Diffusion Property Analysis of RC5, TEA, VEA, and CFN Algorithm using Avalanche Weight Distribution Criterion (AWD) and Avalanche Criterion (AC)

Ardya Suryadinata, ardya.suryadinata@gmail.com

Abstract *- The diffusion concept that was first introduced by Shannon in 1949 is one of the most important cryptographic principles and is currently used as a basis in determining the cryptographic strength of symmetric algorithm, including block cipher. Algorithms that have good diffusion properties are expected to be able to withstand various cryptanalysis attacks. Currently, various kinds of criteria can be used to determine the diffusion properties of block cipher algorithms, including Avalanche Weight Distribution Criterion (AWD) which was introduced in 1999 by E. Aras and Avalanche Criterion (AC) which was introduced in 1973 by Feistel. In this study, AC and AWD were used to determine diffusion properties of RC5 algorithm, Tiny Encryption Algorithm (TEA), Vast Encryption Algorithm (VEA) and Feistel Network Cipher (CFN).*

Keywords – cryptography, symmetric, avalanche criterion, avalanche weight distribution criterion, block cipher

I – INTRODUCTION

In 1949, Shannon proposed the basic theory of symmetric cryptographic systems, including block ciphers and stream ciphers, named confusion and diffusion. Since then, confusion and diffusion have become essential characteristics of a block cipher and methods for obtaining good confusion and diffusion are the focus of block cipher designs [1]. Designing algorithms based on diffusion means that each bit of the ciphertext depends on each bit of the plaintext and each bit of the key, for example, a change in one bit of the plaintext must affect all bits of the ciphertext. Meanwhile, designing algorithms based on confusion principles means making the relationship between the ciphertext statistics and the keys very complex [1].

A test is needed to determine the nature of confusion and diffusion of the block cipher algorithm. Two of the tests that we can use are the avalanche criterion (AC) which was introduced by Feistel in 1973 and the avalanche weight distribution criterion (AWD) which was introduced by E. Aras in 1999. The concept of AC is a change of one bit of plaintext will result in a change in half of the output bits. Whereas, AWD is a simple criterion for analyzing confusion and diffusion properties of block cipher algorithms quickly and thoroughly [1]. In 2003, in his thesis, Savas Arikan [2] analyzed the propagation characteristics of the algorithms RC5, RC6, and Twofish. In his study, each algorithm was tested using AC and AWD, and in its conclusion, Arikan stated that the ciphertext generated by the RC5, RC6, and Twofish algorithms was random (having good diffusion) after the fourth round of the cipher and thus resistant to cryptanalytic attacks.

Some students of Sekolah Tinggi Sandi Negara (STSN)/National Crypto Institute designed block cipher algorithms based on the Tiny Encryption Algorithm (TEA) algorithm made by David Wheeler and Roger Needham in 1994. Some of them are the Vast Encryption Algorithm (VEA) and the Cipher Feistel Network (CFN) algorithm. The design goal of VEA was to make it a more efficient algorithm in software implementation compared to TEA [3]. Meanwhile, the design goal of CFN is to overcome the problem of block size and the key length of the TEA algorithm that had been considered not secure by NIST [4]. These three algorithms have similarities in the basic operations used and the absence of s-boxes as components of the algorithm.

In this study, we analyzed the diffusion properties of the RC5, TEA, VEA, and CFN algorithms using AC and AWD tests. We chose RC5 because it does not use the s-box as a component of the algorithm and has been tested by Savas Arikan using the same tests in his study, so the result obtained from our tests could be compared to make sure that we conducted the tests properly [2]. Meanwhile, we chose TEA, VEA, and CFN because of their similarity in design principle. Finally, we also compared the diffusion properties based on the block size of the algorithm where RC5 compared to TEA algorithm (64-bit block size) and VEA compared to CFN algorithm (128-bit block size)

II – THEORETICAL BASIS

II.1 Diffusion

Diffusion removes redundancy or patterns from plaintext by spreading it to all parts of the ciphertext [5]. Designing algorithms based on the principle of diffusion means that each bit of ciphertext depends on each bit of plaintext and each bit of the key. Diffusion will ensure the statistical properties of plaintext are spread in ciphertext so that an attacker cannot predict the plaintext related to the ciphertext in question, even after examining many similar plaintexts and corresponding ciphertexts [1].

II.2 Avalanche Criterion

In [6] explained that a function $f: \{0,1\}^n \rightarrow \{0,1\}^n$ fulfill avalanche criterion (AC) if a change in one bit of input will change in average half of the output bit, with i and $j \in (1,2,\dots,n)$ are input bit and output bit. The mathematical equation for the definition above is:

$$\frac{1}{2^n}\sum_{j=1}^{n} W\left(a_j^{e_i}\right) = \frac{n}{2} \qquad (1)$$

note: e_i is a n-bit unit vector in i^{th} position, with $i \in \{1,2,\dots,n\}$ and A^{e_i} is the avalanche vector when the i^{th} input bit changed. A^{e_i} notated as:

$$A^{e_i} = f(X) \oplus f(X \oplus e_i);\ X \in \{0,1\}$$

$$A^{e_i} = \left[a_1^{e_i} a_2^{e_i} a_3^{e_i} \dots a_n^{e_i}\right];\ a_j^{e_i} \in \{0,1\} \text{ and}$$

$$W\left(a_j^{e_i}\right) = \sum_{all\ X \in \{0,1\}^n} \left(a_j^{e_i}\right) \qquad (2)$$

is the number of changes to the j^{th}-avalanche variable against all inputs of size 2^n where $0 \leq W\left(a_j^{e_i}\right) \leq 2^n$. Using (1), we then obtained the avalanche parameter ($k_{AVAL}(i)$):

$$k_{AVAL}(i) = \frac{1}{n2^n}\sum_{j=1}^{n} W\left(a_j^{e_i}\right) = \frac{1}{2} \qquad (3)$$

The value of $k_{AVAL}(i)$ is in range of [0,1] and is the probability of overall changes from the overall output bit when the i^{th} input bit changed. In a function with a large value of n it will be hard to get the exact value of AC. Therefore, in the AC test, we use an error limit called relative error (ϵ_A):

$$\epsilon_A = \max_{1 \leq i \leq n} |2k_{AVAL}(i) - 1| \qquad (4)$$

II.3 Avalanche Weight Distribution Criterion

Avalanche Weight Distribution (AWD) is the histogram of Hamming weight vector difference of ciphertext. AWD is a simple criterion for rapid analysis of the nature of confusion and diffusion of a block cipher algorithm [1]. If the AWD criterion is used to measure the diffusion level of a block cipher, the criteria used are as follows: for plaintext pairs (P_1, P_2) that are almost the same, the Hamming histogram of the avalanche weight vector must be completely random. Thus, the AWD curve associated with all possible pairs of almost identical inputs must be distributed binomially around $n/2$ (with n is the size of the algorithm block tested) for block ciphers which have good diffusion properties [1].

In algorithms that meet good diffusion properties, the ideal binomial probability of finding j-bits changes from the n-bits of ciphertext is [2]:

$$B(j) = \frac{\binom{n}{j}}{2^n}, 0 \leq j \leq n \qquad (5)$$

The size of the deviation (D^i) is then calculated to determine the AWD curve deviation of a block cipher algorithm against the ideal binomial distribution curve $B(j)$ using N plaintext

pairs, the difference in unchanged plaintext ($\Delta P = P_1 \oplus P_2$), and the corresponding ciphertext (C_1 dan C_2). In the AWD test, the difference $\Delta C = C_1 \oplus C_2$ with weight j will then add the value of the j^{th}-element to the AWD array with a value of one. Subsequently, D^i and the resemblance parameters (R^i) can be calculated using the equation:

$$D^i = \frac{1}{2N} \sum_{j=0}^{n} |AWD(j) - NB(j)| \quad (6)$$

With i corresponds with the one-bit changes on the plaintext. The value of R^i then calculated using this equation:

$$R^i = 1 - D^i \quad (7)$$

If R^i value is equal with 1, it means that the AWD of the tested algorithm is the same as the expected ideal binomial distribution. Otherwise, if R^i value is equal to 0, it means that the AWD of the tested algorithm does not show similarities with the ideal binomial distribution.

II.4 RC5 Algorithm

RC5 is a block cipher algorithm designed by Ronald Rivest in 1994 [5]. The RC5 algorithm has three parameters, namely: *w* (word size), *r* (number of rounds), and *b* (number of bytes in key *K*). Unlike many other algorithms, RC5 does not use s-boxes and has a simple encryption process. In this study, the RC5 algorithm refers to the RC5 algorithm with the size of $w = 32$ and $b = 16$. The encryption process in the RC5 algorithm begins by dividing the plaintext into two *w*-bits registers *A* and *B*. After we have done the key expansion process, we then encrypt the plaintext as below:

A=A+S[0]; *B=B+S[1]*

For i=1 to r do:

*A=((A⊕B)<<<B)+S[2*i];*

*B=((B⊕A)<<<A)+S[2*i+1];*

The generated ciphertext is the value of *A* and *B* in the last round. One thing to remember is that one round of the RC5 algorithm will update the values *A* and *B*. Therefore, one round in RC5 can be analogous to two rounds on an algorithm based on the Feistel structure in general (for example DES).

II.5 Tiny Encryption Algorithm (TEA)

TEA is a Feistel based block cipher algorithm. TEA is designed to use minimal memory so that it can be efficiently implemented into various types of hardware and software [8]. TEA has a block size of 64 bits and a key size of 128 bits. TEA does not have a specific key schedule function. The following is the encryption process in TEA:

sum = 0; y = V[0]; *z=*V[1]; *delta =* $9e3779b9_H$

for n = 32 to n=0 do:

sum = sum + delta;

$y = y + (((z \ll 4) + K[0]) \oplus (z + sum) \oplus (z \gg 5) + K[1]))$

$z = z + ((y \ll 4) + K[2]) \oplus (y + sum) \oplus (y \gg 5) + K[3]))$

V[0] = *y;* V[1] = *Z*

The generated ciphertext is the concatenation of *V[0]* and *V[1]* in the last round.

II.6 Vast Encryption Algorithm (VEA)

VEA is a block cipher algorithm designed as TEA modification by Kholif Faiz Ma'ruf in 2008 [3]. VEA was designed as a block cipher with simple operations to overcome the efficiency of implementation issues in software and fasten the encryption process. VEA uses the Feistel network structure and has a block size of 128 bits and a key length of 256 bits. Like TEA, the operation used in the VEA algorithm is addition, XOR, and bit shift. In the process of encryption or decryption, VEA uses eleven cycles where each cycle consists of two rounds.

After being divided into four subblocks v_1,v_2,v_3,v_4, the plaintext is then encrypted as follows:

sum = 0; golden = $9e3779b9_H$

for i = 1 to i=11 do:

sum = sum + golden;

$v2 = v2 + (k4 \oplus v3 + k0 \oplus sum + (v0 \oplus k1) \ll 5)$

$v3 = v3 + (k5 \oplus v2 + k1 \oplus sum + (v1 \oplus k2) \gg 5)$

$$v0 = v0 + (k6 + v1 \oplus (k2 + sum \oplus k3 + v3) \ll 4)$$

$$v1 = v1 + (k7 + v0 \oplus (k3 + sum \oplus k0 + v2) \gg 4)$$

The ciphertext is the value of v_1,v_2,v_3,v_4, after the last cycle.

II.7 Cipher Feistel Network (CFN)

CFN is a Feistel based block cipher algorithm designed by I Made Mustika in 2010 [4]. CFN has 128-bit block size and 256-bit key length. The process of encryption and decryption of the CFN algorithm consists of seven cycles, where one cycle consists of two rounds. This algorithm is designed to overcome the problem of block size and the key length of the TEA algorithm that is considered unsafe based on the NIST standards.

For encryption or decryption, the CFN algorithm requires 56 subkeys obtained from the 256-bit input key K $(K_0^0, K_1^0, K_2^0, K_3^0, K_4^0, K_5^0, K_6^0, K_7^0)$. The input for F function of CFN is two subblocks (P_i,P_j) and two subkeys (K_i,K_j). The F function used is:

$$F(P_i, P_j, K_i, K_j) = (((P_i \ll 4) + K_i) \oplus K_j + (sum \oplus (P_j \gg 5))) \oplus sum$$

In the encryption process, $sum = sum + \delta$ with the initial value of sum is zero, and the value of δ is $9e3779b9_H$. The encryption process in the CFN algorithm begins by dividing the plaintext V input into four 32-bit sub-blocks, namely $V_0^0, V_1^0, V_2^0, V_3^0$. The four sub-blocks then become input to the following cycle transformation process:

$$V_0^{2i-1} = V_1^{2i-2}$$
$$V_1^{2i-1} = V_2^{2i-2} + F(V_1^{2i-2}, V_3^{2i-2}, K_2^i, K_7^i)$$
$$V_2^{2i-1} = V_3^{2i-2}$$
$$V_3^{2i-1} = V_0^{2i-2} + F(V_3^{2i-2}, V_1^{2i-2}, K_0^i, K_5^i)$$
$$V_0^{2i-1} = V_3^{2i-2}$$
$$V_1^{2i-1} = V_0^{2i-2} + F(V_1^{2i-2}, V_3^{2i-2}, K_1^i, K_4^i)$$
$$V_2^{2i-1} = V_1^{2i-2}$$
$$V_3^{2i-1} = V_2^{2i-2} + F(V_3^{2i-2}, V_1^{2i-2}, K_3^i, K_6^i)$$

with i is cycle 1 <i <7.

The output of the cycle transformation process is sub-block $V_0^{14}, V_1^{14}, V_2^{14}, V_3^{14}$, and the resulting ciphertext is the result of the concatenation of the four sub-blocks.

III – STUDY METHOD

III.1 Population and Sample

The RC5 and TEA have a plaintext block size of 64 bits. While the VEA and CFN have a plaintext block size of 128 bits. Thus, in this study there were two plaintext populations, the first population is the plaintext with a size of 2^{64} bits for the RC5 and TEA and the second population of 2^{128} bits for the VEA and CFN. In this study, we use a sample of 20,000 of plaintext generated with simple random sampling. In this study, we used the variables shown below:

Table 1 Variables

No	Test Name	Test object	Variables		
			Independent	Control	Dependent
1	AC	RC5	64-bit plaintext	128-bit key	64-bit ciphertext
2		TEA			
3		VEA	128-bit plaintext	256-bit key	128-bit ciphertext
4		CFN			
5	AWD	RC5	64-bit plaintext	128-bit key	64-bit ciphertext
6		TEA			
7		VEA	128-bit plaintext	256-bit key	128-bit ciphertext
8		CFN			

III.2 AC and AWD Test Procedures

In this study, the AC and AWD tests for the RC5 were carried out for the round of the algorithm or cycle for TEA, VEA, and CFN, with one cycle consisting of two rounds. The following is the AC and AWD test procedure that will be used to measure the diffusion rate of the block cipher algorithm [2]:

1. Choose a key (in this study the selected key is zero)
2. Choose plaintext P randomly and calculate plaintext P_i as its pair so that P and P_i are only different in the i^{th} or $P \oplus P_i = e_i$, where e_i is an n-bit unit vector in the i^{th}-position for $i \in \{1, 2, ..., n\}$.
3. P and P_i are then encrypted with the tested algorithm for r-round using the key selected in step 1.
4. From the generated ciphertext (C and C_i), we then calculate the avalanche vector with $\Delta C = C \oplus C_i$. We also calculate the Hamming weight of the avalanche $wt(\Delta C) = j$ where $j \in \{0,1, \ldots, n\}$..
5. Sum the avalanche vector and save it into an avalanche sum array which is an array with n elements.

6. The value of the element in the J[th] AWD array is then incremented by one, for example, *AWD [j] = AWD [j] +1* (AWD Array consists of *n* elements).

Repeat steps 2-6 for *N* times (*N* is sample size).

III.3 Avalanche Parameter Calculation

After conducting the AC test and obtaining the avalanche sum array, we then calculate the avalanche parameter ($k_{AVAL}(i)$) using equation (3) with the value $\sum_{j=1}^{n} W\left(a_j^{e_i}\right)$ is the sum of all avalanche sum array element. For more details, the value of $k_{AVAL}(i)$ is obtained by:

$$k_{AVAL}(i) = \frac{1}{nN}\sum_{j=1}^{n} avalanche\ sum\ array\ [j] \quad (8)$$

The value of the relative error ($\in_A$) of the *r*-round algorithm tested is then calculated using equation (4).

III.4 Resemblance Parameter Calculation

After conducting the AWD test, the deviation (D^i) is then calculated to determine the deviation between the curves of the AWD algorithm and the ideal binomial distribution *B(j)*. After we got the AWD array, we can calculate the deviation (D^i) and the resemblance parameter (R^i) as below [2]:

1. Draw AWD curve that corresponds to $\Delta P = e_i$
2. Calculate the binomial distribution function *B(j)* with $j \in \{0,1,2,\dots,n\}$ using equation (5).
3. Find the sum of the absolute values of the difference between *AWD[j]* dan *B(j)* for each $j \in \{0,1,2,\dots,n\}$.
4. Calculate D^i and R^i using equation (6) and (7).

III.5 Technique for Data Analyzing

In this study, we determine whether an algorithm pass the AC and AWD test by examining the values $\in_A$ and R^i. An *r*-round algorithm is concluded to have good diffusion properties if it produces a value of $\in_A$ <2% and a R^i value > 0.95 or has a deviation of <5% of the ideal R^i value. This determination is based on a study conducted by Deniz Toz et al. [7] which states that the AES algorithm passed the SAC test which is a combination of completeness and AC properties with a value of $\in_A$ 3.2%.

IV – DIFFUSION PROPERTIES ANALYSIS OF RC5, TEA, VEA, AND CFN

IV.1 Diffusion Properties Analysis of RC5

The AC and AWD test results on RC5 show characteristics that can be categorized into three intervals based on the position of changed one-bit of the plaintext at i^{th}-position, $i \in [1 \dots 35], [36 \dots 40], [41 \dots 64]$. The AC test result shows that at intervals $i \in$ [36 ... 40], RC5 has good diffusion properties since the first round. Conversely, at intervals $i \in$ [1 ... 35] and $i \in$ [41 ... 64], RC5 shows good diffusion properties after the fourth round. So, we can conclude that RC5 as a whole achieves good diffusion properties since the fourth round. We can see in Table 2 that in the fourth round RC5 passed the AC and AWD test with a value of $\in_A$ 0.84% and the maximum deviation of the R^i value was 2.71%. Whereas, for the sixteenth round (full round), RC5 has $\in_A$ 0.29% and a maximum deviation of the R^i 1.71%.

We can see the changes in $k_{AVAL}(i)$ and R^i of RC5 from one round to the next round in Table 3. From the table, we can see that $k_{AVAL}(i)$ and R^i of RC5 changes significantly from the first round to the second round.

Table 2 $\in_A$ value and maximum R^i deviation D^i of RC5 (in %)

Round	$\in_A$	D^i
1	79,70	97,74
2	38.44	62,59
3	8,08	17,38
4	**0,84**	**2,71**
5	0,23	1,58
16	0,29	1,71

Table 3 Round to round changes of k_AVAL (i) and Ri value of RC5

Round	Position of changed bit *i*	Changes of $k_{AVAL}(i)$ value (in %)			Changes of R^I value	
		Average	Minimum	Maximum	Average	Minimum
1 to 2	1-35	21,347	20,631	23,168	0,403	0,35
	36-40	0,023	0,02	0,002	0,005	0,029
	41-64	24,323	24,373	24,098	0,515	0,471
2 to 3	1-35	13,578	15,178	11,063	0,427	0,452
	36-40	0,019	0,007	0,045	0,002	0,001
	41-64	10,33	11,548	8,222	0,381	0,415
3 to 4	1-35	2,88	3,62	1,475	0,118	0,146
	36-40	0,046	0,021	0,09	0,001	0,0001
	41-64	1,491	1,79	1,079	0,059	0,068
4 to 5	1-35	0,26	0,325	0,059	0,007	0,011
	36-40	0,045	0,003	0,061	0,0006	0,001
	41-64	0,126	0,141	0,073	0,002	0,006

In Figure 2, we can see the $k_{AVAL}(i)$ curve of RC5 for the first to the fourth round. The curve shows that RC5 achieves good diffusion properties in the fourth round, with the value $k_{AVAL}(i)$ approaching the expected value in the AC test, which is 50%. Resemblance curve on Figure 1 shows that in the first round to the third round, RC5 does not have good diffusion properties where only at the interval of bit changes $i \in$ [36 ... 40] the curve forms a straight line with the R^i value approaching 1. In Figure 1 (d) it is clear that RC5 achieves good diffusion properties in the fourth round, where the resemblance curve forms a straight line with a R^i value greater than 0.97 for all bit change positions i.

Diffusion properties analysis of the components of RC5 shows that the data-dependent rotation is the most influencing component for the diffusion properties generated by RC5. This reason is why the interval of bit changes $i \in$ [36 ... 40] always produce good diffusion properties since the first round.

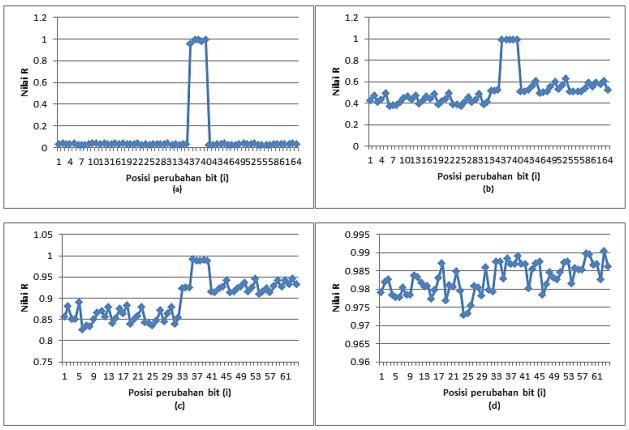

Figure 1 Resemblance curve of AWD test of RC5 with 20.000 samples. (a) first round,(b)second round,(c) third round,(d) fourth round

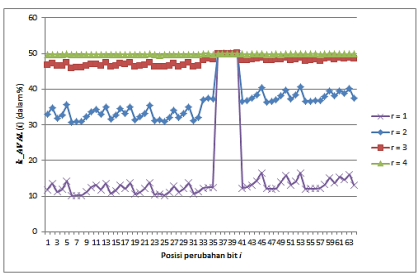

Figure 2 $k_{AVAL}(i)$ curve of RC5

IV.2 Diffusion Properties Analysis of TEA

The AC and AWD test results in TEA have characteristics that can be categorized into two intervals based on the position of changing one bit of plaintext in the 1st position, $i \in$ [1 ... 32], [33 ... 64]. From the results of the AC and AWD tests on TEA, we can conclude that TEA as a whole achieves good diffusion properties since the fifth cycle. We can see in Table 4 that in the fifth cycle of TEA passed the AC and AWD test with a value of $\in_A$ 0.19%, and the maximum deviation of the R^i value was 1.8%. Meanwhile, in the 32nd cycle or the full cycle, TEA passes the AC and AWD tests with a value of $\in_A$ 0.19% and the maximum deviation of the R^i value is 1.69%.

Table 4 $\in_A$ value and maximum R^I deviation D^I of TEA (in %)

Cycle	$\in_A$	D^i
1	87,79	100
2	59,71	99,47
3	25,83	72,30
4	2,76	8,60
5	**0,19**	**1,80**
6	0,31	1,73
32	0,19	1,69

Table 5 Cycle to cycle changes of k_AVAL (i) and Ri value of TEA

Cycle	Position of changed bit *i*	Changes of $k_{AVAL}(i)$ value (in %)			Changes of R^I value	
		Average	Minimum	Maximum	Average	Minimum
1 to 2	1-32	19,912	16,622	24,091	0,474	0,061
	33-64	20,418	14,041	28,192	0,224	0,005
2 to 3	1-32	8,485	15,509	1,206	0,421	0,602
	33-64	13,429	16,94	8,573	0,541	0,271
3 to 4	1-32	1,125	5,722	0,181	0,063	0,312
	33-64	3,764	11,536	0,15	0,209	0,636
4 to 5	1-32	0,012	0,233	0,136	0,001	0,004
	33-64	0,159	1,3	0,405	0,008	0,07
5 to 6	1-32	0,004	0,092	0,009	0,0004	0,0007
	33-64	0,011	0,019	0,019	$1{,}17 \times 10^{-5}$	0,0009

Changes in the value of $k_{AVAL}(i)$ and R^i of TEA from one cycle to the next are explained in Table 5. We can see in the table that the average value of $k_{AVAL}(i)$ changes significantly from the first cycle to the second cycle, while the average value of R^i changes significantly from the second cycle to the third cycle.

In the first to fourth cycle, the worst diffusion properties for each changed bit *i* position interval are generated when the modified bit is the most significant bit (MSB) and the less significant bit (LSB) of each algorithm subblock, i.e., 1, 32, 33, and 64. This result is following with what is stated in [8]

that the top five bits and the four bottom end bits may be weaker than the middle bits. This result is because the bits are only generated by two values of *y* or *z* when it should have been by three values, i.e., by adding *z* or *y* with another value. The uneven diffusion at the initial cycle of TEA is seen more clearly on the $k_{AVAL}(i)$ curve shown by Figure 3 and the resemblance curve shown in Figure 4, where the curve of the first and second cycles has a parabolic shape.

On the curve shown in Figure 3 and Figure 4, we can also see that in the first to the fourth cycle, changing one bit of plaintext at position $i \in$ [1 ... 32] will produce diffusion properties that are better than changes in position $i \in$ [33 ... 64]. This result shows that changing one bit of plaintext in the bit position interval $i \in$ [1 ... 32] has a greater influence on the spread of plaintext bits and produces more random ciphertext.

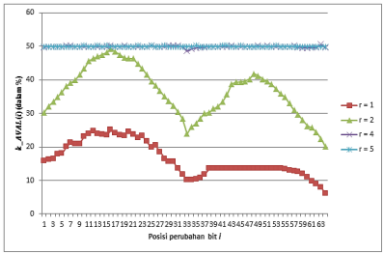

Figure 3 $k_{AVAL}(i)$ curve of TEA

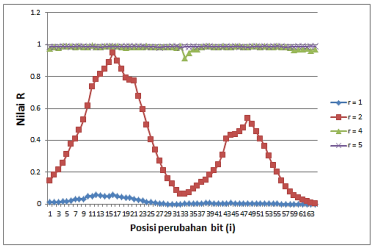

Figure 4 Resemblance curve of TEA

IV.3 Diffusion Properties Analysis of VEA

The AC and AWD test results in VEA have characteristics that can be categorized into four intervals based on the position of changing one bit of plaintext in the i^{th} position, i∈ [1 ... 32], [33 ... 64], [65 ... 96], [97 ... 128]. From the results of the AC and AWD test, we can conclude that VEA as a whole achieves good diffusion properties since the tenth cycle. We can see in Table 6 that in the tenth cycle, VEA passed the AC and AWD test with a value of ϵ_A 0.43% and the maximum deviation of the R^i value was 2.56%. Whereas, in the eleventh cycle or the full cycle, VEA passes the AC and AWD test with a value of ϵ_A 0.18% and the maximum deviation of the R^i value is 1.92%.

Table 6 ϵ_A value and maximum R^I deviation D^I of VEA (in %)

Cycle	ϵ_A	D^i
1	96,88	100
2	92,24	100
3	81,06	100
4	67,3	99,99
5	52,78	99,96
6	36,36	97,94
7	21,2	79,02
8	10,25	44,33
9	2,56	11,406
10	**0,43**	**2,56**
11	0,18	1,92

Table 7 Cycle to cycle changes of k_AVAL (i) and Ri value of VEA

Cycle	Position of changed bit *i*	Changes of $k_{AVAL}(i)$ value (in %)			Changes of R^I value	
		Average	Minimum	Maximum	Average	Minimum
1 to 2	1-32	13,04	5,888	18,321	0,03	0
	33-64	13,981	4,314	21,102	0,038	0
	65-96	15,166	6,431	20,968	0,024	$2{,}28\text{x}10^{-13}$
	97-128	11,793	2,319	20,071	0,056	0
2 to 3	1-32	13,560	6,334	17,067	0,204	$5{,}44\text{x}10^{-8}$
	33-64	13,239	7,113	15,104	0,286	$1{,}91\text{x}10^{-8}$
	65-96	13,389	6,907	17,552	0,237	$1{,}49\text{x}10^{-7}$
	97-128	10,055	5,587	12,485	0,218	$5{,}05\text{x}10^{-14}$
6 to 7	1-32	1,492	7,572	0,099	0,117	0,474
	33-64	1,508	7,632	0,094	0,118	0,48
	65-96	1,11	5,44	0,069	0,087	0,352
	97-128	2,434	7,579	0,031	0,152	0,189
10 to 11	1-32	0,011	0,01	0,0007	0,0002	0,0001
	33-64	0,001	0,004	0,031	0,0003	0,001
	65-96	0,001	0,01	0,029	0,0002	0,0001
	97-128	0,021	0,128	0,016	0,001	0,007

Changes in the value of $k_{AVAL}(i)$ and R^i of VEA from one cycle to the next are explained in Table 7. From the table, we can see that the most significant change of the average value of $k_{AVAL}(i)$ occurs during the first cycle to the second cycle, while the biggest change of the average value of the R^i occurs during the second cycle to the third cycle and during the third cycle to the fourth cycle. Although overall, there is no significant change in $k_{AVAL}(i)$ and R^i from a cycle to the next cycle. This result shows that the spread of plaintext bits in VEA occurs slowly (slow diffusion properties).

At each interval, there are 32 positions for one bit of plaintext change, which also reflects the size of VEA subblock. We can see this clearly in the $k_{AVAL}(i)$ curve shown in Figure 5 and the resemblance curve shown in Figure 6. In the $k_{AVAL}(i)$ curve and resemblance curve, we can see that in the first, third, and sixth cycles the worst diffusion of each interval is

generated when the changed bit is the MSB bit of each algorithm subblock, namely position i = 32, 64, 96, and 128. This result is due to the arrangement of operations in VEA, which causes the effect of the addition operation to be greater than bit shift operations and XOR.

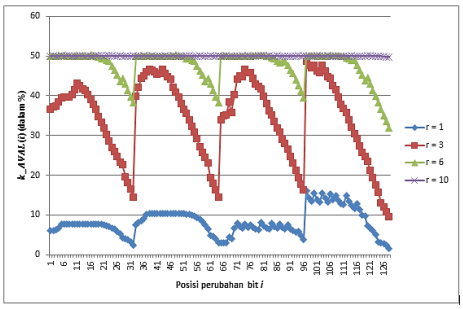

Figure 5 $k_{AVAL}(i)$ curve of VEA

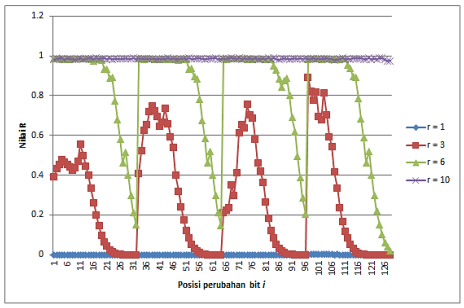

Figure 6 Resemblance curve of VEA

IV.4 Diffusion Properties Analysis of CFN

The AC and AWD test results in CFN have characteristics that can be categorized into four intervals based on the position of changing one bit of plaintext in the i^{th} position, $i \in$ [1 ... 32], [33 ... 64], [65 ... 96], [97 ... 128]. From the results of the AC and AWD test, we can conclude that overall, CFN achieves good diffusion properties since the seventh cycle, which is also a full cycle of CFN. In Table 8, we can see that in the seventh cycle, CFN passed the AC test with a value of ϵ_A 0.48%. Based on the results of the AWD test, in the seventh cycle, CFN passed the AWD test with a deviation of the maximum R^i value of 2.15%.

Changes in the value of $k_{AVAL}(i)$ and R^i of CFN from one cycle to the next are explained in Table 9. We can see that the most significant change in the average value of $k_{AVAL}(i)$ occurs during the second cycle to the third cycle, while the most the average value of R^i of CFN occurs during the third cycle to the fourth cycle. Although, there is no significant change in the value of $k_{AVAL}(i)$ and R^i from a cycle to the next cycle. This result shows that diffusion in CFN occurs slowly.

Table 8 ϵ_A value and maximum R^i deviation D^i of CFN (in %)

Cycle	ϵ_A	D^i
1	95,12	100
2	84,79	100
3	58,95	99,71
4	28	75,16
5	11,93	41,89
6	2,29	9,23
7	**0,48**	**2,15**
8	0,18	2,06

Table 9 Cycle to cycle changes of k_AVAL (i) and Ri value of CFN

Cycle	Position of changed bit i	Changes of $k_{AVAL}(i)$ value (in %)			Changes of R^i value	
		Average	Minimum	Maximum	Average	Minimum
1 ke 2	1-32	13,932	7,874	16,044	0,037	$5,83 \times 10^{-6}$
	33-64	11,935	5,619	16,775	0,006	$5,48 \times 10^{-11}$
	65-96	13,775	7,751	15,855	0,037	$5,83 \times 10^{-6}$
	97-128	11,726	5,161	16,622	0,005	$1,49 \times 10^{-11}$
2 ke 3	1-32	15,362	16,144	15,797	0,337	0,055
	33-64	15,853	12,934	16,639	0,183	0,003
	65-96	15,27	17,137	14,333	0,329	0,065
	97-128	16,06	12,921	17,087	0,184	0,002
5 ke 6	1-32	0,678	3,513	0,193	0,044	0,243
	33-64	1,367	5,041	0,343	0,094	0,339
	65-96	0,628	1,785	0,211	0,04	0,126
	97-128	1,447	4,47	0,267	0,1	0,282
6 ke 7	1-32	0,073	0,525	0,057	0,003	0,034
	33-64	0,214	0,807	0,004	0,011	0,059
	65-96	0,057	0,445	0,062	0,003	0,03
	97-128	0,226	0,907	0,012	0,012	0,07

As in the TEA and VEA, there are 32 positions in each of these different one-bit changes in plaintext interval that also reflect the size of CFN subblock. More clearly can be seen in the $k_{AVAL}(i)$ curve shown in Figure 7 and the resemblance curve shown in Figure 8.

On the $k_{AVAL}(i)$ and resemblance curves, we can see that in the first, third and fifth cycles the worst diffusion properties in each interval are generated when the bits are MSB bits of each algorithm subblock, i.e., 32, 64, 96, and 128. Like VEA, this result is caused by the addition operation, which has a considerable influence on the diffusion properties of the algorithm. However, in contrast to VEA, diffusion generated in CFN is more evenly distributed because of the use of left-shift bit operation in each subblock.

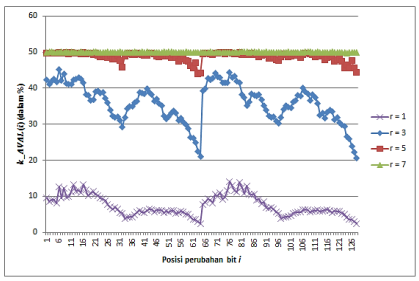

Figure 7 $k_{AVAL}(i)$ curve of CFN

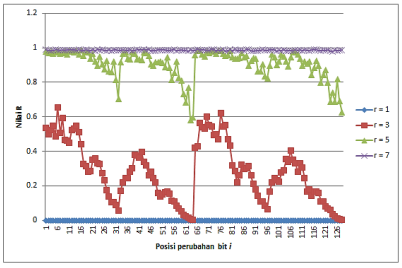

Figure 8 Resemblance curve of CFN

IV.5 Diffusion Properties Comparison of RC5 and TEA

To compare the diffusion properties of RC5 and TEA, as explained above in Part III, one round of RC5 is analogous to two rounds or one cycle of Feistel-based algorithms, such as TEA, VEA, and CFN. Therefore, in this study, one round of RC5 was considered equivalent to two rounds or one cycle of TEA. For uniformity, we will use the term iteration instead of the round and cycle for the following discussion.

Based on the value of $\in_A$ (shown in Table 2 and Table 4) of RC5 and TEA, we can conclude that in each iteration, the spread of plaintext bits into the ciphertext bits in RC5 occurs faster. This property is indicated by the smaller value of $\in_A$, in the first iteration, which is 79.7% for RC5 compared to 87.8% for TEA. For the full iteration (16 rounds for RC5 and 32 cycles for TEA) TEA has better diffusion properties with a value of $\in_A$ 0.2% and the maximum deviation of the R^i value 1.69%.

Comparison of diffusion properties of RC5 and TEA based on the AC and AWD test results can be seen in the form of the $k_{AVAL}(i)$ curve shown in Figure 9 and the resemblance curve shown in Figure 10. In all four curves, we can see different characteristics of diffusion properties based on $k_{AVAL}(i)$ value, R^i value, and bit change position interval in RC5 and TEA. The $k_{AVAL}(i)$ and the resemblance curve of RC5 tend to form a straight line while the $k_{AVAL}(i)$ curve and the resemblance curve of TEA form two parabolic curves which each one describes the bit change position interval.

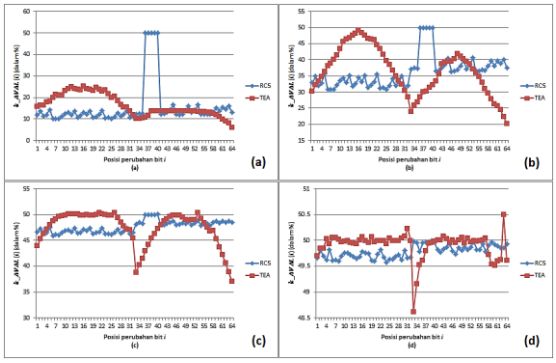

Figure 9 Comparison of the $k_{AVAL}(i)$ curve of RC5 and TEA (a) first iteration (b) second iteration (c) third iteration (d) fourth iteration

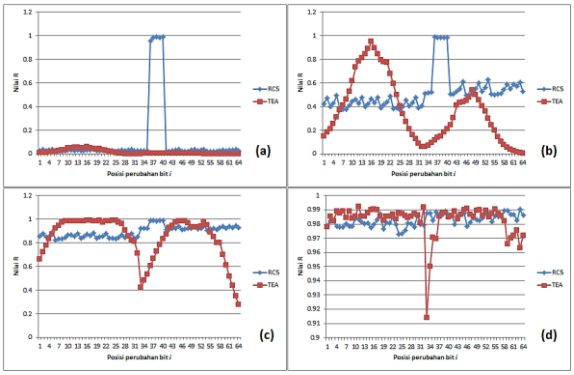

Figure 10 Comparison of Ri curves of RC5 and TEA (a) first iteration (b) second iteration (c) third iteration (d) fourth iteration

IV.6 Diffusion Properties Comparison of VEA and CFN

Based on the value of $\in_A$ of VEA and CFN it is known that in each cycle, the spread of plaintext bits into the ciphertext bits in CFN occurs more quickly. Table 8 to Table 13 show that in the first cycle, VEA and CFN have the same value of $\in_A$. The difference in the value of $\in_A$ in VEA and CFN starts to appear clearly in the second cycle with CFN has $\in_A$ value 84.8%, while VEA has $\in_A$ value 92.24%. In the full cycle (11 cycles for VEA and seven cycles for CFN) VEA produces better diffusion properties with a value of $\in_A$ 0.18% and a maximum deviation of the R^i value of 1.92%.

For more details, the comparison of the diffusion properties of VEA and CFN based on the AC and AWD test results can be seen in the form of the $k_{AVAL}(i)$ curve shown in Figure 11 and the resemblance curve shown in Figure 12. From the curves shown in the Figures, we can see the similarity of diffusion properties based on the value of $k_{AVAL}(i)$, the value

of R^i, and the position interval of bit changes in VEA and CFN. Just as in TEA, the $k_{AVAL}(i)$ curve and the resemblance curve of VEA and CFN have a parabolic shape that describes a bit interval change position.

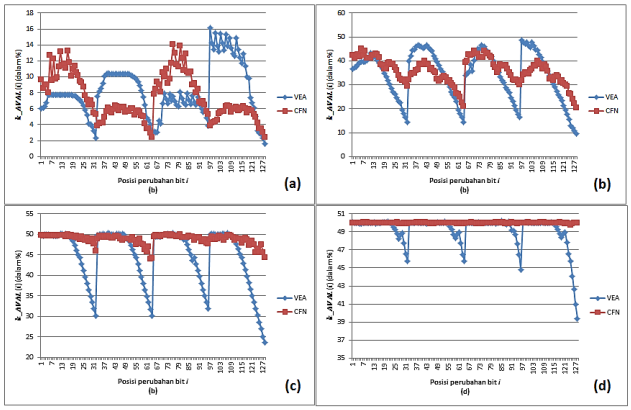

Figure 11 Comparison of $k_{AVAL}(i)$ curve of VEA and CFN algorithm on (a) first cycle (b) third cycle, (c) fifth cycle, and (d) seventh cycle

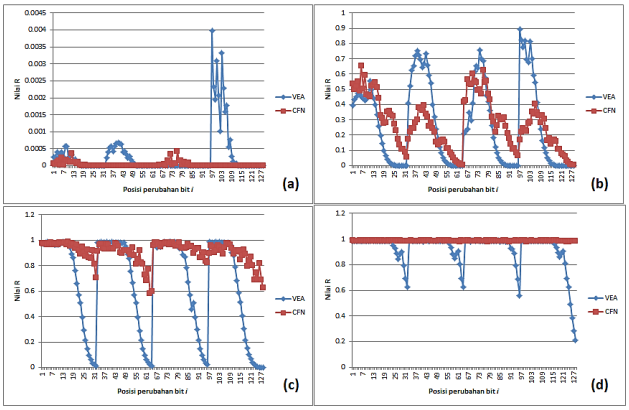

Figure 12 Comparison of R^i curve of VEA and CFN algorithm on (a) first cycle (b) third cycle, (c) fifth cycle, and (d) seventh cycle

V – CONCLUSION

Based on the study conducted, we draw the following conclusions:

1. RC5 achieves good diffusion properties since the fourth round, TEA since the fifth cycle, VEA since the tenth cycle, and CFN since the seventh cycle which is also the full cycle of the algorithm.
2. Comparison of the AC and AWD test results between RC5 and TEA shows that one iteration of RC5 has faster diffusion properties than TEA.
3. Comparison of the AC and AWD test results between the VEA and CFN shows that one cycle of the CFN has faster diffusion properties compared to VEA.

REFERENCES

[1] Aras, Ekrem dan Yucel, Melek D. 2001. Performance Evaluation of SAFERK-64 and S-boxes of SAFER Family. Turk J Elec Engin . 2001. VOL.9 NO.2.

[2] Arikan, Savas. 2003. *Propagation Characteristics of RC5, RC6, and Twofish Ciphers.*Tesis. Ankara. Graduate School of Natural and Applied Sciences. Middle East Technical University.

[3] Ma'ruf, Kholif F. 2008. *Desain Algoritma Block cipher VEA.* Bogor. Sekolah Tinggi Sandi Negara.

[4] Mustika, I Made. 2010. *Desain Algoritma CFN.* Bogor. Sekolah Tinggi Sandi Negara.

[5] Schneier, Bruce. 1996. Applied Cryptography: Protocols, Algorithms, and Source Code in C. Second Edition. New York: John Wiley & Sons, Inc.

[6] Vergili, Isil dan Yucel, Melek D., 2001. Avalanche and Bit Independence Properties for the Ensembles of Randomly Chosen n x n S-Boxes. Turk J Elec Engin . 2001. VOL.9 NO.2.

[7] Toz, Deniz et al. 2006.*Statistical Analysis of Block Ciphers.* Uygulamah Matematik Enstitusu.

[8] Wheeler, David J. dan Needham, Roger M. 1994. *A Tiny Encryption Algorithm*. Computer Laboratory of Cambridge University.