INTERNATIONAL JOURNAL ON ADVANCED TECHNOLOGY, ENGINEERING, AND INFORMATION SYSTEM (IJATEIS)

RANDOMNESS IDENTIFICATION VIA XOR REPOSITION IN LFSR A5/1'S SECOND BLOCK

Jorghie Theodore Pallangan[1], Alz Danny Wowor[2*]
[1,2]Faculty of Information Technology,
Satya Wacana Christian University, Salatiga, Central Java, Indonesia
E-mail: [2)] alzdanny.wowor@uksw.edu

Abstract

This study designs a random number generation method using the LFSR approach with the A5/1 scheme on three feedback functions. XOR is used as an operation in determining the value of the new bit output against the next iteration of the feedback function. Runs Test, Mono Bit, and Block bit, are used as test materials in producing random output against an input. The use of three feedback functions is used in testing, compared to previous studies that produce random numbers. In the plaintext and ciphertext encryption tests, it shows a "Very Small" correlation level with an average value approaching 0. The use of the LFSR A5/1 scheme with three XOR functions produces random output and can be used for Stream Cipher.

Keywords: *Linear Feedback Shift Register, Cryptography, A5/1 Schematic*

1. INTRODUCTION

Cryptography is a necessary algorithm and is often used in securing in-formation. Assurance of security in information can affect the level of user trust in using the algorithm or application. A good algorithm will certainly take into account the complexity of time and space, so that the encryption-decryption process can be optimal. The optimum level of a cryptographic algorithm can be started by paying attention to the key generation process, in which it can accept arbitrary inputs and can produce random outputs, so that in the encryption process, the ciphertext can hide important information from the plaintext (Wowor & Susanto, 2023).

Linear feedback shift registers (LFSRs) can generate a key to randomness. By shifting the input bit, exclusive-or (XOR) is used as the determinant of the maximum period random bit (Herman, 2022). Input bits can be used as generators against the entire bit of the sliding register or feedback function. LFSR with A5/1 scheme is one of the optimal algorithms in generating random bit externals. The chart of the complete LFSR A5/1 scheme is given in Figure 1.

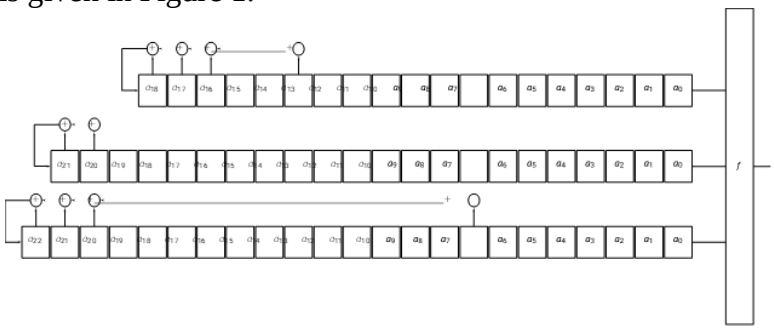

Figure 1. A5/1 Algorithm Design

The A5/1 scheme uses three blocks, where each block has an XOR function process as feedback to carry out the next iteration process. 1. Each block has a role in generating random bit outputs, as well as the selection of each bit entry in each block. In the studies on the A5/1 scheme, there has been no explanation regarding the selection of a13, a16, a17 and a18 as selected entries in the first block, as well as in the second and third blocks.

$$A2 = A20 \oplus A21 \quad (1)$$

The study in the current study is to use the bit entry in the second block, or to reposition another bit to produce another A2. Randomness testing is a reference to distinguish each reposition that produces the best randomness value. In addition, encryption testing is also used to test each bit output generated from the repositioning process in A5/1, so a comparison process is carried out to see if there is a better repositioning compared to Equation 1. There are 22-bits in total and 2-bits taken, because the XOR function is subject to commutative law so that the number of 2-bits in different positions will produce the same output.

2. LITERATURE REVIEW

2.1. Previous Research

In this study, polynomial functions from previous studies were used to underlie the random number generation test. As in Table 1.

Table 1. Related Research

No.	Author Name	Research Title	Research Problem	Method	Results
1	Daurat Sinaga, Chaerul Umam, De Rosal Ignatius Moses Setiadi, Eko Hari Rachmawanto (2018)	"Super Encryption Technique Using Column Transposition Based Vigenere Cipher on Digital Image"	Time required in the process of Encryption and Description of Digital Image	Methods used in the implementation of the combination of column transposition and vigenere cipher methods to secure text files,	The result of the comparison shows that the execution time of the message decryption process is longer for Vigenere cipher compared to Vernam cipher on a grayscale image with a size of 256x256 pixels.
2	Gede Aditra Pradnyana, and Ida Bagus Putu Suarma Putra (2018)	"Securing Digital Data Files with A Combination Algorithm of Triple Transposition Vigenere Cipher and Huffman Method"	Testing the Triple Transposition Vigenere Cipher Algorithm and the Huffman Method	Metode Triple Transposisi Vigenere Cipher dan Metode Huffman	Produces a high level of security and can be used for remote health monitoring systems

3	Herdyan Kharisma Putra and Sunny Arief Sudiro (2018)	"Triple Transposition and Spread Spectrum as Methods for Steganograph Algorithm Development"	Secure data by considering bandwidth usage	Metode Triple Transposition Vig`enere Cipher dan metode spread spectrum)	Generate a security level that can be used for remote health monitoring systems

Research by Sinaga et al. (2018) modified the Shift Row and Mix Column AES operations by looking at the problem of a slow process for processing 1024 bytes of data. The modification process succeeded in producing a better process with 3.45 milliseconds for 1024 bytes, and 2048 bytes required 3-4 milliseconds. The modifications made reduce the process time with an average optimization of 86.143%.

Research by Pradnyana and Putra (2018) designed a cryptographic system using genomic encryption and deterministic chaos methods, resulting in a fast and secure algorithm to secure medical device data in real time. Experimental results and encryption analysis show that the proposed algorithm provides a high level of security and can be used for remote health monitoring systems.

Putra and Sudiro (2018) uses the transposition process of the Triple Transposition Vigenere Cipher and then compresses it with the Huffman Method. This is done as a solution in securing data by considering the use of small memory bandwidth. The cryptography application was successfully designed and can perform the encryption and decryption process properly. The system testing process uses manual calculations with the system and blackbox testing.

2.2. A5/1 Scheme

The A5/1 scheme is used as a security medium against eavesdropping and theft on 2G or GMS networks (Sadkhan & Jawad, 2015). A5/1 is a collection of several LFSRs. With A as the main function of each linear function output AI.

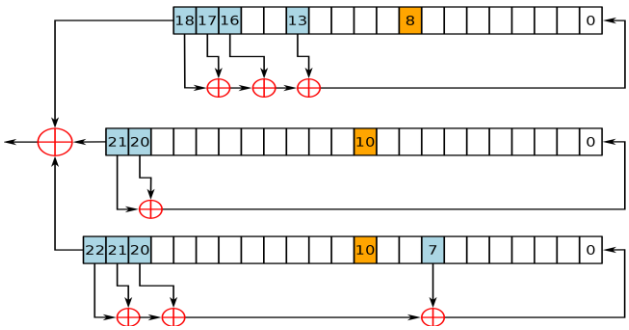

Figure 2. A5/1 Algorithm Schematic

Figure 2 consists of three rows of LFSR, the first row (R1) has a length of 19 bits starting from bits 0 to 18, with the feedback polynomial function contained in bits 13, 16, 17 and 18. The second row (R2) has a length of 22 bits starting from bits 0 to 21, with the feedback polynomial function contained in bits 20 and 22. The third row (R3) has a length of 23 bits starting from bits 0 to 18, with the feedback polynomial function contained in

bits 7, 20, 21, 23. Clocking bits on each row. Where in R1 there is bit 8, while in R2 and R3 there is bit 10.

3. RESEARCH METHODS

The flow of the research design is used as a solution to research problems, as well as to help in the testing process. With the explanation in Figure 3.

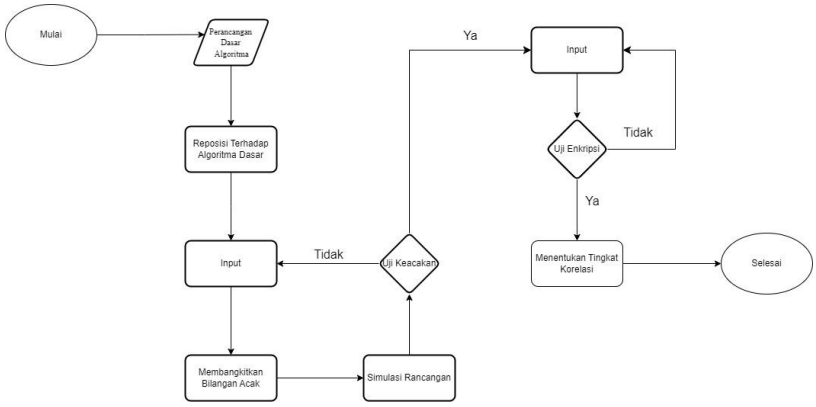

Figure 3. A5/1 Algorithm Design with 3 XOR Functions

Starting with Stage Per Design of the feedback function Ai = 1, 2, 3, followed by re-positioning the XOR operation on block A1, then designing the algorithm for each reposition that has been designed. The stage of generating random bits begins by entering the plaintext, not forgetting that the input is changed to an ASCII code which is then used as a binary number, after obtaining the binary number the padding process is carried out. Simulate the model, by performing the Runs Test, bl,mb operation, which is then saved. If all repositioning plans have been carried out and the data is still lacking, repeat by entering the plaintext, and if the data is sufficient, it will be continued by searching for each randomness generation operation. The average will then be selected in the top 10 as descriptive testing materials. The decryption test begins by entering ciphertext, the input will be converted into an ASCII number which is used as a reference in performing the in level, and classified based on the level of correlation.

4. RESULTS AND DISCUSSION

Figure 1 is used as the basis for the LFSR algorithm using three XOR function blocks, by performing some repositioning on function A1 against the main function f on the XOR operating model.

$$A1 = a13 \oplus a16 \oplus a17 \oplus a18$$
$$A2 = a20 \oplus a21$$
$$A3 = a7 \oplus a20 \oplus a21 \oplus a22 \quad (2)$$

Equation 2 describes the XOR operation that occurs on each block, with Ai expressed on the feedback function, with Ai = {1, 2, 3}, to generate a random bit output on each block.

$$f = A1 \oplus A2 \oplus A3 \quad (3)$$

Equation 3 states as the primary function in combining each AI feedback function.

4.1. Random Number Generation Process

The process of calculating the initialization value is carried out by converting the input (plaintext) into an ASCII code, the output of the ASCII code is converted into a binary number. Padding is used as a constraint on the division of each feedback function, so that each block can be selected bits that will be a random bit generator with the XOR function. after obtaining random bits in each block, and then randomized according to Equation 2.

4.2. Randomness Testing

Microsoft Office Excel is used as a test process tool, there is a limit to plaintext input with a length of only 8 characters or 64 bits. If the length of the character is less than 8, the length of the character will automatically be filled in with "Space", but if it is more than 8 characters, the 9th character and so on will be forgotten.

This test uses statistical methods, including Mono Bit, Block Bit, and Runs Test. The inner output value is stated as "non-random" if $\alpha \leq 0.01$ and for $\alpha \geq 0.01$ is declared "random".

Table 2. Top 10

No.	Algorithm	*p-value*	Experimental Results
1	L1105	0.69725037	RANDOM
2	L401	0.671025191	RANDOM
3	L2015	0.669059667	RANDOM
4	L2018	0.656004083	RANDOM
5	L2116	0.653523574	RANDOM
6	L907	0.651809705	RANDOM
7	L1707	0.651462567	RANDOM
8	L1916	0.647039165	RANDOM
9	L1600	0.643445309	RANDOM
10	L1716	0.637366173	RANDOM
	L2120	0.555686361	RANDOM

Table 2 is the average p value of the Runs Test, Mono Bit, and Block Bit tests. Then the best 10 will be taken as an encryption tester, in the best 10 results the output value is higher than the reference algorithm (L2120). In tables 1, 2, 3 will make the table as a reference.

Table 3. Test Results Runs Test

No.	Algorithm	*p-value*	Experimental Results
1	L1105	0.655152055	RANDOM
2	L401	0.660147117	RANDOM
3	L2015	0.67912007	RANDOM
4	L2018	0.574181903	RANDOM
5	L2116	0.619626264	RANDOM
6	L907	0.608093271	RANDOM
7	L1707	0.547159438	RANDOM
8	L1916	0.63468669	RANDOM
9	L1600	0.62261367	RANDOM
10	L1716	0.583004744	RANDOM
	Average	0.618378522	RANDOM

Table 3 states the p value in the results of the Runs Test with an average value of 0.618378522.

Table 4. Mono Bit Test Results

No.	Algorithm	*p-value*	Experimental Results
1	L1105	0.826500041	RANDOM
2	L401	0.757100057	RANDOM
3	L2015	0.700920004	RANDOM
4	L2018	0.80501998	RANDOM
5	L2116	0.757100057	RANDOM
6	L907	0.791800049	RANDOM
7	L1707	0.839719972	RANDOM
8	L1916	0.679439943	RANDOM
9	L1600	0.649732925	RANDOM
10	L1716	0.748839927	RANDOM
	Average	0.755617295	RANDOM

Table 4 produces the average *Mono Bit* with *a p value* of 0.755617295.

Table 5. Block Bit Test Results

No.	Algorithm	*p-value*	Experimental Results
1	L1105	0.610099016	RANDOM
2	L401	0.595828401	RANDOM
3	L2015	0.627138927	RANDOM
4	L2018	0.588810367	RANDOM

5	L2116	0.583844399	RANDOM
6	L907	0.555535794	RANDOM
7	L1707	0.567508292	RANDOM
8	L1916	0.626990863	RANDOM
9	L1600	0.657989331	RANDOM
10	L1716	0.580253848	RANDOM
	Average	0.599399924	RANDOM

Table 5 in the Block Bit experiment with a p value output with an average of 0.599399924.

4.3. Encryption-Decryption Testing

The encryption test process has a limit of 32 characters or 256 bits, which will be converted into ASCII code, this time the operation uses the modulus 256 as its domain in the EK encryption test and DK description.

$$Oak : P \oplus K = C, \qquad Dk : C \oplus K = P \quad (4)$$

In the encryption test process , three ciphertext inputs are used which are expected to represent variations of the text that are usually used by users. Among other things, the first ciphertext is Mother cooking fried rice in the kitchen, to represent ordinary writing that consists only of a combination of letters. The second ciphertext Xg7l2yvkR mzWbˆ ˆoQf1Jh8aTc*5dPn uses a combination of letters, symbols, and numbers. The last ciphertext is GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG- GT which has the same letter.

The results of the correlation test were used to evaluate the relationship between ciphertext and plaintext. This is done to find out how well the key can disguise the plaintext and eliminate the relationship between the two. The test results for all three plaintexts with each data set are given in Table 6.

Table 6. Correlation Test Results

No	Algorithm	Tested Plaintexts			Average Correlation Rate	
		Plaintext 1	Plaintext 2	Plaintext 3		
1	L1105	-0.0791065573	0.0536663475	0.0160611218	-0.00312636265	VERY SMALL
2	L401	-0.00552138204	-0.0479740015	-0.0583653367	-0.0372869067	VERY SMALL
3	L2015	0.109137513	0.0117155058	0.0178103923	0.0462211373	VERY SMALL
4	L2018	0.0842099542	0.0565328818	-0.0590977002	0.0272150452	VERY SMALL
5	L2116	0.0104899774	0.0390152729	-0.0512419746	-0.000578908098	VERY SMALL
6	L907	-0.124469115	0.0139834417	-0.0315269695	-0.0473375478	VERY SMALL
7	L1707	-0.137821672	0.000831865725	0.046683191	-0.0301022051	VERY SMALL
8	L1916	0.00357317658	0.0277879610	-0.0266656567	0.0015651602	VERY SMALL
9	L1600	0.0359920778	0.0452246544	-0.0141379171	0.0223596050	VERY SMALL
10	L1716	0.00607874040	0.0325805648	-0.0559050021	-0.0057485656	VERY SMALL

Table 6 explains that the average interval is close to 0 with a very small degree of correlation.

4.4. Discussion

The repositioning of the XOR function on the second block of the LFSR A5/1 successfully increased the level of randomness of the output produced. This is proven through statistical testing using Runs Test, Mono Bit, and Block Bit, which show a P-value of more than 0.01, so the resulting data is considered random. These results show that modifications in the XOR function can produce random numbers that are quantified and meet the standards required in cryptographic applications. Some previous studies have stated that LFSR with XOR is already quite effective in generating random numbers, but this study shows that with certain repositioning, more optimal results can be achieved. One of the strengths of this study is the use of three feedback functions, which increases the variety of results. This study aims to explore the improvement in random number generation through the modification of the XOR function in the second block of LFSR A5/1.

5. CONCLUSION

In this study, the repositioning of each algorithm can generate random numbers with a very good correlation rate. The test was carried out with Microsoft Excel as a tool, using the Runs Test, Mono Bit and Block Bit methods as random number generators, resulting in an average p value of less than 0.01 for each algorithm. After obtaining the output value of the randomization method, it is carried out on average for the method and the best 10 are taken for encryption tests. The encryption test was conducted using three different plaintexts with a very low correlation rate, while the algorithm used as a reference determined a low correlation level. It can be concluded that repositioning obtains a safe correlation of keys and random outputs in a cryptography.

REFERENCES

Herman, A. J. (2022). *Desain Pembangkit Kunci LFSR dengan Skema A5/1 Menggunakan 7 Blok Bit Fungsi XOR*.

Pradnyana, G. A., & Putra, I. B. P. S. (2018). Pengamanan Berkas Data Digital Dengan Algoritma Kombinasi Triple Transposition Vigenere Cipher Dan Metode Huffman. *Jurnal Pendidikan Teknologi Dan Kejuruan*, *15*(1).

Putra, H. K., & Sudiro, S. A. (2018). Triple Transposisi dan Spread Spectrum sebagai Metode untuk Pengembangan Algoritme Steganogra fi. *Jurnal Ilmiah KOMPUTASI*, *17*(2), 161–168.

Sadkhan, S. B., & Jawad, N. H. (2015). Simulink based implementation of developed A5/1 stream cipher cryptosystems. *Procedia Computer Science*, *65*, 350–357.

Sinaga, D., Umam, C., & Rachmawanto, E. H. (2018). Teknik Super Enkripsi Menggunakan Transposisi Kolom Berbasis Vigenere Cipher Pada Citra Digital. *Dinamika Rekayasa*, *14*(1), 57–64.

Wowor, A. D., & Susanto, B. (2023). One to many (new scheme for symmetric cryptography). *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, *21*(4), 762–770.

Copyrights