International Journal of Computing Sciences Research (ISSN print: 2546-0552; ISSN online: 2546-115X)
Vol. 8, pp. 3187-3201
doi: 10.25147/ijcsr.2017.001.1.213
https://stepacademic.net

Short Paper*

Enhancing the RECTANGLE Algorithm for Text Data Encryption in Mobile Applications Utilizing BBSXOR, NonceXOR, and Modulo Addition

Karylle Andrei C. Velarde
College of Information Systems and Technology Management, Pamantasan ng Lungsod ng Maynila, Philippines
kacvelarde2020@plm.edu.ph

Raphael P. Enciso
College of Information Systems and Technology Management, Pamantasan ng Lungsod ng Maynila, Philippines
rpenciso2020@plm.edu.ph
(corresponding author)

Vivien A. Agustin
College of Information Systems and Technology Management, Pamantasan ng Lungsod ng Maynila, Philippines
vaagustin@plm.edu.ph

Jonathan C. Morano
College of Information Systems and Technology Management, Pamantasan ng Lungsod ng Maynila, Philippines
jcmorano@plm.edu.ph

Leisyl M. Mahusay
College of Information Systems and Technology Management, Pamantasan ng Lungsod ng Maynila, Philippines
lmocampo@plm.edu.ph

Jamillah S. Guialil
College of Information Systems and Technology Management, Pamantasan ng Lungsod ng Maynila, Philippines
jsguialil@plm.edu.ph

Date received: May 7, 2024
Date received in revised form: July 2, 2024; July 8, 2024; July 27, 2024
Date accepted: July 30, 2024

Recommended citation:

Velarde, K. A. C., Enciso, R. P., Agustin, V. A., Morano, J. C., Mahusay, L. M., & Guialil, J. S. (2024). Enhancing the RECTANGLE Algorithm for Text Data Encryption in Mobile Application Utilizing BBSXOR, NonceXOR, and ModuloAddition. *International Journal of Computing Sciences Research*, *8*, 3187-3201. https://doi.org/10.25147/ijcsr.2017.001.1.213

**Special Issue on International Research Conference on Computer Engineering and Technology Education (IRCCETE). Guest Associate Editors: Dr. Roben A. Juanatas (National University-Manila) and Dr. Nelson C. Rodelas (University of East).*

Abstract

Purpose –This study proposes enhancing the RECTANGLE algorithm by addressing its problems and security by modifying its key schedule and encryption algorithm.

Method –To enhance the Key Schedule Algorithm, the researchers have chosen a CSPRNG, the BlumBlumShub. For the Encryption Algorithm, researchers chose to incorporate ModuloAddition and NonceXOR into the algorithm.

Results –The proposed enhancement was compared with existing algorithms across three tests: Frequency, Correlation Coefficient, and Avalanche Effect. The Proposed Enhancement excelled, achieving a 96% success rate in the Frequency Test, fewer moderate positive relationships in the Correlation Coefficient, and surpassing the 50% threshold in the Avalanche Effect Test with a 55% avalanche effect result.

Conclusion – This study significantly enhances the RECTANGLE algorithm through targeted modifications on its key schedule and encryption algorithm. Tests and analysis show that it effectively addresses known weaknesses, resulting in improved security properties.

Recommendations – Further work on refining the modifications made to the algorithm is recommended to mitigate any potential performance degradation observed in the study.

Research Implications – The enhanced RECTANGLE Algorithm showcases notable advancements in randomness, key generation, and confusion properties, enhancing its security and efficacy in encryption tasks. Despite minor performance impacts from modifications, its overall robustness in lightweight block cipher algorithms highlights the importance of continual refinement to meet evolving security standards.

Keywords – RECTANGLE, encryption, spn, block cipher, lightweight

INTRODUCTION

Security measures are essential to protect data flows between smartphone users, as alteration and interception can compromise availability, integrity, and confidentiality. Therefore, encryption is crucial (Dhanda et al., 2020). Many mobile encryption apps use techniques like AES, but these methods require significant memory and power, making them impractical for resource-constrained applications (Salunke et al., 2019).

Recently, lightweight block cipher algorithms have emerged to offer optimal security and efficiency for limited-resource devices. Among these, RECTANGLE is notable for its competitive encryption speed (Zhang et al., 2015). However, RECTANGLE's key schedule algorithm is weakened by non-robust round key generation (Naser & Naif, 2022). The security of the encryption algorithm relies on the key schedule algorithm's effectiveness (Afzal et al., 2020). Effective encryption algorithms must spread plaintext alterations throughout the ciphertext, a property known as confusion (Zakaria et al., 2021). Yan et al. (2019) found that poor distribution of RECTANGLE key bits in the diffusion path increases vulnerability to cryptanalytic attacks. Thus, refining RECTANGLE's key schedule algorithm is crucial for improving its confusion, diffusion properties, and robust key generation, thereby strengthening overall security (Zakaria et al., 2021).

This research aims to enhance the RECTANGLE block cipher by improving its key schedule algorithm with a CSPRNG function for robust round keys, incorporating a nonce function for better confusion, and using modulo addition to enhance key bit distribution along the diffusion path.

LITERATURE REVIEW

RECTANGLE Block Cipher

RECTANGLE algorithm is a lightweight block cipher algorithm that has a block size of 64 bits and a key size of 80 or 128 bits (Zhang et al., 2015). The RECTANGLE Algorithm comprises two parts: the key schedule algorithm, which generates the round keys, and the encryption algorithm, which encrypts the plaintext using the round keys.

Encryption Algorithm

The RECTANGLE design adopts an SPN (Substitution-Permutation Network) structure for encrypting data, with 25 rounds (Omrani et al., 2018). Each round encompasses three processes: AddRoundKey, SubColumn, and ShiftRow

1. AddRoundKey: This step involves performing an XOR operation between the current cipher state and the round key state.
2. SubColumn: In this operation, a column substitution is conducted using an S-box,

depicted in Figure 1. The S-box utilized functions as a 4-bit to 4-bit S-box.

3. ShiftRow: In this process, every row is rotated to a particular position. While Row 0 remains unaffected, Row 1 is shifted by 1, Row 2 by 12, and Row 3 by 13 bits.

Key Schedule Algorithm

The Key Schedule Algorithm consists of 3 processes mainly KeySubColumn, FeistelTransformation, and RoundConstants.

1. KeySubColumn – The reconstruction of the 4 uppermost rows and 8 rightmost columns is accomplished using the S-box, as illustrated in Figure 1.
2. FeistelTransformation - A Feistel Transformation is applied on the following rows:

   $RowKey'0 = (RowKey0 <<< 8) \oplus RowKey1$, $RowKey'1 = RowKey2$,

   $RowKey'2 = (RowKey2 <<< 16) \oplus RowKey3$, and $RowKey'3 = RowKey0$
3. RoundConstants- These are 5-bit round constants that are XORed with the key state.

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

Figure 1. S-BOX Conversion Table.

Previous Modifications on the RECTANGLE Block Cipher

Extended RECTANGLE Algorithm Using 3D Bit Rotation

The RECTANGLE lightweight block cipher is found to lack essential cryptographic security properties such as confusion and diffusion. Zhang et al. (2015) observed that RECTANGLE lacks confusion, a vital security aspect, while Yan et al. (2019) identified poor key bit distribution along the diffusion path, making it vulnerable to cryptanalytic attacks.

To improve RECTANGLE's security, a 3D cipher design is proposed, inspired by AES (Nakahara, 2016). The 3DBitRotation function is introduced for the permutation operation that significantly improves confusion and diffusion. Experimental findings show that the Extended 3D RECTANGLE outperforms the original RECTANGLE in security and strength, effectively enhancing its confusion and diffusion (Zakaria et al., 2020).

Modifications of Key Schedule Algorithm on RECTANGLE Block Cipher

The RECTANGLE block cipher is highly efficient in encryption speed compared to other lightweight algorithms, but it struggles with generating robust round keys. A key schedule algorithm must produce round keys with qualities of randomization and confusion (Zakaria et al., 2021). Yan et al. (2019) noted that poor key bit distribution on the diffusion path

makes RECTANGLE vulnerable to cryptographic attacks. Enhancing the key schedule algorithm (KSA) can improve the confusion property. To address this vulnerability, three approaches were developed: adjustments to the Feistel Transformation, Key Sub-Column, and Round Constants functions (Zakaria et al., 2021).

METHODOLOGY

In this study, the researchers first analyzed the problems of the RECTANGLE Algorithm which are the non-robust key generation, and the lack of confusion and diffusion property. The strength and effectiveness of an encryption algorithm heavily rely on the robustness of its encryption key (Afzal et al., 2020). Therefore, to enhance the Key Schedule Algorithm, the researchers have chosen a CSPRNG (Cryptographically Secure Pseudorandom Number Generator), the BlumBlumShub, which is the most preferable algorithm for key generation (Divyanjali et al., 2014). Considering the importance of Confusion and Diffusion properties in encryption algorithms, the researchers chose to incorporate ModuloAddition (De Los Reyes et al., 2018) and NonceXOR (Zakaria, 2022) to further improve these properties.

BlumBlumShubXOR

The Blum Blum Shub (BBS) is renowned for its provable security, linked to the complexity of the NP-complete quadratic residue problem. Breaking the BBS algorithm is equivalent to solving this problem, making BBS highly desirable for cryptographic applications, particularly in key generation (Divyanjali et al., 2014).

The process of Blum Blum Shub XOR involves initializing the algorithm by selecting two large prime numbers, p and q, both congruent to 3 modulo 4, and computing their product n = p * q as the modulus. A seed value, X_0, which is relatively prime to N, is chosen to begin the generation process. The main formula is shown in Equation 1:

$$X_{i+1} = (X_i)^2 \bmod N \qquad \textit{Equation 1}$$

A sequence of pseudorandom bits is generated by iteratively squaring the current value xi modulo n to produce the next value xi+1. From each xi+1, a single pseudorandom bit is extracted. BlumBlumShubXOR is a function that has 2 fragmented functions inside, first, it generates random bits, and XOR the KeyState to this generated random bit.

Figure 2 shows the process the key undergoes on the BlumBlumShubXOR function. The Blum Blum Shub Pseudo-Random Number Generator accepts 4 parameters which are p, q, seed, and random bit length which is the output as discussed previously. A round constant Rcp[i] and Rcq[i] will be used to store the value for p and q on rounds 1, 2, 4, 8, and 16 as shown in Table 1. The length will be equal to the number of bits based on the main key.

Table 1. Round Constants

Rounds	1	2	4	8	16
Rcp	0fcf	9b7f	9ed7	b98f	b0bf
Rcq	be07	0fcf	44e3	aadb	9647

In rounds 1, 2, 4, 8, and 16 the Blum Blum Shub CSPRNG will generate random 128 bits that will be used to XOR with the current KeyState as shown in Figure 2.

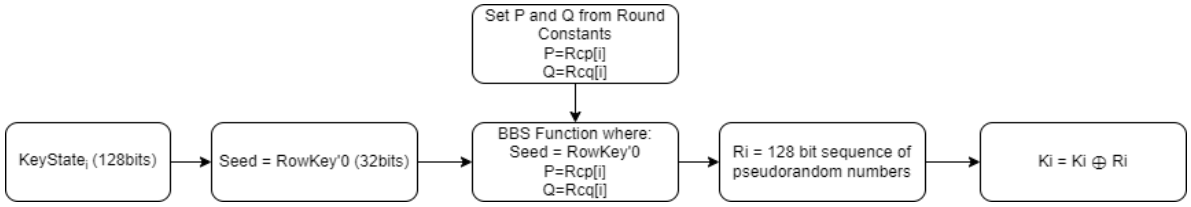


Figure 2. BlumBlumShubXOR

Modulo Addition

The operation of ModuloAddition is one of the most applied operations in symmetric cryptography. It is used in RC6, MARS, Twofish, and Rabbit stream ciphers (Dehnavi et al., 2016). According to De Los Reyes et. al (2018), the AES cipher has shown an increase in the diffusion property of the encryption with ModuloAddition. The Modulo addition is added after the AddRoundKey operation to further improve the diffusion of keys to the plaintext. Modulo addition in the encryption process is a byte operation using the Equation 2:

$$S'x = (Sx + Bwx) \bmod 64 \qquad \textit{Equation 2}$$

Where $S'x$ is the new byte after ModuloAddition in the resulting state, Sx represents a byte of the current state undergoing modulo addition, Bwx is the corresponding byte of the subkey added to the current state matrix, and 64 denotes the block length. The purpose of employing Modulo Addition at this stage is to amplify the distribution of keys within the encryption process. This extra step contributes to a more robust diffusion property, ensuring that changes in the plaintext propagate extensively throughout the ciphertext.

NonceXOR

Nonce or 'Number used once', is a value that is used only once within a specified context. The nonce is a random 128-bit data that is modifiable and embedded in the algorithm. The inclusion of a nonce in the key schedule algorithm aims to enhance the confusion property during round key generation. Moreover, it serves to eliminate any correlation between the secret key and the generated round keys (Zakaria, 2022). The nonce that was used in the algorithm is "NCS VLRD", derived from the researchers' last

names as shown in Table 2. After performing substitution from the SubColumn, a NonceXOR is added to further improve the confusion property of the algorithm.

Table 2. Nonce

Nonce	
ASCII	NCS VLRD
Hexadecimal	4E 43 53 20 56 4C 52 44
Binary	01001110 01000011 01010011 00100000 01010110 01001100 01010010 01000100

Conceptual Framework

Figure 3 illustrates the proposed modifications designed for the RECTANGLE Algorithm to effectively address three identified issues. The process begins with user input of plaintext and key, converted from ASCII to hexadecimal format, followed by encryption using RECTANGLE. The Key Schedule Algorithm first generates round keys from the secret key, with modifications enhancing key generation. Round Constants are replaced with BlumBlumShubXOR to enhance encryption security. The generated round keys encrypt the plaintext in RECTANGLE cipher rounds. ModuloAddition and NonceXOR are employed to increase complexity and randomness in the resulting ciphertext, enhancing confusion and diffusion attributes. These adjustments aim to strengthen encryption resilience against vulnerabilities. The ciphertext is then produced as the output.

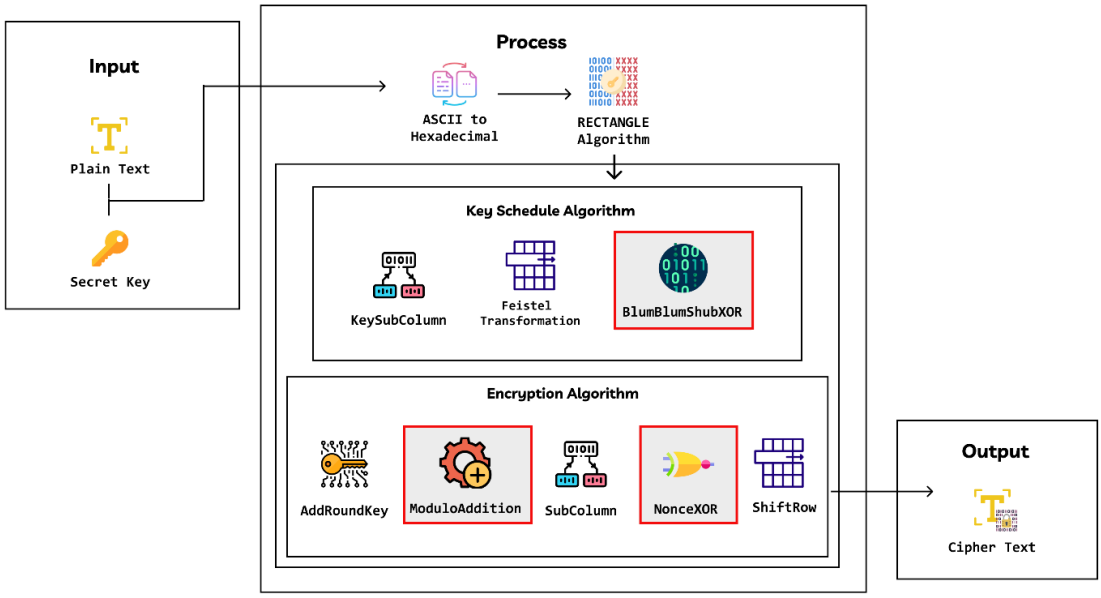


Figure 3. IPO Model of the Enhancement of the RECTANGLE Algorithm

RESULTS

This section encompasses a comprehensive comparison of the results derived from different algorithms, including the original RECTANGLE Algorithm, the Modified Key

Schedule Algorithm, the Extended 3D RECTANGLE, and the Proposed Enhancement of the RECTANGLE Algorithm. To facilitate a thorough evaluation, three distinct tests will be meticulously executed to obtain a comprehensive finding. The data acquired from these tests will serve as the foundation for drawing conclusions and insights.

Frequency Test

The goal of the first problem is to improve round key generation to achieve a 90% success rate on the Frequency Test of each round using the Blum-Blum-Shub CSPRNG in the Key Schedule Algorithm. Researchers used the NIST Statistical Test Suite's frequency test to demonstrate this improvement. The test is considered passed if the p-value is greater than or equal to 0.01 (Bassham et al., 2010).

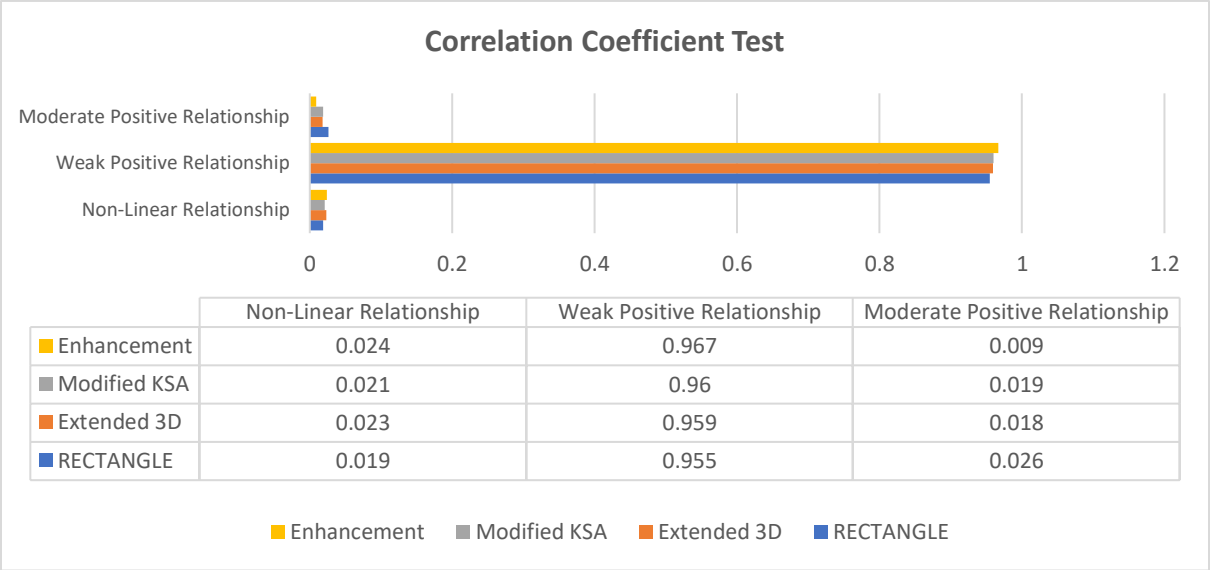
Figure 4 shows the Frequency Test Results of the four algorithms. Green-highlighted values are p-value rounds that passed, while red-highlighted values are those that failed.

Figure 4. Frequency Test Results

Round p-value	RECTANGLE	Extended 3D	Modified KSA	Proposed Enhancement
1	0.000000	0.000000	0.000002	0.378138
2	0.000000	0.000000	0.000000	0.437274
3	0.000000	0.000000	0.000648	0.671779
4	0.253551	0.253551	0.000000	0.804337
5	0.000000	0.000000	0.213309	0.500934
6	0.000000	0.000000	0.000000	0.253551
7	0.000000	0.000000	0.025193	0.005490
8	0.006990	0.006990	0.000296	0.012650
9	0.006990	0.006990	0.074177	0.048716
10	0.834308	0.834308	0.804337	0.739918
11	0.534146	0.534146	0.000000	0.568055
12	0.000000	0.000000	0.000000	0.066882
13	0.000000	0.000000	0.025193	0.074177
14	0.090936	0.090936	0.000000	0.090936
15	0.000000	0.000000	0.001232	0.671779
16	0.000737	0.000737	0.000000	0.100508
17	0.006196	0.006196	0.378138	0.253551
18	0.000000	0.000000	0.000000	0.100508
19	0.000000	0.000000	0.000569	0.911413
20	0.000035	0.000035	0.000000	0.178278
21	0.009998	0.009998	0.000045	0.534146
22	0.000134	0.000134	0.000000	0.082177
23	0.000134	0.000134	0.195163	0.060239
24	0.000000	0.000000	0.000000	0.232760
25	0.002624	0.002624	0.232760	0.275709
TOTAL:	**4/25**	**4/25**	**8/25**	**24/25**

Correlation Coefficient Test

Figure 5 presents the correlation coefficient comparison results. While all algorithms exhibit weak positive correlations, the Proposed Enhancement notably displays fewer instances of Moderate Positive Relationships, outperforming the others.

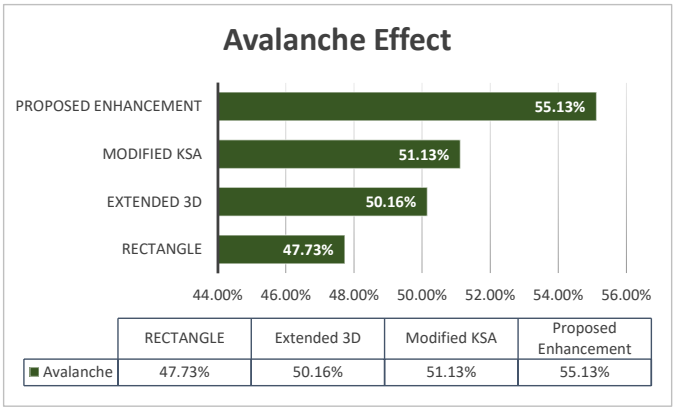


	Non-Linear Relationship	Weak Positive Relationship	Moderate Positive Relationship
Enhancement	0.024	0.967	0.009
Modified KSA	0.021	0.96	0.019
Extended 3D	0.023	0.959	0.018
RECTANGLE	0.019	0.955	0.026

Figure 5. Correlation Coefficient Test Results

Avalanche Effect Test

A comparative analysis evaluating the avalanche effect was conducted to assess the performance of the Proposed Enhancement against other algorithms, as presented in Figure 6. The avalanche effect of an algorithm should produce at least a 50% change in the output text (Abikoye et al., 2019).



	RECTANGLE	Extended 3D	Modified KSA	Proposed Enhancement
Avalanche	47.73%	50.16%	51.13%	55.13%

Figure 6. Avalanche Effect Test Results

Performance Test

Figure 7 shows the performance of the algorithms when it comes to their speed and throughput. The original algorithm retains its position as the fastest among the evaluated algorithms. Despite the series of modifications and enhancements, the proposed algorithm demonstrates commendable performance improvements. Notably, these enhancements targeted and successfully improved the specific properties within the algorithm that required refinement.

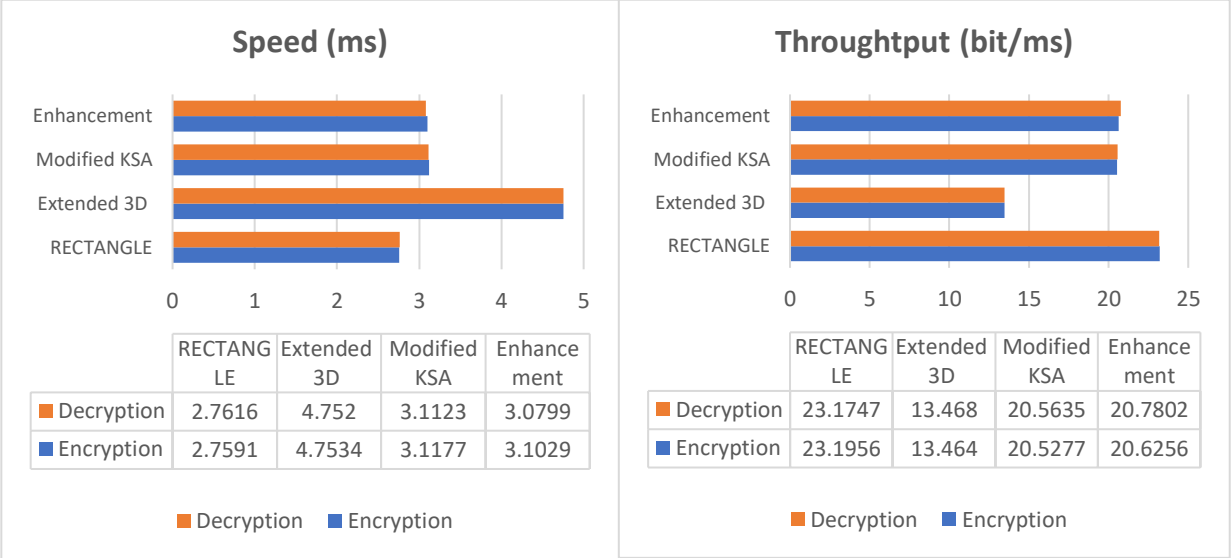
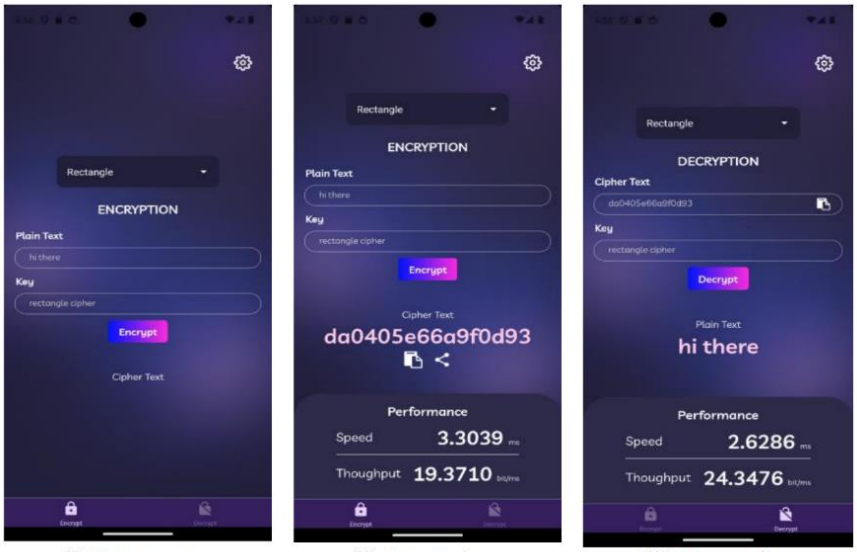


Figure 7. Speed and Throughput Test Result



(i) Homepage (ii) Encryption (iii) Decryption

Figure 8. Mobile Application

Mobile Application Implementation

The Mobile App offers users the option to encrypt or decrypt data, requiring two essential inputs: a plaintext message and a key. After encryption or decryption, the app promptly presents the resulting output text and detailed performance metrics, including speed and throughput. For encryption, users input their plaintext message and encryption key, initiating the process. Once completed, the app reveals the encrypted text and provides insights into encryption performance, such as speed and overall throughput. Similarly, for decryption, users input the encrypted message and decryption key. After

decryption, the app displays the decrypted text and offers a comprehensive analysis of decryption performance, including speed and throughput metrics. Figure 8 shows the screenshots from the mobile application.

DISCUSSIONS

This section includes an in-depth discussion of the observed outcomes, potential reasons for variations, and implications of the findings. For the Frequency test results, the original RECTANGLE Algorithm and Extended 3D, only 4 out of 25 rounds passed the desired p-value, indicating issues with the algorithm's strength and security. In the Modified Key Schedule Algorithm, 8 out of 25 rounds passed the desired p-value, an improvement over the original but still concerning regarding strength and security. For the Proposed Enhancement, 24 out of 25 rounds (96%) met the desired p-value, demonstrating that the algorithm is non-random. This surpasses the 90% success rate requirement, highlighting significant improvement in the algorithm's robustness and performance.

In the Correlation Coefficient Test despite sharing weak positive relationships, the Proposed Enhancement demonstrates an advantage with fewer moderate positive relationships, leading to more favorable overall outcomes compared to the other algorithms. For the avalanche effect test, the Proposed Enhancement has reached the desired result, which is a 50% avalanche effect, showcasing superior outcomes. This enhances the diffusion property, addressing a previously identified concern. While the two other algorithms demonstrated nearly identical avalanche effects, the Proposed Enhancement stood out as the most impactful among the four, highlighting its substantial improvement in enhancing the diffusion property, which was previously a challenge. Lastly, the performance achievement underscores the success of the proposed modifications in significantly enhancing the algorithm's overall performance, showcasing tangible advancements in areas that demanded improvement.

CONCLUSIONS AND RECOMMENDATIONS

The enhancement has significantly improved the overall RECTANGLE Algorithm, as evidenced by the test results. Incorporating BlumBlumShubXOR into the Key Schedule Algorithm (KSA) has notably enhanced randomness, achieving a 92% success rate in p-values from the Frequency Test. This addresses previous shortcomings in round key generation. Additionally, integrating the NonceXOR function into the encryption algorithm has effectively improved the confusion property of RECTANGLE. The Correlation Coefficient Test reveals advancements in non-linear relationship qualities, enhancing overall efficacy. Furthermore, the ModuloAddition function has enabled the enhanced algorithm to achieve a commendable 55% avalanche effect, surpassing the minimum threshold of 50%, indicating robust randomization capabilities.

While the modifications have slightly decreased the algorithm's performance, the enhanced version still demonstrates commendable improvements and outperforms the

previous enhancements in terms of security. Despite the original algorithm being the fastest among the evaluated algorithms, the enhanced algorithm's heightened security compensates for its decreased performance, making it a robust encryption algorithm with competitive performance.

Researchers recommend exploring alternative enhancement techniques, such as experimenting with different key schedule designs or substitution-permutation network structures, to further improve algorithm performance and security. Additionally, refining the algorithm modifications is suggested to mitigate potential performance degradation by optimizing the implementation of new functions or adjusting parameter values for a better balance between performance and security.

IMPLICATIONS

The findings from the tests conducted on the enhanced RECTANGLE Algorithm reveal significant improvements across different tests conducted. By enhancing randomness, addressing issues in key generation, improving confusion and diffusion properties, and achieving robust randomization capabilities, the algorithm demonstrates heightened security and efficacy in encryption tasks. While some modifications may slightly impact performance, the overall enhancement solidifies the algorithm's position as a competitive and robust encryption solution in the realm of lightweight block cipher algorithms. These results underscore the importance of continuous refinement and adaptation in cryptographic algorithms to meet evolving security requirements and challenges.

ACKNOWLEDGEMENT

The researchers extend their appreciation to their thesis adviser, coordinators, and panel for their invaluable guidance and unwavering support throughout this research endeavor. Their expert knowledge, mentorship, and encouragement played a pivotal role in this study. Additionally, the researchers wish to acknowledge the Department of Science and Technology (DOST) for their funding and support for this research.

FUNDING

The study did not receive funding from any institution.

DECLARATIONS

Conflict of Interest

The researcher declares no conflict of interest in this study.

Informed Consent

Not Applicable, since the study focuses on algorithm development and does not involve human subjects or personal data, informed consent is not required.

Ethics Approval

Not Applicable. It solely pertains to the development and testing of encryption algorithms, which does not necessitate ethics approval.

REFERENCES

Abikoye, O. C., Haruna, A., Abubakar, A., Oluwatobi, A. N., & Asani, E. O. (2019). Modified advanced encryption standard algorithm for information security. *Symmetry, 11*(12), 1484. https://doi.org/10.3390/sym11121484

Afzal, S., Yousaf, M., Afzal, H., Alharbe, N., Rafiq, M (2020). Cryptographic Strength Evaluation of Key Schedule Algorithms. Security and Communication Networks. 2020. 1-9. 10.1155/2020/3189601.

Bassham, L., Rukhin, A., Soto, J., Nechvatal, J., Smid, M, Leigh, S, Levenson, M., Vangel, M., Heckert, N., & Banks, D. (2010). *A statistical test suite for random and pseudorandom number generators for cryptographic applications*. National Institute of Standards and Technology, U.S. Department of Commerce. Retrieved from https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=906762

Dehnavi, S. M., Rishakani, A. M., Shamsabad, M. R. M., Maimani, H., & Pasha, E. (2016). *Cryptographic Properties of Addition Modulo 2^n*. Cryptology ePrint Archive. https://eprint.iacr.org/2016/181

Dhanda, S. S., Singh, B., & Jindal, P. (2020). Lightweight cryptography: a solution to secure IoT. *Wireless Personal Communications, 112*(3), 1947–1980. https://doi.org/10.1007/s11277-020-07134-3

Divyanjali, Ankur, & Pareek, V. (2014). An Overview of Cryptographically Secure PseudorandomNumber Generators and BBS. *International Journal of Computer* Applications *(IJCA), 2*, 19-28.

De Los Reyes, E. M., Sison A. M. & Medina R. P. (2018, October). Modified AES Cipher Round and Key Schedule. In *Proceedings of the 2018 International Conference on Intelligent Informatics and Biomedical Sciences* (ICIIBMS) (pp. 146-146). IEEE. https://ieeexplore.ieee.org/document/8549995

Nakahara Jr, J. (2008, December). 3D: A three-dimensional block cipher. In M. K. Franklin, L. C. K. Hui, & D. S. Wong (eds), *Proceedings of the International Conference on Cryptology and Network Security* (pp. 252-267). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-89641-8_18

Naser, N. M., & Naif, J. R. (2022). A systematic review of ultra-lightweight encryption algorithms. *International Journal of Nonlinear Analysis and Applications, 13*(1), 3825–3851.

Omrani, T., Rhouma, R., & Sliman, L. (2018, July). Lightweight Cryptography for Resource-Constrained Devices: A Comparative Study and Rectangle Cryptanalysis. In M. Bach Tobji, R. Jallouli, Y. Koubaa, & A. Nijholt (eds), *Lecture Notes in Business Information Processing* (pp. 107–118). Springer. https://doi.org/10.1007/978-3-319-97749-2_8

Salunke, R., Bansod, G., & Naidu, P. V. (2019, July). Design and implementation of a lightweight encryption scheme for wireless sensor nodes. In K. Arai, R. Bhatia, & S. Kapoor (eds), *Advances in Intelligent Systems and Computing* (pp. 566–581). Springer. https://doi.org/10.1007/978-3-030-22868-2_41

Yan, H., Luo, Y., Chen, M., & Lai, X. (2019). New observation on the key schedule of yan. *Science China Information Sciences, 62*(3), Article Number 32108. https://doi.org/10.1007/s11432-018-9527-8

Zakaria, A. A. B. (2022). *An improved rectangle lightweight block cipher based on 3D rotation* method. USIM Research Repository System. Retrieved from https://oarep.usim.edu.my/jspui/handle/123456789/18001

Zakaria, A. A., Azni, A. H., Ridzuan, F., Zakaria, N. H., & Daud, M. (2021). Modifications of key schedule algorithm on RECTANGLE Block Cipher. In M. Anbar, N. Abdullah, & S. Manickam (eds), *Communications in Computer and Information Science* (pp. 194–206). Springer. https://doi.org/10.1007/978-981-33-6835-4_13

Zakaria, A. A., Azni, A. H., Ridzuan, F., Zakaria, N. H., & Daud, M. (2020). Extended RECTANGLE algorithm using 3D bit rotation to propose a new lightweight block cipher for IoT. *IEEE Access, 8*, 198646–198658. https://doi.org/10.1109/access.2020.3035375

Zakaria, A. A., Azni, A. H., Ridzuan, F., Zakaria, N. H., & Daud, M. (2020, June). Randomness Analysis on RECTANGLE Block Cipher. In *Proceedings of the 7th International Cryptology and Information Security Conference 2020* (pp. 133-142). Retrieved from https://mscr.org.my/cryptology/proceeding/Cryptology2020.pdf#page=147

Zhang, W., Bao, Z., Lin, D., Rijmen, V., Yang, B., & Verbauwhede, I. (2015). RECTANGLE: a bit-slice lightweight block cipher suitable for multiple platforms. *Science China Information Sciences, 58*(12), 1–15. https://doi.org/10.1007/s11432-015-5459-7

Author's Biography

Karylle Andrei Velarde is a Computer Science Student at Pamantasan ng Lungsod ng Maynila.

Raphael Enciso is a Computer Science Student at Pamantasan ng Lungsod ng Maynila and works as a Web Developer.

Vivien Agustin is a Professor and the Associate Dean of the College of Information System and Technology Management at Pamantasan ng Lungsod ng Maynila.

Jonathan Morano is a Professor and a Faculty Member of the College of Information System and Technology Management at Pamantasan ng Lungsod ng Maynila.

Leisyl Mahusay is a Professor and a Faculty Member of the College of Information System and Technology Management at Pamantasan ng Lungsod ng Maynila.

Jamillah Guialil is a Professor and a Faculty Member of College of Information System and Technology Management at Pamantasan ng Lungsod ng Maynila.