STATISTICAL ANALYSIS OF 3D RECTANGLE ENCRYPTION ALGORITHM

[1]ABDUL ALIF ZAKARIA, [2]A. H. AZNI, [3]FARIDA RIDZUAN, [4]NUR HAFIZA ZAKARIA, [5]MASLINA DAUD

[1,2,3,4]Faculty of Science and Technology, Universiti Sains Islam Malaysia, Nilai 71800, Malaysia,
[1,5]CyberSecurity Malaysia, Menara Cyber Axis, Cyberjaya 63000, Malaysia,
[2,3,4]CyberSecurity and System Research Unit, Faculty of Science and Technology, Universiti Sains Islam Malaysia, Nilai 71800, Malaysia
Email: [1]alif@cybersecurity.my, [2]ahazni@usim.edu.my

Abstract - The statistical analysis of the 3D RECTANGLE to test the randomness of the lightweight block cipher is presented in this paper. Lightweight block ciphers use less computing power than conventional algorithms, making them more suitable for use in low-resource devices. Randomness property is important for an encryption algorithm to ensure that the output does not contain any message pattern. The NIST Statistical Test Suite is used to perform the randomness tests. Nine data categories of block cipher rare used to produce 1,000 cipher text samples from the algorithm. From the conducted testing, the 3D RECTANGLE passed 88.89% of the randomness tests. Based on the 1% significance level, the analysis indicates that 3D RECTANGLE appears to be non-random. The experimental results reveal the weakness of the algorithm that can be addressed in future studies.

Keywords - Randomness, 3D RECTANGLE, Lightweight Block cipher, Statistical Analysis, Encryption.

I. INTRODUCTION

The IoT collects real-time data from devices such as radio frequency identification technology, laser scanners, sensors, and global positioning systems that pose major security concerns [1]. As a result, lightweight block ciphers gain popularity because of the security provided with lesser computing resources[2]. Hence, there is a need for continuous research and development of lightweight algorithms. Since 2011, many algorithms have been developed such as NVLC [3],RARE [4],DoT [5],LCA [6], ILEA [7], and TED [8].

3D RECTANGLE is a lightweight algorithm invented for the usage of IoT devices [9].The algorithm was developed to improve the cryptographic strength of the original RECTANGLE block cipher. RECTANGLE is well known for its minimal hardware costs as well as its outstanding software efficiency [10]. Although RECTANGLE lightweight block cipher is highly efficient, the security of the algorithm requires further investigation. One of the proposed solutions to improve RECTANGLE security strength is by adopting the 3D bit rotation method which lead to the development of the 3D RECTANGLE algorithm.

Randomness test is a procedure used to determine the minimum security requirement of an encryption algorithm [11]. The statistical analysis may determine if the assessed algorithm satisfies the security requirement. A non-random encryption algorithm map ears to be susceptible to cryptographic attacks [12]. Since an unauthorized user should not be able to predict the cryptographic sequences any simpler than a brute force attack, an encryptional gorithm must be able to perform as a pseudorandom number generator to produce random outputs[13]. The NIST Statistical Test Suite has been widely adopted to evaluate the randomness of encryption algorithms such as AES, Serpent, Two fish, RC6, and MARS [14]. Therefore, it is important for the 3D RECTANGLE to be evaluated using the same method.

The organization of this paper is arranged as follows. In Section II, details of the 3D RECTANGLE lightweight algorithm construction are presented. Next, the randomness tests are described in Section III. Section IV discusses the results of the statistical analysis of 3D RECTANGLE. Lastly, Section V summarizes the conclusion.

II. 3D RECTANGLE LIGHTWEIGHT BLOCK CIPHER

The 3D RECTANGLE algorithm structure consists of a 64-bitcipher block and a 128-bit key. 3D RECTANGLE encryption operation is carried out in 25 rounds utilizing a 3D bit rotation technique. The construction of 3D RECTANGLE is built on encryption and key schedule algorithms that are discussed in the following sections.

A. ENCRYPTION ALGORITHM

The 3D RECTANGLE cipher state is presented in the form of 4 x 4 x 4 matrices which is also known as a cube. Let $W = w_{63}||\ldots||w_1||w_0$illustrate the cipher state where by the initial 16 bits, $w_{15}||\ldots||w_1||w_0$ are positioned in $Slice_0$ and the consecutive 16 bits $w_{31}||\ldots||w_{17}||w_{16}$ are ordered in $Slice_1$.Thesubsequent 16 bits are designated as$Slice_2$ and $Slice_3$ as shown in Figure 1.

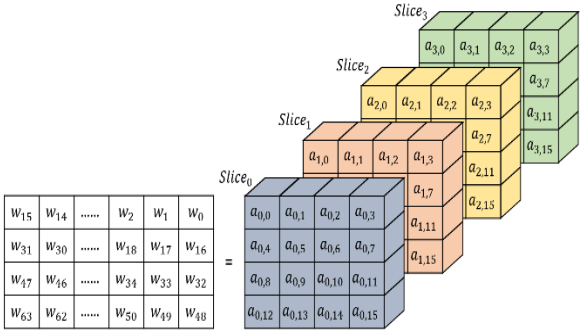

Figure 1: 3D Cipher State

Every encryption round consists of four processes namely AddRoundKey,3DBitRotation,SubColumn, and Shift Row. An additional Add Round Key is required at the end of the final round. The round transformation of the 3D RECTANGLE block cipher is shown in Algorithm 1.

Algorithm 1: Pseudocode of 3D RECTANGLE (encryption)	
1:	*RoundKeysGeneration* (*Key*)
2:	for i = 0 to 24 do
3:	*AddRoundKey* (*State*, K_i)
4:	*SubColumn* (*State*)
5:	*ShiftRow* (*State*)
6:	end for
7:	*AddRoundKey* (*State*, K_{25})

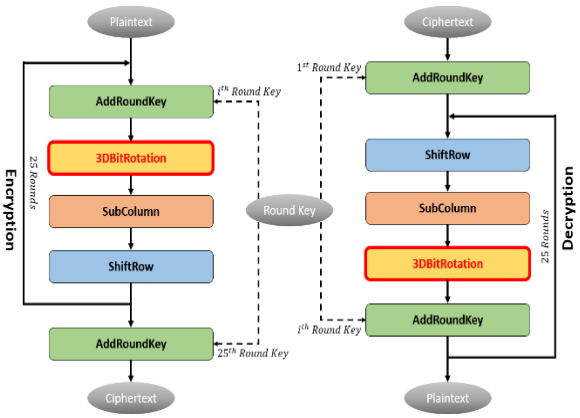

Figure 2: 3D RECTANGLE Process

Figure 2 depicts the overall structure of the 3D RECTANGLE.

The encryption process is outlined below:
1. *Add Round Key*: The cipher state (*a*) and the round key (*K*)are bitwise XO Red.

2. *3DBitRotation*:The cipher state is rotated in the following clockwise direction:
a) $Slice_0$: Rotate 0^o (no rotation).
b) $Slice_1$: Rotate 90^o.
c) $Slice_2$: Rotate 180^o.
d) $Slice_3$: Rotate 270^o.

Figure 3 depicts each of the cipher state *Slice* before and after applying the 3DBitRotation operation.

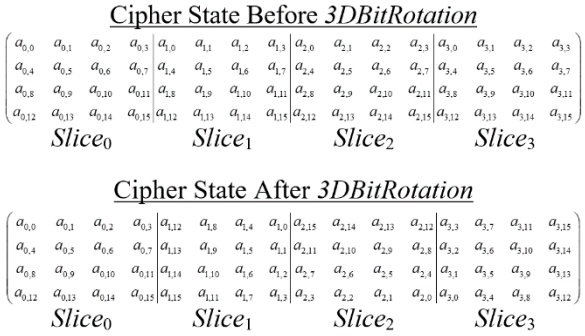

Figure 3: 3D Bit Rotation Cipher State

3.*SubColumn*: The column is substituted by implementing an S-box as displayed in Table 1. Input of the S-box is defined as $Col_j = a_{3,j}||a_{2,j}||a_{1,j}||a_{0,j}$for $0 \le j \le 15$, and $S(Col_j) = b_{3,j}||b_{2,j}||b_{1,j}||b_{0,j}$represents the out put as shown in Figure 4.

x	0	1	2	3	4	5	6	7
$S(x)$	6	5	C	A	1	E	7	9
x	8	9	A	B	C	D	E	F
$S(x)$	B	0	3	D	8	F	4	2

Table 1: S-box

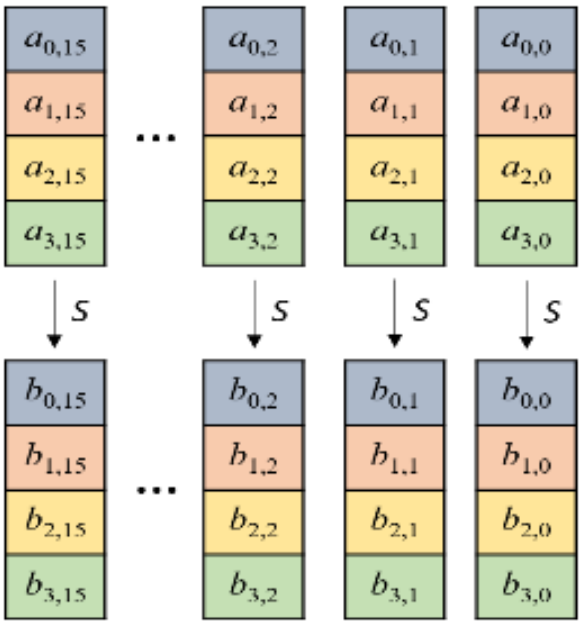

Figure 4:SubColumn

4. *Shift Row*: Each row of the cipher state is left rotated at a certain position. $Row_0$is unchanged,meanwhile,Row_1, Row_2, and Row_3 are left rotated by1, 12, and 13 bits as shown in Figure 5.

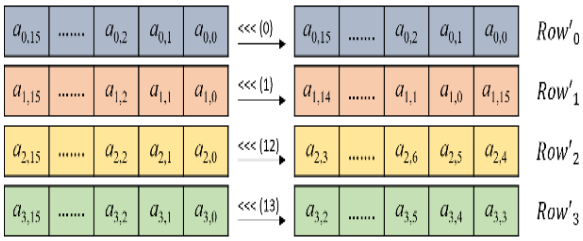

Figure 5: ShiftRow

B. KEY SCHEDULE ALGORITHM

Round keys are generated from the key schedule algorithm that will be used in the encryption process. Let $V = v_{127}||\ldots||v_1||v_0$represent the encryption key. Every 32 bits are grouped to obtain $RowKey_3$, $RowKey_2$, $RowKey_1$, and $RowKey_0$as depicted in Figure 6.

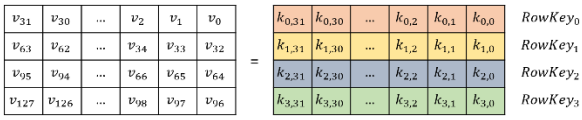

Figure 6: Key State

16 rightmost columns of every *Row Key* are grouped to form the 64-bit i^{th} round key $K_i$for $0 \le i \le 24$. After the K_i extraction is completed, the round keys values in every round are updated using the below processes:

1.*KeySubColumn*:Four uppermost rows and eight rightmost columns are reformed by applying the S-box in Table 1, i.e., $k'_{3,j}||k'_{2,j}||k'_{1,j}||k'_{0,j} = S(k'_{3,j}||k'_{2,j}||k'_{1,j}||k'_{0,j})$, for $0 \le j \le 7$as shownin Figure 7.

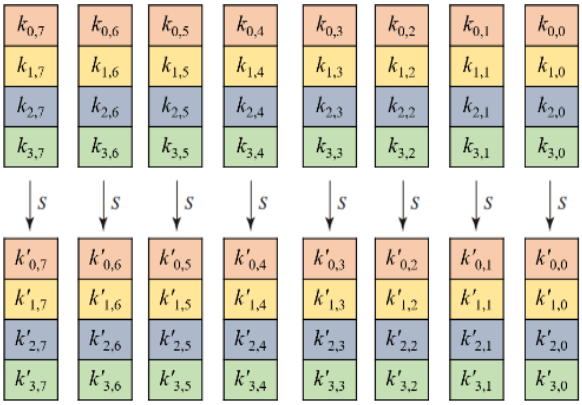

Figure 7: KeySubColumn

2.*Feistel Transformation*: 1-round Feistel transformation operation is implemented, i.e., $RowKey_0' = (RowKey_0 <<< 8) \oplus RowKey_1$, $RowKey_1' = RowKey_2$, $RowKey_2' = (RowKey_2 <<< 16) \oplus RowKey_3$, and $RowKey_3' = RowKey_0$as shown in Figure 8.

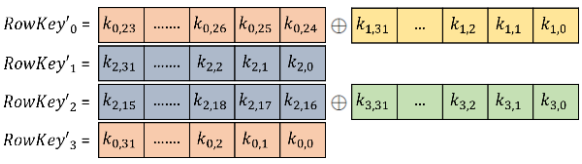

Figure 8: Feistel Transformation

3.*RoundConstants*: Rc_i is a 5-bits round constant produced by a Linear Feedback Shift Register as illustrated in Table 2. In every round, 5-bits (rc_4, rc_3, rc_2, rc_1, rc_0)of the round constant are one-bit left rotated andthe new value rc_0 is constructed by $rc_4 \oplus rc_2$. The 5-bit key state is then XORed with Rc_i, i.e., $(k'_{4,0}||k'_{3,0}||k'_{2,0}||k'_{1,0}||k'_{0,0}) = (k_{4,0}||k_{3,0}||k_{2,0}||k_{1,0}||k_{0,0}) \oplus Rc_i$. Finally, the revised key state produced K_{25}.

i	0	1	2	3	4	5	6	7	8	9	10	11	12
Rc_i	01	02	04	09	12	05	0B	16	0C	19	13	07	0F
i	13	14	15	16	17	18	19	20	21	22	23	24	
Rc_i	1F	1E	1C	18	11	03	06	0D	1B	17	0E	1D	

Table 2: Round Constants

III. RANDOMNESS TEST

Statistical analysis of the 3D RECTANGLE algorithm is conducted by employing the NIST Statistical Suite that is inclusive of 15 individual tests with multiple input parameters[15]. The statistical package emphasizes several features of non-randomness that may emerge in block cipher outputs.

Non-parameterized test selection includes eight statistical tests such as Random Excursion Variant (18 p-values), Random Excursion (8 p-values), Cumulative Sums (2 p-values), Binary Matrix Rank (1 p-value), Frequency (1 p-value), Longest Runs of Ones (1 p-value), Spectral DFT (1 p-value), and Runs (1 p-value).The remaining seven tests namely Non-Overlapping (148 p-values), Serial (2 p-values), Approximate Entropy (1 p-value), Linear Complexity (1 p-value), Overlapping Templates (1 p-value), Maurers Universal (1 p-value), and Block Frequency (1 p-value)are classified as parameterized tests which permits parameter values to be entered.

In order to assess the randomness of an encryption algorithm, a significance level, α needs to be set to at least 0.1% (0.001) but not bigger than 1% (0.01). Meanwhile, the sample size, *s* must be at least the inverse of α (i.e., 1 ÷ 0.001 = 1,000 samples).If the *p*-value ≥ α, the sample is considered to be random with a confidence level of 99.9% [16].In contrast, if the *p*-value < α, the sample is regarded as non-random.
The proportion of the testing samples determines the randomness of a block cipher which is defined as follows:

$$p_\alpha = (1-\alpha) - 3\sqrt{\frac{\alpha(1-\alpha)}{s}}$$

where *s*is equal to1,000 testing samples and α is equal to 0.01.If the number of rejected samples exceeds the proportion p_α, the testing sample is not random.

According to Table 3, nine data categories are adopted to generate the input data in plaintext and key formats for the block cipher. Every data category produced a unique set of 1,000 testing samples. The key and block sizes determine the block number obtained from each sample [17].The derived blocks combined the cipher texts to generate large bit sequences for the statistical analysis.

Data Category	Plaintext	Key	Derived Blocks	Derived Bits
Random Plaintext/ Random Key (RPRK)	15,625 random 64-bit plaintext	1 random 128-bit key	15,625	1,000,000
Strict Plaintext Avalanche (SPA)	245 random 64-bit plaintext	All zero	15,680	1,003,520
Strict Key Avalanche (SKA)	All zero	123 random 128-bit keys	15,744	1,007,616
Plaintext/ Ciphertext Correlation (PCC)	15,625 random 64-bit plaintext	1 random 128-bit key	15,625	1,000,000
Ciphertext Block Chaining Mode (CBCM)	All zero	1 random 128-bit key	15,625	1,000,000
Low-Density Key (LDK)	8,257 random 64-bit plaintext	3,241 specific 128-bit keys	8,257	528,448
High-Density Key (HDK)	8,257 random 64-bit plaintext	3,241 specific 128-bit keys	8,257	528,448
Low-Density Plaintext (LDP)	2,081 random 64-bit plaintext	2,081 specific 128-bit keys	2,081	133,184
High-Density Plaintext (HDP)	2,081 random 64-bit plaintext	2,081 specific 128-bit keys	2,081	133,184

Table 3: Data Categories

IV. EXPERIMENTAL RESULTS

The NIST provided the recommendation for the number of input bits to conduct the randomness analysis (Rukhin et al., 2001). A minimum of 100 bits of input are required for Cumulative Sums, Runs, Block Frequency, and Frequency tests. For Overlapping Templates, Linear Complexity, Random Excursion Variant, and Random Excursion tests, at least one million bits are required for the experiment. Meanwhile, Longest Runs of Runs, Spectral DFT, Binary Matrix Rank, and Maurers Universal tests require a minimum of 128, 1,000, 38,912, and 387,480 bits correspondingly. On the other hand, Non-Overlapping Templates, Serial, and Approximate Entropy tests do not have specific input bits requirements.

Based on the input, each data category generated several ciphertext lengths as shown in Table 3.The 15 statistical tests may be used to examine data categories such as PCC, SKA, CBC, SPA, and RPRK [18]. Due to insufficient data length, only 11 tests for HDK and LDK are conducted, while 10 tests for HDP and LDP are executed.

To evaluate if a sample passes or fails a randomness test, an acceptable rejection range is required. The sample passes the test if the rejected sequences are inside the given range. Conversely, the test fails if the rejected sequences are outside of the range. The assessed samples for the Random Excursion Variant and Random Excursion tests are fewer than 1,000 samples because of an insufficient total number of cycles produced by the algorithm as indicated in Tables 4, 5, and 6.The N/A notation indicates that the statistical analysis is not permitted to be performed due to an insufficient cipher text output.

	Statistical Test	Data Category		
		RPRK	PCC	CBC
		Range of Acceptable Rejection: [0, 20]		
1	Cumulative Sums	999/1000	984/1000	986/1000
2	Longest Runs of Ones	985/1000	996/1000	999/1000
3	Binary Matrix Rank	987/1000	989/1000	984/1000
4	Spectral DFT	985/1000	990/1000	994/1000
5	Runs	990/1000	988/1000	993/1000
6	Serial	985/1000	981/1000	998/1000
7	Approximate Entropy	995/1000	991/1000	983/1000
8	Block Frequency	981/1000	997/1000	995/1000
9	Frequency	993/1000	998/1000	986/1000
10	Non-Overlapping Templates	993/1000	981/1000	997/1000
11	Maurer's Universal	992/1000	997/1000	990/1000
12	Linear Complexity	991/1000	996/1000	988/1000
13	Overlapping Template	993/1000	989/1000	987/1000
		Range of Acceptable Rejection: [0, 14]		
14	Random Excursion Variant	589/595	612/617	580/584
15	Random Excursion	587/595	609/617	577/584

Table 4: Randomness Test Results (RPRK, PCC, and CBC)

	Statistical Test	Data Category		
		SPA	SKA	LDK
		Range of Acceptable Rejection: [0, 20]		
1	Cumulative Sums	999/1000	979/1000*	992/1000
2	Longest Runs of Ones	982/1000	990/1000	993/1000
3	Binary Matrix Rank	989/1000	995/1000	981/1000
4	Spectral DFT	998/1000	993/1000	984/1000
5	Runs	982/1000	986/1000	988/1000
6	Serial	985/1000	994/1000	982/1000
7	Approximate Entropy	987/1000	972/1000*	985/1000
8	Block Frequency	992/1000	977/1000*	997/1000
9	Frequency	995/1000	991/1000	983/1000
10	Non-Overlapping Templates	986/1000	968/1000*	987/1000
11	Maurer's Universal	981/1000	978/1000*	986/1000
12	Linear Complexity	990/1000	996/1000	991/1000
13	Overlapping Template	995/1000	988/1000	989/1000
		Range of Acceptable Rejection: [0, 14]		
14	Random Excursion Variant	625/633	619/622	N/A
15	Random Excursion	621/633	613/622	N/A

Table 5: Randomness Test Results (SPA, SKA, and LDK)
* indicates fail statistical analysis

	Statistical Test	Data Category		
		HDK	HDP	LDP
		Range of Acceptable Rejection: [0, 20]		
1	Cumulative Sums	993/1000	991/1000	983/1000
2	Longest Runs of Ones	986/1000	997/1000	996/1000
3	Binary Matrix Rank	998/1000	990/1000	999/1000
4	Spectral DFT	999/1000	988/1000	996/1000
5	Runs	987/1000	990/1000	992/1000
6	Serial	982/1000	981/1000	992/1000
7	Approximate Entropy	985/1000	989/1000	998/1000
8	Block Frequency	986/1000	983/1000	982/1000
9	Frequency	993/1000	997/1000	994/1000
10	Non-Overlapping Templates	996/1000	995/1000	982/1000
11	Maurer's Universal	988/1000	983/1000	994/1000
12	Linear Complexity	984/1000	990/1000	985/1000
13	Overlapping Template	999/1000	997/1000	998/1000
		Range of Acceptable Rejection: N/A		
14	Random Excursion Variant	N/A	N/A	N/A
15	Random Excursion	N/A	N/A	N/A

Table 6: Randomness Test Results (HDK, HDP, and LDP)

In general, the 3D RECTANGLE passed 10 out of 15 statistical tests in total. The block cipher failed Block Frequency, Approximate Entropy, Cumulative Sums, Non-Overlapping Templates, and Maurer's Universal. The results as shown in Tables4, 5, and 6 suggest that the 3D RECTANGLE algorithm does not pass all of the statistical analysis.

In terms of data category, the 3D RECTANGLE passed eight data categories including CBC, PCC, RPRK, SPA, LDK, HDK, LDP, and HDP. 3D RECTANGLE only failed the SKA data category. The experimental result of the SKA is impacted by the sensitivity of the block cipher against alterations of the encryption key. The results demonstrate that the weakness of the 3D RECTANGLE key schedule algorithm contributes to the lack of randomization of the block cipher output.

A comparison of randomness test results against the original RECTANGLE[19]is displayed in Table 7. Even though both RECTANGLE variants do not manage to pass all 15statistical tests, the 3D RECTANGLE achieved better randomness than the original RECTANGLE design.

Algorithm	Results	Statistical Test	Data Category
3D RECTANGLE	Pass	10	8
	Fail	5	1
RECTANGLE	Pass	10	6
	Fail	5	3

Table 7: Comparison of Randomness Test Results

Overall, based on the 1% significance level, the cipher text output of the 3D RECTANGLE block cipher is not random. According to the results, the 3D RECTANGLE managed to pass 88.89% out of the 9 data categories and 66.67% out of the 15 statistical tests. From the finding, it is suggested to improve the design of the 3D RECTANGLE key schedule algorithm since it is found to be the main weakness of the block cipher.

V. CONCLUSION

The capability of a block cipher to operate as a pseudorandom number generator is an essential design principle. The NIST Statistical Test Suite is capable of evaluating a block cipher's randomness criterion. The randomness of the 3D RECTANGLE

lightweight block cipher was tested using 1,000 unique samples. The findings indicate that the lightweight block cipher is non-random at the 1% significance level. The fact that an encryption algorithm passes all of the statistical analysis does not ensure its security strength. A secure lightweight block cipher, on the other hand, should pass all randomness tests to meet the security requirement of an encryption algorithm. In future research, modifications of the 3D RECTANGLE block cipher design are suggested to improve the security strength of the algorithm.

REFERENCES

[1] W. K. Ahmed and R. S.Mohammed, "Lightweight authentication methods in IoT: Survey," in Proc. International Conference on Computer Science and Software Engineering., IEEE, 2022, pp. 241-246.

[2] I.N. M. Shah and E. S.Ismail, "Randomness analysis on lightweight block cipher, PRESENT,"J. Comput. Sci., vol. 16, pp. 1639-1647, 2020.

[3] S. Q. A. Al-Rahman, A. M.Sagheer, and O. A. Dawood, "NVLC: New variant lightweight cryptography algorithm for internet of things," in Proc. Annual International Conference on Information and Sciences., IEEE, 2018, pp. 176-181.

[4] T. Omrani, R. Becheikh, O. Mannai, R. Rhouma, and S. Belghith, "RARE: A robust algorithm for rapid encryption," in Proc. International Conference for Internet Technology and Secured Transactions., IEEE, 2018, pp. 23-28.

[5] J. Patil, G. Bansod, andK. S.Kant, "DoT: A new ultra-lightweight SP network encryption design for resource-constrained environment," in Proc.International Conference on Data Engineering and Communication Technology, Advances in Intelligent Systems and Computing., Springer, Singapore, 2019,pp. 249-257.

[6] B. Aboshosha, M. Dessouky, R. Ramadan, and A. El-Sayed, "LCA-Lightweight cryptographic algorithm for IoT constraint resources," in Proc. International Conference on Electronic Engineering., 2019, pp. 374-380.

[7] P. Jha, H. Y.Zorkta, D. Allawi, and M. R.Al-Nakkar, "Improved lightweight encryption algorithm (ILEA)," in Proc. International Conference for Emerging Technology., IEEE, 2020, pp. 1-4.

[8] C. Thorat, V. Inamdar, and B. Jadhav, "TED: A lightweight block cipher for IoT devices with side-channel attack resistance," Int. J. Inf. Technol. Secur., vol. 12, pp. 83–96, 2020.

[9] A.A.Zakaria, A. H.Azni, F. Ridzuan, N. H.Zakaria, and M. Daud, "Extended RECTANGLE algorithm using 3D bit rotation to propose a new lightweight block cipher for IoT,"IEEE Access, vol. 8, pp. 198646–198658, 2020.

[10] A.Senol, "Improved differential attacks on RECTANGLE," Master's Thesis. Middle East Technical University, 2017.

[11] S. Ariffin and N. A. M.Yusof, "Randomness analysis on 3D-AES block cipher," in Proc.International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery., IEEE, 2017, pp. 331–335.

[12] N. A. M.Yusof, "Statistical analysis on enhanced 3D-AES block cipher cryptographic algorithm," Master's Thesis. Universiti Teknologi MARA, 2021.

[13] L. C. N.Chew, I. N. M.Shah, N. A. N.Abdullah, N. H. A.Zawawi, H. A.Rani, and A. A.Zakaria, "Randomness analysis on SPECK family of lightweight block cipher," Int. J. Cryptol. Res., 2015, vol. 5, pp. 44-60.

[14] M. Aljohani, I. Ahmad, M. Basheri, andM. O.Alassafi, "Performance analysis of cryptographic pseudorandom number generators," IEEE Access., vol. 7, pp. 39794–39805, 2019.

[15] A.Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, and S. Vo,A statistical test suite for random and pseudorandom number generators for cryptographic applications, NIST Special Publication 800-22 Revision 1a, 2001.

[16] E. Simion and P. Burciu, "A note on the correlations between NIST cryptographic statistical tests suite,"UPB Sci. Bull. Ser. A Appl. Math. Phys., vol. 81, pp. 209–218, 2019.

[17] N. A. N. Abdullah, L. C. N.Chew,A. A.Zakaria, K. Seman and N. M.Norwawi, "The comparative study of randomness analysis between modified version of LBlock block cipher and its original design,"Int. J. Comput. Inf. Technol., vol. 4, pp. 867–875, 2015.

[18] M. Imdad,S.N.Ramli, and H. Mahdin, "An enhanced key schedule algorithm of PRESENT-128 block cipher for random and non-random secret keys," Symmetry., vol. 14, pp. 1-22.

[19] A.A.Zakaria, A. H.Azni, F. Ridzuan, N. H.Zakaria, and M. Daud, "Randomness analysis on RECTANGLE block cipher," in Proc. Cryptology and Information Security Conference., 2020. pp. 133–142.

★★★