Design of AL-13 Block Cipher Algorithm Based On Extended Feistel Network

Mohamad Ali Sadikin¹, Bety Hayat Susanti²

¹Faculty of Engineering, National Crypto Institute, Bogor, Indonesia
²Faculty of Engineering, National Crypto Institute, Bogor, Indonesia
¹mohamadalisadikin@gmail.com, ²bety.hayat@stsn-nci.ac.id

Abstract. In this paper, we design a new block cipher algorithm called AL-13. The structure of AL-13 is simple. We use Extended Feistel Network Type II with input-output 128 bits and key input 256 bits. We also use Dragon’s key schedule, binary addition, modulus addition, and XOR to give diffusion, and two functions F and N for confusion. We conducted Strict Avalanche Criterion (SAC) tests to determine whether AL-13 satisfy cryptographic measurement parameters.

1. Introduction
Symmetrical cryptosystems are an encryption system in which the sender and receiver share a single, common key that is used to encrypt and decrypt a message. There are two kinds of it, i.e. block cipher and stream cipher [1]. In block ciphers, the plaintext is divided into block of a fixed length, which are then encrypted into blocks of ciphertext of the same length using the same key [2]. There are seven criteria for evaluating block ciphers, which estimates the security level, the key size, throughput, block size, complexity cryptographic mapping, additional data, and error propagation [3]. Whereas according to Shannon, there are two principles that are often used in encryption block cipher, confusion and diffusion. It is a difficult challenge for the designers to design a strong and secure block cipher algorithm that fulfill the criteria, and have principles of confusion and diffusion [4]. In this paper, we present a 128-bit block cipher called AL-13 which is designed using Feistel structure [5] and is considered to have the principles of confusion and diffusion [3]. Principles of confusion and diffusion [6] that already exist can be combined or repeated many times with different techniques and different combinations to get stronger security.

2. AL-13 BLOCK CIPHER ALGORITHM
AL-13 Cipher Algorithm is a block cipher algorithm with a 256-bit key input and 128-bit plaintext. AL-13 algorithm is designed based on Extended Feistel Network type II structure. The number of rounds is 16-round, where in each round there are two distinct functions, namely F and N function. F and N function using the component operations of addition, XOR, delta, and SUM. We use Dragon’s key schedule for AL-13’s key schedule [7].

2.1. Notations
⊕ bitwise exclusive-OR operation
⊞ modulo addition operation
Delta golden number (\(\sqrt{5}-1\))*2^{31}
SUM initial value
\| concatenation of two operands
2.2. Key Scheduling
Key scheduling process of AL-13 algorithm based on Dragon’s cipher algorithm with key input length is 256-bit. For more details, see in [7].

2.3. Encryption and Decryption Scheme
Encryption/decryption scheme of AL-13 algorithm is to process 128-bit plaintext into 128-bit ciphertext and vice versa. The following is a description of encryption and decryption of AL-13 algorithm process.

1. Partition input plaintext/ciphertext
   At this stage, 128-bit plaintext/ciphertext input is partitioned into four subblock 32-bit ($A_0, B_0, C_0, D_0$).

2. Transformation round
   The next stage is the transformation round with the same operation for each round. We use as many as 16 rounds of iteration.

3. The formation of a 128-bit ciphertext/plaintext
   128 bit ciphertext/plaintext formed from the merger of subblock $A_{16}, B_{16}, C_{16}, D_{16}$ by 32 bits.

![Figure 1. Structure of Encryption/Decryption of AL-13 Algorithm.](image-url)

2.4. F-Function
In the round process, the subblock of plaintext goes into the second F function with the following mechanism: Subblock input of C and B goes into F function. 32-bit subblock of B XORed by subkeys $Rk_0^n$. Thereafter, we do the addition modulo of these results with subblock C which is then XORed with subkey $Rk_1^n$. Later, we do the addition modulo of those results with delta and SUM. After that, we XOR the results with subblock C to produce output.
2.5. N-Function
In the round process, the subblock of plaintext goes into the second F function with the following mechanism: Subblock input of A and D goes into N function. We do the addition modulo 32-bit subblock of D with subkeys $R_k^n_1$. Hereafter, the result is XORed with subblock A (result = X) which is then modulo added with delta and SUM. Later, we do XOR subblock of A with subkeys $R_k^n_0$. After that, we XOR the results with X to get the output.

3. Rational Design and Implementation of AL-13 Algorithm

3.1. Rational Design of AL-13 Algorithm

3.1.1. The Use of F and N Function
In the AL-13 algorithm there are F function and N function that implements the concept of confusion and diffusion.

1. Confusion
The F function on the AL-13 algorithm fulfill confusion requirement. The results of XOR operation between $K_0$ and Delta is added with SUM. The goal is to obscure the incoming subblock, so that the adversary will have difficulty to find the key.
Confusion on the N function of AL-13 algorithm generated from the XOR operation conducted by $K_0$ with $K_1$ and addition operation between Delta and SUM. The goal is the same with the goal of F function.
a. Delta
Delta is the golden number which is equal to \((\sqrt{5}-1) \times 2^{31}\). Different delta is used in every round and so there are no multiplication of bits that do not change.

b. SUM
We use zero initial value for SUM. The SUM value will change every round due to the influence of the addition with delta. Thus, the value is always different in each round.

2. Diffusion
Diffusion generated from operations of addition and XOR which is intended to reduce the relationship between input and output. So, we can prevent attempts to guess the key.

3. The combination of addition and XOR operations
Couple addition and XOR operations do not satisfy the nature of associative and distributive properties that can not be converted into a simpler form to simplify the process of cryptanalysis.

3.1.2. The Use of Key Scheduling Based on Dragon’s Algorithm
In the key scheduling based on Dragon’s algorithm, there are combination of confusion and diffusion to eliminate the relationship between key input and subkey for encryption. Thus, it will be a strong parameter for the algorithm [7].

3.1.3. The Use of Extended Feistel Network Type II
AL-13 algorithm using Extended Feistel Network type II. From the test results of SAC in Extended Feistel Network type I, Extended Feistel Network type II, and Extended Feistel Network type III, we obtained that Extended Feistel Network type II and Extended Feistel Network type III fulfill the criteria of avalanche probability, i.e. 0.5 [8].

3.2. Implementation of Encryption and Decryption Process of AL-13 Algorithm

**Figure 4.** Implementation of Encryption the AL-13 Algorithm on C++.

**Figure 5.** Implementation of Decryption the AL-13 Algorithm on C++.

Figure 4 and 5 shows the implementation of encryption and decryption process of AL-13 algorithm in C++ programming. The implementation is used to proof that the result of encryption and decryption process in AL-13 algorithm is equal between in C++ programming and based on paper and pencil.
4. Strict Avalanche Criterion (SAC) Test on AL-13 Algorithm

In this section, we discuss the analysis results of SAC test on AL-13 algorithm with plaintext as independent variable and key as independent variable. We refer to [9] for the SAC test.

4.1. Analysis Results of SAC Test on AL-13 Algorithm with Independent Variable of Plaintext

Based on data from SAC test results of AL-13 algorithm with the plaintext to be treated as independent variables showed that the AL-13 algorithm passed SAC with minimum and maximum values as shown in Table 1.

| Table 1. Results of SAC test on AL-13 algorithm with plaintext as independent variable |
|---------------------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Min                             | Vector Unit     | Avalanche Vector | SAC Value (%) | Relative Error | Description      |
| 98                              | 101             | 49,23629         | 0,0152742     | Passed          |
| 100                             | 94              | 50,7454          | 0,014908      | Passed          |

According to Table 1, the largest error value is 0,0152742 and obtained by $\varepsilon_s = \max_{i,s,j,n} |2k_{SAC}(i,j) - 1| = |2 \cdot 0,4923629 - 1| = 0,0152742$.

Interval value of SAC $= \frac{1}{2}(1 - \varepsilon_s) \leq k_{SAC}(i,j) \leq \frac{1}{2}(1 + \varepsilon_s)$

$= \frac{1}{2}(1 - 0,0152742) \leq k_{SAC}(i,j) \leq \frac{1}{2}(1 + 0,0152742)$

$= 0,4923629 \leq k_{SAC}(i,j) \leq 0,5076371$

Interval value of SAC (%) $= 49,23629\% \leq k_{SAC}(i,j) \leq 50,76371\%$.

Table 1 shows that AL-13 algorithm has good diffusion properties as indicated by the largest error value that less than 2%. The value of the largest error occurs at different bit positions (unit vector) into bit position 98 and the avalanche vector to 101, meaning that when the input plaintext bits to 98 changed then output bits to 101 will be changed with a probability of 49,23629%.

In general, it can be said that changing one bit plaintext will cause changes in output bits with a probability of 50% with the largest relative error value is 0,0152742. So, it can be stated that the AL-13 algorithm has good diffusion properties.

4.2. Analysis Results of SAC Test on AL-13 Algorithm with Independent Variable of Key

The SAC test results of AL-13 algorithm with keys are treated as independent variables showed that AL-13 algorithm passed SAC test with minimum and maximum values as shown in Table 2.

| Table 2. Results of SAC test on AL-13 Algorithm with key as independent variable |
|---------------------------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Min                             | Vector Unit     | Avalanche Vector | SAC Value (%) | Relative Error | Description      |
| 98                              | 46              | 49,26223         | 0,014755      | Passed          |
| 121                             | 11              | 50,72099         | 0,01442       | Passed          |

According to Table 2, the largest error value is 0,014755 and obtained by $\varepsilon_s = \max_{1\leq i,j,n} |2k_{SAC}(i,j) - 1| = |2 \cdot 0,4926223 - 1| = 0,014755$.

Interval value of SAC $= \frac{1}{2}(1 - \varepsilon_s) \leq k_{SAC}(i,j) \leq \frac{1}{2}(1 + \varepsilon_s)$

$= \frac{1}{2}(1 - 0,014755) \leq k_{SAC}(i,j) \leq \frac{1}{2}(1 + 0,014755)$

$= 0,4926223 \leq k_{SAC}(i,j) \leq 0,5073775$

Interval value of SAC (%) $= 49,26223\% \leq k_{SAC}(i,j) \leq 50,73775\%$.

Table 2 shows that AL-13 algorithm has good confusion properties as indicated by the largest error value that less than 2%. The value of the largest error occurs at different bit positions (unit vector) into bit position 98 and the avalanche vector to 46, meaning that when the input plaintext bits to 98 changed then output bits to 46 will be changed with a probability of 49,26223%. In general, it can be said that changing one bit plaintext will cause changes in output bits with a probability of 50% with the largest relative error value is 0,0147554. Thus, it can be stated that AL-13 algorithm has good confusion properties.
5. Conclusion
In this paper, we proposed a new block cipher algorithm called AL-13 that uses Extended Feistel Network type II structure with 256-bit key input, 128-bit plaintext, and 16 total of rounds. F-function and N-function in AL-13 algorithm is used to fulfill the concept of confusion and diffusion. Furthermore, SAC test is used to test AL-13 algorithm. The test results show that AL-13 algorithm has good diffusion and confusion properties.

References

[1] Girault M, Juniot L and Robshaw M J B 2005 The Feasibility of On-the-Tag Public Key Cryptography France Telecom Research and Development
[2] Stallings W 2014 Cryptography and Network Security: Principle and Practice, Sixth Edition. (New Jersey: Prentice Hall) p 63-83
[3] Menezes A J, Van Oorschot P C and Vanstone S A 1997 Handbook of Applied Cryptography (Boca Raton: CRC press CLC) p 223-224
[4] Kevin S C 2004 A Block Cipher Cryptosystem Using Wavelet Transforms Over Finite Fields IEEE Transactions On Signal Processing, Vol. 52, No. 10. p 2975-2991
[5] Youssef A M 1997 Analysis and Design of Block Cipher Thesis in Department of Electrical and Computer Engineering, Queen’s University
[6] Schneier B 1996 Applied Cryptography: Protocols, Algorithms, and Source Code in C Second Edition (New York: John Wiley & Sons Inc) p 442
[7] Chen K., et al 2004 Dragon : A Fast Word Based Stream Cipher Int. Conf. on Information Security and Cryptology (ICISC) 2004 LNCS 3506 p 33-50
[8] Ibrahim S Maarof M A and Idris N B 2005 Avalanche Analysis of Extended Feistel Network Proceedings of the Postgraduate Annual Research Seminar FSKSM 2005 UTM p 17-18
[9] Vergili I and Yucel M D 2001 Avalanche and Bit Independence Properties for the Ensembles of Randomly Chosen n x n S-boxes Turk J Elec Engin, vol. IX p 50-62