One to many (new scheme for symmetric cryptography)

Alz Danny Wowor[1], Bambang Susanto[2]
[1]Department of Informatics Engineering, Faculty of Information Technology, Satya Wacana Christian University, Salatiga, Indonesia
[2]Department of Mathematics, Faculty of Science and Mathematics, Satya Wacana Christian University, Salatiga, Indonesia

Article Info

Article history:

Received Jun 19, 2020
Revised Oct 07, 2020
Accepted Feb 08, 2021

Keywords:

Cryptanalyst
One-to-many scheme
Symmetric cryptography

ABSTRACT

Symmetric cryptography will always produce the same ciphertext if the plaintext and the given key are the same repeatedly. This condition will make it easier for cryptanalysts to perform cryptanalysis. This research introduces a one-to-many cryptography scheme, which can produce different ciphertexts even if the input given is the same repeatedly. The one-to-many encryption scheme can produce several ciphertexts with differences of up to 50%. The avalanche effect test obtained an average of 52.20%, better than modern cryptography Blowfish by 25.46% and 6% better than advanced encryption standard (AES). One-to-many can produce different n-ciphertexts, which will certainly make it more difficult for cryptanalysts to perform cryptanalysis and require n-times longer to break than other symmetric cryptography.

Corresponding Author:

Alz Danny Wowor
Department of Informatics Engineering, Faculty of Information Technology, Satya Wacana Christian University
Jl. Notohamidjojo 1-10, Blotongan, Salatiga, Indonesia
Email: alzdanny.wowor@uksw.edu

1. INTRODUCTION

The need for information security becomes an extraordinary transformation in the field of cryptography. Cryptographers create algorithms to secure information by emphasizing the strength of mathematics in taking advantage of the complexity of algorithms and key secrecy. The more complex the algorithm design, the better the information security. On the other hand, the speed of the process in algorithm is also taken into account. This is the focus among cryptographers, namely how to create a powerful algorithm against the attack of cryptanalysts yet has a fast processing time.

Symmetric cryptography is still an interesting study and is still very feasible to study [1]. One of them is the advanced encryption standard (AES) has become a popular criterion and is widely used commercially. There are several studies [2]-[14] that use algorithms recommended by the National Institute of Standards and Technology (NIST) as the information security, although several studies have been claimed to break the algorithm [15]-[20].

Today's advancement of computer technology also has created a challenge for the development of cryptography. The idea of quantum computation can enhance various cryptanalysis techniques because quantum mechanical system calculation will produce a much less time complexity than that of classical calculation model [21], [22]. Computational advancement certainly encourages cryptographers to review the cryptographic schemes that will be used.

There a lot of methods that can be done to make an algorithm to be more resistant to attacks. Henriques and Vernekar [23] combined symmetric and asymmetric algorithms to secure communication between devices in an IoT system. In [24] proposed the use of acoustic echo cancellation (AEC) and elliptic curve cryptography (ECC) encryption techniques with a combination of symmetric and asymmetric algorithms. In [25] modified shift row and mix column in AES, so that the processing time can be faster. In [26] developed a cryptographic technique based on quantum key distribution (QKD). In [27] designed and implemented a real-time cryptographic algorithm to secure medical devices that provide a high level of security for remote health monitoring systems.

The existing symmetrical cryptography is the type of one-to-one encryption scheme, which means that every input plaintext and key always produces the same ciphertext every time a process is carried out. In solving of cryptography, a criptanalyst will certainly try to look for patterns from percipext and makes an analysis of keys and plaintexts. The existing cryptography algorithms sometimes at certain parts are very vulnerable to extreme testing and as a result the patterns in ciphertext will be easier to be found.

This research designs a new scheme that can answer that problem, which is the one-to-many scheme. This scheme will create different ciphertext even though the input plaintext and keys are always the same. Any changes that occur in ciphertext will take longer time and certainly will also make it difficult for cryptanalysts to see the relationship between plaintext and ciphertext. Thus, this scheme certainly is more secure than one-to-one encryption scheme. The initial design of this scheme is more on how to get several balanced ciphertext than on the complexity of algorithm in the encryption-decryption process.

2. PROPOSED RESEARCH

2.1. One to many scheme

One-to-many is an encryption scheme that can create different chipertexts, although using the same plaintext and key, and decryption process that can retrieve the plaintext. Etymologically, one-to-many consists of two words, “one” is a plaintext, and “many” is a process generating a different ciphertext for each time it is encrypted.

$$E_K(p_a) = c_{a_i} \tag{1}$$

The one-to-many scheme encryption is described in (1). The encryption (E_k) uses plaintext (p_a) and key as inputs, which can create ciphertext $c_{a_i}, i \in \mathrm{Z}^+$. All encryption processes can create different c_{a_i} although with the same input.

$$D_K(c_{a_i}) = p_a \tag{2}$$

In contrast, the process of D_K decryption function in (2) uses a key and ciphertext as inputs. A one-to-many scheme should be able to return each different ciphertext (c_{a_i}) resulting from the encryption process to be a plaintext. This study uses a random number generator to show that the one-to-many scheme can be applied in cryptographic processes as shown in Figure 1.

2.2. Implementation of one-to-many scheme

The design of one-to-many algorithm scheme is shown in Figure 1. The plaintext input (P) and key (K) are in the form of text. K is processed into seed (x_0) in cryptographically secure pseudo random number generator (CSPRNG) chaos which will produce n-number of random numbers sequence $\{A_1, A_2, \cdots, A_n\}$. The chose key process scheme will manage and select the compatible key for the encryption-decryption process so that it produces the appropriate plaintext.

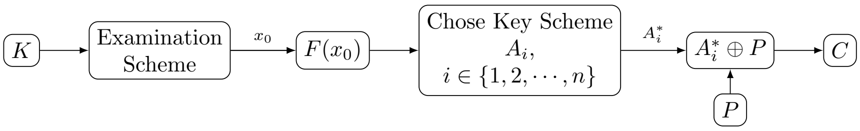


Figure 1. General scheme of one to many encryption

$K = \{k_1, k_2, \cdots, k_n\}$ is the symbol of the key and $P = \{p_1, p_2, \cdots, p_n\}$ is the symbol of plaintext. Each $k_i \in K$ plays role as input to the examination scheme [28], and produces x_0 as the seed for the CSPRNG chaos function (F).

$$F(x_0) = \{A_1, A_2, \cdots A_n\} \tag{3}$$

Each $A_i, i = 1, 2, \cdots, n$ is the data set of CSPRNG chaos random numbers, where $A_1 = \{a_{11}, a_{12}, \cdots, a_{1k}\}$, $A_2 = \{a_{21}, a_{22}, \cdots, a_{2k}\}, \cdots, A_n = \{a_{n1}, a_{n2}, \cdots, a_{nk}\}$. From (3), using the chosen key A_n^* from the key selection scheme, the encryption process is obtained as:

$$E_K \to P \oplus A_i^* = \{c_1 = p_1 \oplus a_{1r}^*,\ c_2 = p_2 \oplus a_{2r}^*,\ \cdots,\ c_n = p_n \oplus a_{nr}^*\} \tag{4}$$

In the general, the following equation is obtion:

$$c_i = p_i \oplus a^*_{ir} \tag{5}$$

The decryption process can also be done by generating each random number using key input as well.

Let $K = \{k_1, k_2, \cdots, k_n\}$ as the key, and ciphertext $C = \{c_1, c_2, \cdots, c_n\}$. Each random number from CSPRNG chaos is obtained from $F(K) = \{A_1, A_2, \cdots A_n\}$. The one-to-many scheme will select A^*_i as the chosen key for the decryption process.

$$D_K \to C \oplus A^*_i = \{p_1 = c_1 \oplus a^*_{1r},\ p_2 = c_2 \oplus a^*_{2r},\ \cdots,\ p_n = c_n \oplus a^*_{nr}\} \tag{6}$$

So that plaintext will be recovered as a decryption process.

$$p_i = c_i \oplus a^*_{ir} \tag{7}$$

2.3. Key generation

The key generation starts from the examination process [28], each key input (k_i) and the index constant (b_i), where $k_i, b_i \in \mathrm{Z}^{256}$ for $i = 1, 2, \cdots, 8$ which is used to find the ratio $x_0 = p_a/q_a$, where p_a and q_a are obtained from (8) and (9).

$$p_a = (k_1 + k_2 + k_3 + k_4 + k_5 + k_6)/6 = \sum_{i=1}^{6} k_i/6 \tag{8}$$

$$q_a = k_1b_1 + k_2b_2 + k_3b_3 + k_4b_4 + k_5b_5 + k_6b_6 + k_7b_7 + k_8b_8 = \sum_{i=1}^{8} k_ib_i \tag{9}$$

The value of x_0 is used as the seed in (10) that generates a CSPRNG chaos-based number serial. This process is in accordance with the search function $F(x_0)$ as shown in Figure 1. The random number will be used as a key in the encryption or decryption process.

$$x_{n+1} = tx_n(1 - x_n) \tag{10}$$

Each chaotic number is in decimal form and cannot be used as a key, so it is converted to an integer using the truncation function in (11).

$$T(x, size) = \|x \times 10^{count}\|, x \neq 0 \tag{11}$$

The “chose key scheme” is the most important process, because here algorithm will choose the key availability and set which key will be selected as the chosen key A^*_n to be used in the encryption process in (4) or decryption in (6). The designed algorithm can produce several different and balanced key sets $\{A_1, A_2, \cdots, A_n\}$ as shown earlier in (3). After selecting a set which will be used as the key, algorithm will provide a marker that is included in the ciphertext, so that in the process of decryption, algorithm can recognize the marker and will refer to the set of keys used.

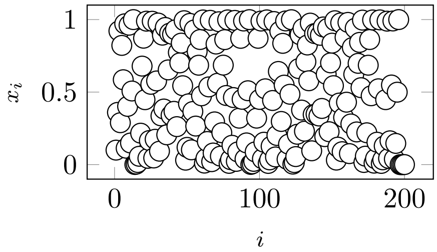
3. RESULT AND ANALYSIS

3.1. Test of the key changes

Test of key changes in one-to-many scheme is carried out to see how well an algorithm produces ciphertext if there is a small change (1-bit change) in the key. The plaintext “satya wacana” and two keys “fti uksw” and “ftj uksw” are selected, where the change from the letter i to j is a 1-bit difference. This test also uses the correlation statistical test and the mean difference, to see the difference between each ciphertext produced.

i	x_i	bil-1	bil-2	bil-3	bil-4	bil-5
1	0.099998932494109	099	998	932	494	109
2	0.359996583976590	359	996	583	976	590
3	0.921596174007104	921	596	174	007	104
4	0.289026664250287	289	026	664	250	287
5	0.821961006410555	821	961	006	410	555
6	0.585364441404410	585	364	441	404	410
7	0.970851648574852	970	851	648	574	852

(a)



(b)

Figure 2. Logistic function iteration with “fti uksw” key: (a) data of the first 7 iteration and (b) the first 200 iteration

The first test uses the key “fti uksw”, and the index value $B = 1, 2, \cdots, 8$ is selected. Based on (8) and (9), $x_0 = 0.0256580696623239$ is obtained. To generate a random value of CSPRNG chaos, value $t = 4$ is used in (10). The results of the first 7 iteration values are shown in Figure 2(a) and the results of the first 200 iterations are shown in n Figure 2(b). In cryptography, keys must be in integer number. Therefore, the result of iteration is divided to be five integers, i.e., set-i, $\{1, 2, 3, 4, 5\}$ respectively. Each set-i is A_i in (3), which is used as the key for encryption-decryption processes in (5) and (7).

The scatter plot in Figure 2(b) is a visualization of the first 200 iterations, which proves that the logistic function with the key input “fti uksw” can generate random numbers. Based on the table in Figure 2(a), there are five datasets that can be used as keys. Using (3), we obtain A_1 = set-1, A_2 = set-2, $\cdots$, A_5 = set-5.

Table 1. Key test “fti uksw”

i-th encryptions	Ciphertext-i	Value correlation	Category correlation	Mean difference
1	D6C80D9A966941D2F422035C0720232E20	0.393295	Fair	62.06
2	5945C893228CCA237235F29822B3B46C21	0.086143	Very low	68.69
3	17A8221167D9FFE5653E3A00D8F456B922	−0.515177	Fair	99.81
4	61317B73FBB4B5EAD47BFB9FAF479B2523	0.232599	Low	77.00
5	E0AFDC998EB4CFCA719639F391581FC824	0.343136	Low	82.06

Based on (4), the key selection process can be carried out. The selection scheme will choose the key that will be used in the encryption process. Table 1 shows the result of five encryptions using the key “fti uksw”. These results indicate that one-to-many scheme can produce different ciphertext for each encryption process. The difference between each ciphertext and plaintext can be seen using the correlation value, which respectively there are two ciphertexts that have relationship with plaintexts in moderate and low categories, and one ciphertext that has a relationship with the very low category.

Mean difference is the average absolute difference between plaintext and ciphertext. The greater the distance between plaintext and ciphertext, the better it will be. The results obtained in Table 1 show the mean difference value of different ciphertexts. These results provide information that one-to-many scheme can create different ciphertext for each encryption. Thus, cryptanalysts will need more work to be able to solve this algorithm.

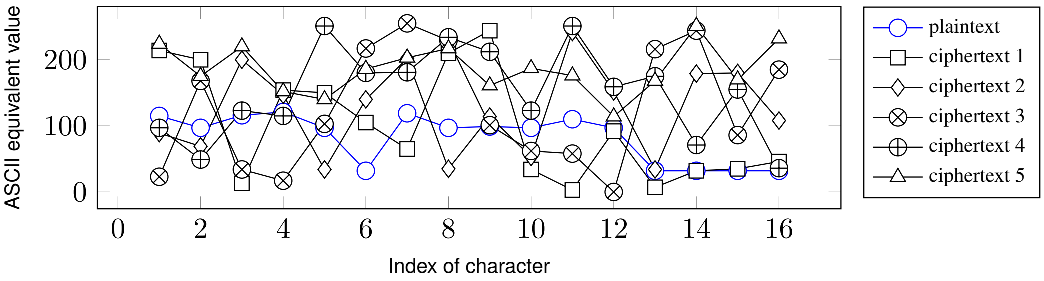
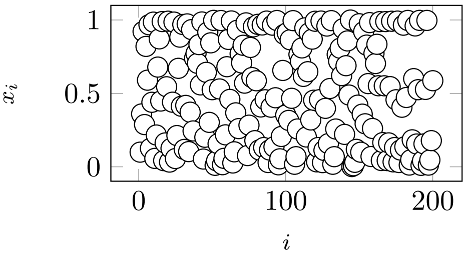


Figure 3. Comparison plot of plaintext and ciphertext for key “fti uksw”

Visualization of each ciphertext of the key “fti uksw” in Cartesian coordinates in Figure 3 shows a sequence of different numbers and do not form a pattern that facilitates direct guessing. It appears that each ciphertext obtained is different although with the same input. This result indicates that one-to-many scheme can make plaintext and ciphertext that do not have a strong relationship statistically.

i	x_i	set-1	set-2	set-3	set-4	set-5
1	0.100089355076193	100	089	355	076	193
2	0.360285904306499	360	285	904	306	499
3	0.921919885858189	921	919	885	858	189
4	0.287934439669651	287	934	439	669	651
5	0.820112792487100	820	112	792	487	100
6	0.590111200344443	590	111	200	344	443
7	0.967519886289935	967	519	886	289	935

(a)



(b)

Figure 4. Logistic function iteration with "ftj uksw" key: (a) data of the first 7 iteration and (b) the first 200 iteration

The second test using the key "ftj uksw", and using the same index value $B = 1, 2, \cdots, 8$, $x_0 = 0.0256818986893376$ is obtained. The results of the first 7 iteration values are shown in Figure 4(a) and the results of the first 200 iterations are shown in n Figure 4(b). Taking the value of $t = 4$, the iteration results are obtained as shown in Figure 4(a). Five datasets A_1 = set-1, A_2 = set-2, $\cdots$, A_5 = set-5 are used as keys in the encryption-decryption processes.

The iteration result of the logistic function with the "ftj uksw" key is very different from the "fti uksw" as shown in Figure 2(b) and Figure 4(b). These results indicate that the examination algorithm is designed to be able to produce a unique initialization value of x_0, so it is very sensitive towards small (1-bit) changes in the input. The nature of the butterfly effect, which has a large effect on output even though it is a small change in input, also has a significant effect on the encryption results. This can be seen in the differences of each ciphertext as shown in Table 1 and Table 2.

Table 2. Key test "ftj uksw"

i-th encryptions	Ciphertext-i	Value correlation	Category correlation	Mean difference
1	D7C90D98956E3EDE1A3AA739C787E0FD20	−0.39498	Fair	96.38
2	CC7E0B1FD18F7E1DBAF874AB3EC9CC1F21	−0.06211	Very low	81.31
3	D6E9E93079E8EDD0EE9C26BCAB49093222	0.342195	Low	91.25
4	BF93CE1648789816ADBBB0B59A89119323	0.182852	Very low	75.44
5	34533007CBCC284C4A65C235F807F6DC24	−0.58524	Fair	89.81

Table 2 shows the test results using the key "ftj uksw", which it indicates that a one-to-many scheme can produce five different ciphertexts at each encryption process. Statistically, the resulting correlation value is close to 0 with sufficient and very low categories. It indicates that the algorithm in making of plaintext and ciphertext has no statistical relationship. The visualization in Figure 5 also shows that each ciphertext produced is different although with the same input. Test of 1-bit key difference shows that one-to-many scheme can produce different ciphertext, even though given key inputs with a small different. The visualization of the graphic shows a different ciphertext which it produces a different sequence of numbers and does not form a pattern that facilitates direct guessing, and as a result, cryptanalysts will need more work to solve this algorithm.

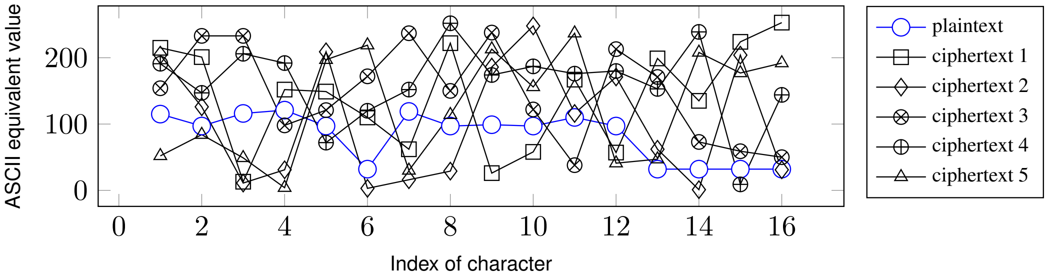


Figure 5. Comparison plot of plaintext and ciphertext for key "ftj uksw"

3.2. Bit differences on each ciphertext

The next test is to see the bit difference on each ciphertext. The bigger the bit difference for each ciphertext, the better it is. The one-to-many scheme produces five ciphertexts, so that the number of comparisons for each ciphertext is $C_2^5 = 10$. Table 3 shows the comparison of ciphertexts for each key. With m, n to be the selected ciphertext, $m = n \in \{1, 2, 3, 4, 5\}$. The symbol "$><$" denotes the comparison of each ciphertext.

Table 3. Bit differences on each ciphertext

No	Ciphertext *m* >< Ciphertext *n*	"fti uksw"	"ftj uksw"
1	1 >< 2	63	72
2	1 >< 3	69	60
3	1 >< 4	72	66
4	1 >< 5	64	61
5	2 >< 3	66	62
6	2 >< 4	65	64
7	2 >< 5	65	67
8	3 >< 4	69	54
9	3 >< 5	65	73
10	4 >< 5	66	67

The average change of bit ciphertext for the key "fti uksw" is 66.4 bits or 51.87%. Meanwhile, for the key "ftj uksw" an average of 64.6 bits (50.47%) is obtained. The minimum score for the overall data is $54(42.1875\%)$, and the maximum score is $73(57.03\%)$, and overall, the mean ciphertext difference is 50%. These results indicate that one-to-many scheme can produce different ciphertext and balanced ciphertext.

3.3. Differences test plaintext-ciphertext

This test is done to see whether the change in 1 bit can make a difference in the overall ciphertext produced. The greater the distance between plaintext and ciphertext indicates the algorithm that is designed is also getting better. Conversely, the smaller the difference between the plaintext and the ciphertext produced the less well the algorithm designed. The absolute difference between plaintext and ciphertext is shown in Table 4 using key "fti uksw" and "ftj uksw".

Table 4. Differences ciphertext-plaintext by key

Char-i	"fti uksw"					"ftj uksw"				
	Cp 1	Cp 2	Cp 3	Cp 4	Cp 5	Cp 1	Cp 2	Cp 3	Cp 4	Cp 5
1	99	26	92	18	109	100	89	99	76	63
2	103	28	71	48	78	104	29	136	50	14
3	103	84	82	7	104	103	105	117	90	68
4	33	26	104	6	32	31	90	73	99	114
5	53	63	6	154	45	52	112	24	25	106
6	73	108	185	148	148	78	111	200	88	172
7	54	83	136	62	88	57	7	118	33	79
8	113	62	132	137	105	125	68	111	75	21
9	145	15	2	113	14	73	87	139	74	25
10	63	44	35	26	53	39	151	59	90	4
11	107	132	52	141	53	57	6	72	66	84
12	5	55	97	62	146	40	74	91	84	44
13	25	2	184	143	113	167	30	139	122	216
14	0	147	212	39	56	103	169	41	105	25
15	3	148	54	123	1	192	172	23	15	214
16	14	76	153	5	168	221	1	18	115	188

It can be seen from both tables that the plaintext-ciphertext distance produced by the algorithm design is quite good. The difference of 1 bit in the key, does not make the difference in the distance in each ciphertext from the two tables also differ by 1 bit. The results shown are quite varied and spread well. However, further testing needs to be done to ensure that the spread of distance that occurs is already good. This becomes a note in the next study to be able to test the plaintext-ciphertext difference with more key variations.

3.4. Avalanche effect test

Comparison of one-to-many scheme with several other cryptographs is done by observing the avalanche effect value in carrying out the encryption process. In [29] has tested several classical and modern cryptographic algorithms while this research continues its study with testing on AES and also one one-to-many scheme. Table 5 shows the result of the avalanche effect test for five encryption processes using the same key and plaintext. One-to-many scheme can produce different avalanche effect value for each encryption, different from other algorithms that always produce the same value.

Table 5. Result avalanche effect test

No	Algorithm	The avalanche effect (%)				
		Test-1	Test-2	Test-3	Test-4	Test-5
1	Caesar cipher	01.56	01.56	01.56	01.56	01.56
2	Playfair cipher	06.25	06.25	06.25	06.25	06.25
3	Vigenere cipher	03.13	03.13	03.13	03.13	03.13
4	One-to-many	54.29	47.79	49.26	51.47	52.94
5	S-box fleksibel	49.75	49.75	49.75	49.75	49.75
6	DES	55.68	55.68	55.68	55.68	55.68
7	AES	45.58	45.58	45.58	45.58	45.58

The data encryption standard (DES) algorithm has the highest avalanche effect value and the lowest Caesar cipher value. These results are reasonable because both DES and AES use all block cipher principles in compiling their algorithms, while Caesar cipher only uses transposition for each text input. The one-to-many scheme also does not use block cipher principle because the randomness of the generated keys use the logistical function is a determining factor in the encryption-decryption processes. On average, one-to-many scheme can produce an avalanche effect value of 52.20%. Although, it is still lower than DES, but it is still better than modern Blowfish cryptography by 25.46% and 6% more than AES, both of which are still widely used as information security algorithms.

3.5. Time requirement of cryptanalysis

Suppose an algorithm uses a key block length of 128 bits, the time needed to perform cryptanalysis is 2^{128} units. If a computer can crack 10^6 keys in one second, the time needed to crack the key is 5.4×10^{24} years. As the first step, the one-to-many scheme also uses 128 bits which its algorithm can produce several ciphertexts. In general, suppose that one-to-many scheme can produce different n-ciphertexts, the ciphertext search process requires $n \times 5.4 \times 10^{24}$ years. This result will certainly complicate the cryptanalysis. Although the possibility of performing the cryptanalysis still remains but it will take n-times process longer time than cracking ordinary cryptography.

4. CONCLUSION

The one-to-many encryption scheme produces five ciphertexts that have different and balanced visualization. The plaintext-ciphertext correlation test is also close to zero. This condition illustrates that the algorithm can make plaintext that is not statistically related to ciphertext. A plaintext input will produce several ciphertexts, and each of them has a variety of more than 50%. An average of 52.20% is obtained from the avalanche effect test, which is better than modern Blowfish cryptography by 25.46% and AES by 6%, both of which are still widely used as information security algorithms. This condition will be preferable because it makes difficult for cryptanalysts to find the relationship between plaintext and ciphertext directly by a process of trial and error. Many different forms of ciphertext produced by one-to-many scheme will make it very difficult for performing a cryptanalysis. Although the cryptanalysis process can still be done but it will require a longer process and time compared to algorithms that are solved ordinarily. If there are n ciphertexts, it will take n-times for time and process to find the plaintext.

ACKNOWLEDGEMENT

The authors are grateful to Satya Wacana Christian University Research and Community Service Center (BP3M) for the research funding support through the Internal Fundamental Research scheme in the 2017/2018 fiscal year.

REFERENCES

[1] A. Biryukov, J. Daemen, S. Lucks, and S. Vaudenay, "Topics and Research Directions for Symmetric Cryptography," *Early Symmetric Crypto Workshop*, 2017. [Online]. Available: https://orbilu.uni.lu/handle/10993/30953

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

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

[4] 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*, 2018, vol. 481, pp. 435-444, doi: 10.1007/978-981-13-2622-6_42.

[5] C. R. Dongarsane, D. Maheshkumar, and S. V. Sankpal , "Performance Analysis of AES Implementation on a Wireless Sensor Network," *Techno-Societal* , 2019, pp. 87-93, doi: 10.1007/978-3-030-16848-3_9.

[6] C. Ashokkumar, B. Roy, M. B. S. Venkatesh, and B. L. Menezes, "S-Box Implementation of AES Is Not Side Channel Resistant," *Journal of Hardware and Systems Security*, 2019. [Online]. Available: https://eprint.iacr.org/2018/1002.pdf

[7] 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*, vol. 43, no. 31, 2019, doi: 10.1007/s10916-018-1145-9.

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

[9] C. A. Sari, G. A. D. R. I. M. Setiadi, and E. H. Rachmawanto, "An improved security and message capacity using AES and Huffman Coding on Image Steganography," *TELKOMNIKA*, vol. 17, no. 5, pp. 2400-2409, 2019, doi: 10.12928/telkomnika.v17i5.9570.

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

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

[12] J. M. Espalmado and E. 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, 2017, doi: 10.11591/ijece.v7i2.pp1032-1041.

[13] E. R. Arboleda, J. L. Balaba, and J. C. 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, 2017, doi: 10.11591/eei.v6i3.627.

[14] 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, 2018, doi: 10.11591/ijece.v8i6.pp5144-5152.

[15] X. Bonnetain, M. Naya-Plasencia, and A. Schrottenloher, "Quantum Security Analysis of AES," *IACR Transactions on Symmetric Cryptology*, vol. 2019, pp 55-93, 2019, doi: 10.13154/tosc.v2019.i2.55-93.

[16] D. G´erault, P. Lafourcade, M. Minier, and C. Solnon, "Revisiting AES Related-Key Differential Attacks with Constraint Programming," *Cryptology*, vol. 139, 2017. [Online]. Available: https://eprint.iacr.org/2017/139.pdf

[17] A. Biryukov and D. Khovratovich , "Related-Key Cryptanalysis of the Full AES-192 and AES-256," *Advances in Cryptology - ASIACRYPT*, 2009, vol. 5912 , pp 1-18, doi: 10.1007/978-3-642-10366-7_1.

[18] J. Kim, S. Hong, and B. Preneel, "Related-key rectangle attacks on reduced AES-192 and AES-256," *International Conference on the Theory and Application of Cryptology and Information Security*, 2007, vol. 4593, pp 225–241, doo: 10.1007/978-3-540-74619-5_15.

[19] S. Lucks, "Ciphers secure against related-key attacks," *International Workshop on Fast Software Encryption*, 2004, vol. 3017, pp. 359–370, doi: 10.1007/978-3-540-25937-4_23.

[20] D. Deutsch and R. Jozsa, "Rapid Solution of Problems by Quantum Computation," *Proceedings The Royal Society London A, Mahtematical Physical & Engineering Sciences*, 1992, vol. 439, pp. 553–558, doi: 10.1098/rspa.1992.0167.

[21] R, P. Feynman, "Simulating Physics with Computers," *International Journal of Theoretical Physics*, vol. 21, pp. 467–488, 1982, doi: 10.1007/BF02650179.

[22] R. A. Perlner and D. A. Cooper, "Quantum resistant public key cryptography: a survey," *IDtrust '09: Proceedings of the 8th Symposium on Identity and Trust on the Internet*, 2009, doi: 10.1145/1527017.1527028.

[23] M. S. Henriques and N. K. Vernekar, "Using Symmetric and Asymmetric Cryptography to Secure Communication Between Devices in IOT," *International Conference on IoT and Applications* (ICIOT), Nagapattinam, India, 2017, pp. 1-4, doi: 10.1109/ICIOTA.2017.8073643.

[24] A. Hafsa, A. Sghaier, M. Zeghid, J. Malek, and M. Machhout, "An improved co-designed AES-ECC cryptosystem for secure data transmission," *International Journal of Information and Computer Security*, vol. 13, no. 1, pp. 118-140, 2020, doi: 10.1504/IJICS.2020.10028859.

[25] R. Riyaldhi, Rojali, and A. Kurniawan, "Improvement of Advanced Encryption Standard Algorithm with Shift Row and S.Box Modification Mapping in Mix Coloumn," *Discovery and innovation of computer science technology in artificial intelligence era: The 2nd International Conference on Computer Science and Computational Intelligence*, 2017, vol. 116, pp. 401-407, doi: 10.1016/j.procs.2017.10.079.

[26] G. Ribordy, J.-D. Gautier, N. Gisin, O. Guinnard, and H. Zbinden, "Fast and User-Friendly Quantum Key Distribution," *Journal of Modern Optics*, vol. 47, pp. 517-531, 2000, doi: 10.1080/09500340008244057.

[27] M. Aledhari, A. Marhoon, A. Hamad, and F. Saeed, "A New Cryptography Algorithm to Protect Cloud-based Healthcare Services," *Proceedings of the Second IEEE/ACM Interntional Confrence on Connected Health: Applications, Systems and Engeneering Technologies*, Philadelphia, PA, USA, 2017, pp. 37-43, doi: 10.1109/CHASE.2017.57.

[28] A. D. Wowor and V. B. Liwandouw, "Domain Examination of Chaos Logistics Function As A Key Generator in Cryptography," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 6, pp. 4577–4583, 2018, doi: 10.11591/ijece.v8i6.pp4577-4583.

[29] S. Ramanujan and M. Karuppiah, "Designing an Algorithm with High Avalanche Effect", *International Journal of Computer Science and Network Security*, vol. 11, no. 1, pp. 106-111, 2011. [Online]. Available: http://paper.ijcsns.org/07_book/201101/20110116.pdf

BIOGRAPHIES OF AUTHORS

Alz Danny Wowor is currently a lecturer at the Faculty of Information Technology, Satya Wacana Christian University in Salatiga, Indonesia. He received bachelor and master degree in mathematics and informatics from Satya Wacana Christian University, in 2005 and 2011 respectively. His researches are in fields of Primitive Cryptography, Symmetric Cryptography: Block Cipher and Pseudorandom. He can be contacted at email: alzdanny.wowor@uksw.edu.

Bambang Susanto He is currently a lecturer at the Department of Mathematics, Faculty of science and mathematics, Satya Wacana Christian University in Salatiga, Indonesia. His obtained Bachelor Degree in Mathematics from Diponegoro University (Indonesia) in 1988. His received master and doctor degree in mathematics from Bandung Institute of Technology, in 1992 and 2005 respectively. His researches are in fields of Data Mining, Big Data Analysis, Business Intelligence, Risk Analysis, Primitive Cryptography, Symmetric Cryptography. He can be contacted at email: bambang.susanto@uksw.edu.