shown in Figure 10.

a_{37}(64)	a_{29}(49)	a_{18}(48)	a_{53}(33)	a_{47}(32)	a_{21}(17)	a_{27}(16)	a_{48}(01)
a_{30}(63)	a_{39}(50)	a_{41}(47)	a_{46}(34)	a_{22}(31)	a_{58}(18)	a_{05}(15)	a_{43}(02)
a_{23}(62)	a_{34}(51)	a_{42}(46)	a_{57}(35)	a_{40}(30)	a_{12}(19)	a_{64}(14)	a_{06}(03)
a_{11}(61)	a_{59}(52)	a_{16}(45)	a_{51}(36)	a_{63}(29)	a_{25}(20)	a_{45}(13)	a_{56}(04)
a_{24}(60)	a_{09}(53)	a_{26}(44)	a_{01}(37)	a_{20}(28)	a_{60}(21)	a_{03}(12)	a_{19}(05)
a_{54}(59)	a_{31}(54)	a_{49}(43)	a_{32}(38)	a_{04}(27)	a_{38}(22)	a_{15}(11)	a_{50}(06)
a_{17}(58)	a_{14}(55)	a_{08}(42)	a_{35}(39)	a_{44}(26)	a_{10}(23)	a_{33}(10)	a_{62}(07)
a_{61}(57)	a_{13}(56)	a_{02}(41)	a_{07}(40)	a_{36}(25)	a_{28}(24)	a_{52}(09)	a_{55}(08)

Figure 9. Vertical Retrieval Scheme

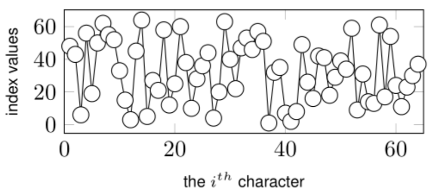

Figure 10. Graphic of the index values

3.2.3. Zigzag retrieval scheme

The input-2 scheme is used in Figure 6 which the retrieval scheme is done in zigzag form from the lower left to the upper right. The retrieval plots are based on index values $j = 1$ to $j = 64$, which the complete plots are shown in Figure 11. Retrieval starts from a_{46} to a_{25}, so that the square transposition output can be obtained based on byte $L = \{l_1, l_2, l_3, \cdots, l_8\}$ where $l_1 = \{a_{46}, a_{42}, a_{16}, \cdots, a_{63}\}$, $l_2 = \{a_{40}, a_{17}, a_{64}, \cdots, a_{29}\}$, $\cdots$ $l_8 = \{a_5, a_{11}, a_{09}, \cdots, a_{25}\}$. The geometric interpretation of the value of the zigzag input-2 and zigzag retrieval scheme is shown in Figure 12.

a_{50}(29)	a_{24}(37)	a_{33}(44)	a_{06}(50)	a_{48}(55)	a_{09}(59)	a_{13}(62)	a_{25}(64)
a_{52}(22)	a_{44}(30)	a_{22}(38)	a_{58}(45)	a_{49}(51)	a_{15}(56)	a_{38}(60)	a_{51}(63)
a_{29}(16)	a_{07}(23)	a_{57}(31)	a_{20}(39)	a_{05}(46)	a_{43}(52)	a_{55}(57)	a_{10}(61)
a_{64}(11)	a_{14}(17)	a_{30}(24)	a_{39}(32)	a_{04}(40)	a_{27}(47)	a_{60}(53)	a_{11}(58)
a_{47}(07)	a_{01}(12)	a_{32}(18)	a_{62}(25)	a_{03}(33)	a_{37}(41)	a_{59}(48)	a_{23}(54)
a_{28}(04)	a_{63}(08)	a_{18}(13)	a_{54}(19)	a_{45}(26)	a_{36}(34)	a_{35}(42)	a_{12}(49)
a_{42}(02)	a_{31}(05)	a_{40}(09)	a_{61}(14)	a_{19}(20)	a_{02}(27)	a_{53}(35)	a_{26}(43)
a_{46}(01)	a_{16}(03)	a_{41}(06)	a_{17}(10)	a_{34}(15)	a_{56}(21)	a_{21}(28)	a_{08}(36)

Figure 11. Zigzag retrieval scheme

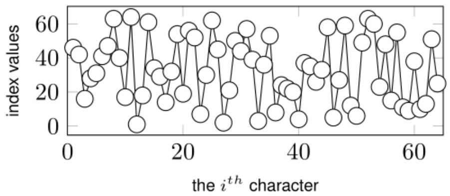

Figure 12. Graphic of the index values

3.2.4. Rice plow retrieval scheme

The transposition technique by adopting the rice plow process can be done with the assumption of a square as a rice field plot. Each bit plot is adjusted to the rice plow process starting from the outside point towards the midpoint, which the complete plots are shown in Figure 13. The input-2 scheme is used to fill in each input from the square so that the retrieval using rice plow plot can be carried out.

The retrieval process starts from the lower right corner (a_{08}) with a rotating plot around the square towards the center (a_{04}). The value of the transposition output index of the input-2 scheme and rice scheme retrieval is $L = \{l_1, l_2, \cdots, l_8\}$; where $l_1 = \{a_{08}, a_{21}, a_{56}, \cdots, a_{46}\}$, $l_2 = \{a_{42}, a_{28}, a_{47}, \cdots, a_{24}\}$, $\cdots$ $l_8 = \{a_{43}, a_{43}, a_{27}, \cdots, a_{04}\}$. The visualization of the transposition index value is shown in Figure 14.

a_{50}(15)	a_{24}(16)	a_{33}(17)	a_{06}(18)	a_{48}(19)	a_{09}(20)	a_{13}(21)	a_{25}(22)
a_{52}(14)	a_{44}(39)	a_{22}(40)	a_{58}(41)	a_{49}(42)	a_{15}(43)	a_{38}(44)	a_{51}(23)
a_{29}(13)	a_{07}(38)	a_{57}(55)	a_{20}(56)	a_{05}(57)	a_{43}(58)	a_{55}(45)	a_{10}(24)
a_{64}(12)	a_{14}(37)	a_{30}(54)	a_{39}(63)	a_{04}(64)	a_{27}(59)	a_{60}(46)	a_{11}(25)
a_{47}(11)	a_{01}(36)	a_{32}(53)	a_{62}(62)	a_{03}(61)	a_{37}(60)	a_{59}(47)	a_{23}(26)
a_{28}(10)	a_{63}(35)	a_{18}(52)	a_{54}(51)	a_{45}(50)	a_{36}(49)	a_{35}(48)	a_{12}(27)
a_{42}(09)	a_{31}(34)	a_{40}(33)	a_{61}(32)	a_{19}(31)	a_{02}(30)	a_{53}(29)	a_{26}(28)
a_{46}(08)	a_{16}(07)	a_{41}(06)	a_{17}(05)	a_{34}(04)	a_{56}(03)	a_{21}(02)	a_{08}(01)

Figure 13. Rice plow retrieval scheme

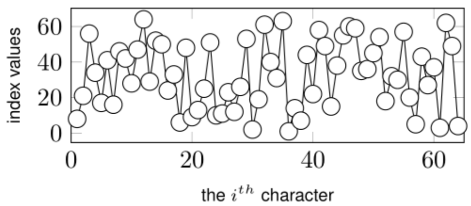

Figure 14. Graphic of the index values

3.3. Testing of randomness on index values

The method used in randomness testing is Mono Bit frequency Test, Bit Block frequency Test, and Run Test, with $\alpha = 0.01$. Each transposition index value is declared as random if two or three test results have p-value $> \alpha$. The complete test results are shown in Table 1. The combination of each input scheme and retrieval scheme is carried out to see how well the pair of schemes are designed or selected so that square transposition can produce a random index value.

The schemes of input 1 & vertical output obtain the highest p-value with the average value of 0.273, and the ones with the lowest score are the schemes of input 1 & zigzag scheme with a smaller average of p-value, which is 0.101. Overall, all pairs of input and output schemes can maintain the p-value that results in a random index value and the pair of schemes can produce a better index value when compared to the transposition method in AES and DES algorithms.

Table 1. Randomness test result for each scheme

2*No	2*Input & Retrieval Scheme	*p-value*			2*Result
		Mono Bit	Block Bit	Run-Test	
1	Input-1 & Horizontal Scheme	0.1341	0.1230	0.1853	random
2	Input-1 & Vertical Scheme	0.2727	0.2194	0.1432	random
3	Input-2 & Zigzag Scheme	0.1204	0.2282	0.2031	random
4	Input-2 & Rice Plow Scheme	0.1950	0.1917	0.1981	random
5	Input-1 & Zigzag Scheme	0.1012	0.1118	0.2116	random
6	Input-1 & Rice Plow Scheme	0.1018	0.2014	0.1102	random
7	Input-2 & Horizontal Scheme	0.1210	0.1052	0.2056	random
8	Input-2 & Vertical Scheme	0.2044	0.2015	0.1186	random
9	DES	2.569×10^{-8}	0.0539	9.172×10^{-6}	non random
10	AES	4.251×10^{-8}	0.0042	1.456×10^{-4}	non random

The use of output-1 and output-2 schemes plays an important role in yielding the output of random index values. Selection of a pair of schemes using a combination of horizontal, vertical, zigzag, and plow or others that have a patterned index will generate poor transposition index value. It happens because the input and output scheme has the same or similar line direction.

3.4. Correlation testing

Correlation value (r) can be used to see the magnitude of the relationship between input (x) and output (y) of statistically related algorithms. The correlation interval is $-1 \leq r \leq 1$, and if r approaches 0, then the algorithm is able to make the input and output not statistically related. In this condition, if $r < 0$, the absolute value $|r|$ can be used to find out the distance r from 0.

Correlation testing uses three plaintext inputs which it is expected to represent text variations that might be used by users. Input "fti uksw" is to represent traditional text input because usually, users use it. The second more extreme test is the use of the same input, which is "xyyyyyyyy" (not "yyyyyyyyy" because this correlation formula is undefined). The third test is "$aL4t1G4" which also represents a variety of symbols, numbers, and letters that are used as input.

The results obtained in Table 2 show that the output of each scheme of the square transposition has an average correlation value close to 0. Thus, it indicates that the relationship between input and output is not related statistically. Consequently, the square transposition succeeds in disguising the information, so that the distribution of redundancies occurs well and will certainly increase the diffusion effect on the cryptography algorithm.

Table 2. Testing result of input-output correlation

2*No	2*Transposition Method	Correlation Value $\|r\|$			2*Average
		fti uksw	xyyyyyyy	$aL4t1G4	
1	Input-1 & Horizontal Retrieval	0.249	0.331	0.217	0.266
2	Input-1 & Vertical Retrieval	0.162	0.127	0.142	0.162
3	Input-2 & Zigzag Retrieval	0.254	0.267	0.324	0.254
4	Input-2 & Rice Plow Retrieval	0.313	0.375	0.252	0.313
5	Input-1 & Zigzag Retrieval	0.112	0.009	0.018	0.112
6	Input-1 & Rice Plow Retrieval	0.016	0.090	0.040	0.016
7	Input-2 & Horizontal Retrieval	0.138	0.268	0.265	0.138
8	Input-2 & Vertical Retrieval	0.076	0.098	0.184	0.076
9	DES	0.342	0.126	0.374	0.342
10	AES	0.376	0.429	0.277	0.376

The transposition of DES and AES algorithms has resulted in higher average correlation values than the value from the schematic combination of square transposition so that it can be said that each pair of schemes can generate a better transposition algorithm. Of course, the use of square transposition in cryptography will increase the strength of overall cryptographic algorithms. Optimization of the transposition process using square transposition is a part that needs to be done by cryptographers to improve or modify the weak parts of the algorithm.

4. CONCLUSION

The determination of a pair of input and output schemes in square transposition should be based on schemes that have different lines to obtain a good transposition process. Combination of schemes that were carried out produced less patterned geometric visualization that oscillates irregularly, so that the transposition method could generate random index values. This result is also seen in randomness testing in which the overall obtained p-value is greater which $\alpha = 1\%$ so that the square transposition can produce better a transposition method when it is compared to AES and DES values which the index is not random. Square transposition produces an average correlation value closer to 0 for testing the text input when compared to AES and DES transpositions. Thus, the square transposition manages to disguise the information on the input so that it is not visible in the output. Besides, the square transposition can spread the distributed redundancies well, so that it will increase the diffusion effect on the cryptographic algorithm. The result shows that the algorithm in the square transposition optimizes the transposition process that previously has non-random index values. This design optimizes algorithm processes by concentrating on the diffusion effect and by not giving a burden on the complexity of time and space. Algorithm modification is a process that every cryptographer needs to do to produce a more efficient algorithm in cryptography to secure information.

ACKNOWLEDGEMENT

The researcher would like to thank the Bureau of Research, Publication and Community Service (BP3M) of Universitas Kristen Satya Wacana Salatiga for providing the funding assistance through the Fundamental Internal Research Scheme in 2018/2019.

REFERENCES

[1] W. M. Daley, "Federal Information Processing Standards Publication," *Data Encryption Standard (DES)*, U.S. Department of Commerce: National Institute of Standards and Technology (NIST), 1979, pp. 1-22.

[2] NIST, "Federal Information Processing Standards Publication," *Advanced Encryption Standard (AES)*, U.S. Department of Commerce: National Institute of Standards and Technology (NIST), November 2001, pp. 1-47.

[3] A. Biryukov and C. De Cannière, "Data Encryption Standard (DES)," *IBM Journal of Research and Development, Springer*, 2011, pp. 243-250.

[4] J. Daemen and V. Rijmen, *The Design of Rijndael: AES The Advanced Encryption Standard*, Springer-Verlag, 2001.

[5] M. Yang, B. Xiao and Q. Meng, "New AES Dual Ciphers Based on Rotation of Columns," *Wuhan University Journal of Natural Sciences, Springer*, Vol. 24, pp. 93-97, March 2019, doi: 10.1007/s11859-019-1373-y.

[6] A. Arab, M. J. Rostami and B. Ghavami, "An image encryption method based on chaos system and AES Algorithm," *The Journal of Supercomputing, springer*, vol. 75, pp. 6663-6682, May 2019, doi: 10.1007/s11227-019-02878-7.

[7] A. A. Thinn and M. M. S. Thwin, "Modification of AES Algorithm by Using Second Key and Modified SubBytes Operation for Text Encryption," *Computational Science and Technology part of Lecture Notes in Electrical Engineering, Springer*, vol. 481, pp. 435-444, August 2018, doi: 10.1007/978-981-13-2622-6-42.

[8] C. R. Dongarsane, D. Maheshkumar and S. V. Sankpal, "Performance Analysis of AES Implementation on a Wireless Sensor Network", *Tech. Soc. Springer*, pp. 87-93, November 2019, doi: 10.1007/978-3-030-164843-3-9.

[9] C. Ashokkumar, R. M. Bholanath, S. V. Bhargav and B. L. Menezes, "S-Box Implementation of AES Is Not Side Channel Resistant," *Journal of Hardware and Systems Security, Springer*, vol. 4, no. 2, pp. 86-97, December 2019, doi: 10.1007/s41635-019-00082-w.

[10] T. Manojkumar, P. Karthigaikumar and V. Ramachandran, "An Optimized S-Box Circuit for High Speed AES Design with Enhanced PPRM Architecture to Secure Mammographic Images," *Journal of Medical Systems, Springer*, vol. 43, no. 31, p. 31, January 2019, doi: 10.1007/s10916-018-1145-9.

[11] S. D. Putra, M. Yudhiprawira, S. Sutikno, Y. Kurniawan and A. D. Ahmad, "Power Analysis Attack Against Encryption Devices: A Comprehensive Analysis of AES, DES, and BC3," *TELKOMNIKA Telecommunication, Computing, Electronics and Control*, vol. 17, no. 3, pp. 2182-1289, June 2019, doi: 10.12928/TELKOMNIKA.v17i3.9384.

[12] C. S. 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, Computing, Electronics and Control*, vol. 17, no. 5, pp. 2400-2409, October 2019, doi: 10.12928/TELKOMNIKA.v17i5.9570.

[13] G. C. Prasetyadi, R. Refianti and A. B. Mutiara, "File Encryption and Hiding Application Based on AES and Append Insertion Steganography," *TELKOMNIKA Telecommunication, Computing, Electronics and Control*, vol. 16, no.1, pp. 361-367, February 2018, doi: 10.12928/TELKOMNIKA.v16i1.6409.

[14] B. F. Cruz, K.N. Dominggo, E. Froilan, J. B. Cotiangco and C. B. Hilario, "Expanded 128-bit Data Encryption Standard," *International Journal of Computer Science and Mobile Computing*, vol. 6, no. 8, pp. 133-142, August 2017.

[15] C. A. Sari, E. H. Rachmawanto and C. A. Haryanto, "Cryptography Triple Data Encryption Standard (3DES) for Digital Image Security," *Scientific Journal of Informatics*, vol. 5, no. 2, pp. 105-117, November 2018.

[16] S. Pavithra, P. Muthukannan and V. Prabhakaran, "An Enhanced Cryptographic Algorithm Using Bi-Modal Biometrics," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 8, no. 11, pp. 2575-2582, September 2019, doi: 10.35940/ijitee.K1870.0981119.

[17] E. R. Arboleda, J. L. Balaba and J. L. Espineli, "Chaotic Rivest-Shamir-Adlerman Algorithm with Data Encryption Standard Scheduling," *Bulletin of Electrical Engineering and Informatics* , vol. 6, no. 3, pp. 219–227, September 2017, doi: 10.11591/eei.v6i3.627.

[18] P. B. Mane and A. O. Mulani, "High Speed Area Efficient FPGA Implementation of AES Algorithm," *International Journal of Reconfigurable and Embedded Systems (IJRES)*, vol. 7, no. 3, pp. 157-165, November 2018, doi: 10.11591/ijres.v7.i3.pp157-165.

[19] S. D. Putra, A. S. Ahmad, S. Sutikno, Y. Kurniawan and A. D. W. Sumari, "Revealing AES Encryption Device Key on 328P Microcontrollers with Differential Power Analysis," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 6, pp. 5144-5152, December 2018, doi: 10.11591/ijece.v8i6.pp.5144-5152.

[20] R. Srividya and B. Ramesh, "Implementation of AES using Biometric," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 5, pp. 4266-4276, October 2019, doi: 10.11591/ijece.v9i5.pp4266-4276.

[21] J. M. B. Espalmado and E. R. Arboleda, "DARE Algorithm: A New Security Protocol by Integration of Different Cryptographic Techniques," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 7, no. 2, pp. 1032–1041, April 2017, doi: 0.11591/ijece.v7i2.pp1032-1041.

[22] R. Rizky, Rojali and A Kurniawan, "Improvement of Advanced Encryption Standard Algorithm with Shift row and S.box Modification Mapping in Mix Column," *The 2nd International Conference on Computer Science and Computational Intelligence*, vol. 116, pp. 401–407, Feb 2017, doi: 10.1016/j.procs.2017.10.079.

[23] H. V. Gamido, A. M. Sison and R. P. Medina, "Implementation of Modified AES as Image Encryption Scheme," *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, vol. 6, no. 3, pp. 301-308, September 2018, doi: 10.52549/ijeei.v6i3.490.

[24] M. Aledhari, A. Marhoon, A. Hamad and F. Saeed, "A New Cryptography Algorithm to Protect Cloud-Based Healthcare Services," *IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*, 2017, pp. 37-43, doi: 10.1109/CHASE.2017.57.

[25] P. Jindal, A. Kaushik and K. Kumar, "Design and Implementation of Advanced Encryption Standard Algorithm on 7th Series Field Programmable Gate Array," *2020 7th International Conference on Smart Structures and Systems (ICSSS)*, 2020, pp. 1-3, doi: 10.1109/ICSSS49621.2020.9202114.

[26] E. M. D. Reyes, A. M. Sison and R. Medina, "Modified AES Cipher Round and Key Schedule," *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, vol. 7, no. 1, 29-36, 2018, doi: 10.52549/ijeei.v7i1.652.

[27] P. Sharma, "A New Image Encryption using Modified AES Algorithm and its Comparision with AES," *International Journal of Engineering Research Technology (IJERT)*, vol. 9, no. 8, pp. 194-197, August 2020