ANALISIS ALGORITMA RC4 SEBAGAI METODE ENKRIPSI WPA-PSK PADA SISTEM KEAMANAN JARINGAN WIRELESS LAN

Vivien Septyaningtyas Arintamy[1], Niken Dwi Wahyu Cahyani [2], Asep Mulyana [3]
[1.2]Departemen Elektro dan Komunikasi, Fakultas Teknik Elektro, Universitas Telkom Bandung
vivien111090110@gmail.com, niken.cahyani@mymail.unisa.edu.au, asepmulyana@telkomuniversity.ac.id

ABSTRAK

Wireless Fidelity atau WiFi adalah teknologi yang digunakan sebagai *Wireless Local Area Network* atau WLAN. Salah satu kendala kompetensi pada WLAN adalah kerentanan terhadap aspek keamanan jaringan. WPA-PSK adalah salah satu metode sistem keamanan jaringan *wireless* yang sudah banyak digunakan dan menggunakan algoritma RC4. Kelemahan algoritma RC4 yang paling dikenal adalah *Bit Flipping Attack* atau BFA yang bertujuan untuk mengetahui sebagian atau keseluruhan *plaintext* dari *ciphertext* tanpa harus mengetahui kunci.

Diajukan CRC-32 bit sebagai teknik untuk memberikan nilai pada *plaintext* sebelum dilakukan enkripsi. Penambahan CRC-32 pada algoritma RC4 bertujuan untuk memperkuat plaintext pada saat proses enkripsi. Hasil enkripsi yang berupa ciphertext lalu akan dikirim ke penerima dan di tengah perjalanan akan dilakukan BFA oleh attacker dengan mengubah 1 bit dari ciphertext dari 0 ke 1 atau sebaliknya. Pada proses dekripsi akan dilakukan pengecekkan kembali apakah data yang diterima rusak atau tidak dengan cara membangkitkan kembali kunci dan pengecekkan nilai CRC cocok atau tidak.

Dari hasil pengujian yang dilakukan terhadap RC4, penambahan CRC pada *plaintext* sebelum dilakukan enkripsi berhasil meningkatkan ketahanan RC4 terhadap serangan *Bit Flipping Attack*. Kualitas dari RC4 yang sudah dimodifikasi CRC dipengaruhi oleh *avalanche effect* dan waktu komputasi saat eksekusi. Dengan menggunakan *plaintext* yang sama dengan kunci yang berbeda, serta menggunakan *plaintext* yang berbeda dengan kunci yang sama, masing-masing menghasilkan nilai *avalanche effect* yang sama baiknya, yaitu berkisar antara 45% sampai dengan 60% namun ada beberapa yang menghasilkan nilai AE kurang dari 45%, dipengaruhi oleh jumlah karakter yang berbeda, nilai key dan seed yang diinputkan dan banyaknya percobaan. Akan tetapi, kelemahan pada metode yang diajukan ini adalah waktu komputasi yang dibutuhkan untuk enkripsi / dekripsi menjadi lama dibandingkan RC4 tanpa penambahan CRC-32.

Kata Kunci : RC4, *Bit Flipping Attack*, CRC, Enkripsi, Dekripsi, *Avalanche Effect*.

ABSTRACT

Wireless Fidelity or WiFi technology is used as a Wireless Local Area Network or WLAN. One of the obstacles is the competence of the WLAN vulnerability to network security aspects. WPA-PSK is one of the methods of wireless security system that is already widely used and uses the RC4 algorithm. The weakness of the RC4 algorithm is the best known Bit Flipping Attack or BFA. BFA attack aimed to find part or all of the plaintext from the ciphertext without knowing the key.

It's proposed CRC-32 bits as a technique to assign a value to the plaintext prior to encryption. Addition of CRC-32 on the RC4 algorithm aims to strengthen the plaintext during the encryption process. Ciphertext encrypted form will then be sent to the recipient. Then in testing, the wireless media, the attacker did Bit Flipping Attack (BFA) by changing one bit of the ciphertext from 0 to 1 or 1 to 0, during the process of delivery. In the decryption process will be checking again whether the data received is damaged or not by reviving the keys and checking CRC values match or not.

From the results of tests performed on RC4, the addition of CRC in plaintext before encrypting managed to increase resistance to attack RC4 Bit Flipping Attack. The quality of the modified CRC RC4 is affected by the avalanche effect and the execution time of computing time. By using the same plaintext with different keys, as well as using different plaintext with the same key, each value of the avalanche effect produces equally good, ranging from 45% to 60% but there is a few result that give AE under 45 %, because of the different characters and key and seed that added. In RC4 with CRC-32, computation time for encryption / decryption become longer than RC4 without CRC-32 adding.

Keywords: *RC4, Bit Flipping Attack, CRC, Encryption, Decryption, Avalanche Effect.*

1. PENDAHULUAN

1.1 Latar Belakang

Kemajuan teknologi dari waktu ke waktu telah banyak mengubah pola pikir dan pola hidup manusia. Salah satunya adalah teknologi jaringan. Teknologi jaringan memungkinkan pertukaran informasi baik berbentuk data maupun suara antar perangkat, mereduksi tingkat kompleksitas point-to-point dan menekan biaya komunikasi. Karena berbagai kemudahan yang diberikan itulah yang membuat komunikasi jaringan kurang aman. Banyak penyerang sering melakukan aksinya dengan memanfaatkan teknologi jaringan yang cukup membahayakan. Demi mencegah hal ini, dibuatlah metode keamanan

jaringan yang saat ini sudah mengalami banyak perkembangan. Salah satu yang paling dikenal adalah metode WPA-PSK.

Wi-Fi Protected Access – Pre Shared Key atau WPA-PSK merupakan pengembangan dari sistem keamanan jaringan sebelumnya, yaitu WEP. WPA-PSK sendiri menggunakan metode enkripsi dengan algoritma RC4. Menurut penelitian sebelumnya di Institut Teknologi Bandung, algoritma RC4 adalah algoritma kunci simetris mudah diimplementasikan, banyak digunakan oleh user dan memiliki kecepatan waktu yang lebih unggul dibandingkan dengan metode enkripsi lainnya [11]. Meskipun begitu, algoritma RC4 juga memiliki beberapa kelemahan, diantaranya adalah retan terhadap Bit Flipping Attack. Bit Flipping Attack atau BFA adalah merupakan serangan pada algoritma stream cipher dengan tujuan untuk mengubah hasil enkripsi dengan cara mengubah bit ciphertext tertentu yang tentu saja dapat menurunkan performa kinerja RC4 sebagai metode enkripsi.

Pada tugas akhir ini, akan di lakukan pengujian mengenai tingkat ketahanan RC4, contoh serangan BFA pada RC4 dan performasi saat melakukan proses enkripsi sebelum dan sesudah dilakukan BFA. Dan juga perhitungan CRC (dalam hal ini menggunakan CRC-32 bit) yang digunakan pada saat enkripsi dan perhitungan rata-rata waktu komputasi yang dibutuhkan RC4 untuk proses enkripsi serta Avalanche Effect (AE) untuk mengetahui parameter yang digunakan untuk mengetahui perubahan bit dengan membandingkan ciphertext hasil enkripsi dari dua buah input plaintext atau kunci yang hanya berbeda 1 bit. Dan kemudian dilakukan analisis ketahanan algoritma RC4 sebelum diaplikasikan pada WPA-PSK sebagai sistem keamanan jaringan wireless LAN.

1.2 Tujuan

Adapun tujuan dari tugas akhir ini adalah untuk menguji dan mangamati sistem kerja algoritma RC4 dalam melakukan proses enkripsi menggunakan Java Netbeans sebagai alat bantu dalam melakukan pengujian. Menguji ketahanan algoritma RC4 dengan mekakukan contoh serangan Bit Flipping Attack dan menganalisis performasi algoritma RC4. Dan untuk menghitung rata-rata waktu yang dibutuhkan algoritma RC4 dalam melakukan proses enkripsi yang berpengaruh pada tingkat keamanan WPA-PSK sebagai sistem keamanan jaringan nirkabel.

1.3 Rumusan Masalah

Rumusan masalah dari tugas akhir ini adalah untuk :

1. Menganalisis algoritma RC4 pada WPA-PSK sebagai sistem keamanan jaringan WLAN dan rata-rata waktu komputasi yang dibutuhkan algoritma RC4 untuk melakukan proses enkripsi.
2. Untuk mengetahui ketahanan algoritma RC4 yang rentan akan serangan Bit Flipping, dimana attacker atau penyerang mengubah ciphertext untuk menghasilkan perubahan yang dapat diprediksi oleh plaintext. Hal ini dapat mempengaruhi kinerja WPA-PSK sebagai sistem keamanan jaringan

1.4 Batasan Masalah

Adapun batasan masalah pada tugas akhir ini adalah :

1. Analisis proses enkripsi dan dekripsi algoritma RC4 dengan proses enkripsi dan dekripsi algoritma RC4 dengan penambahan CRC-32 sebagai penanganan Bit Flipping Attack atau BFA.
2. Percobaan penyerangan BFA di lakukan seolah-olah data di serang oleh attacker dengan mengubah 1 bit pada data pada proses enkripsi yang dilakukan oleh user di server lokal atau localhost.
3. Perhitungan Avalanche Effect atau AE pada percobaan dengan menginputkan plaintext yang sama dengan key yang berbeda dan plaintext yang berbeda dengan key yang sama. Nilai AE yang baik berkisar antara 45 % — 60 %.

2. Landasan Teori

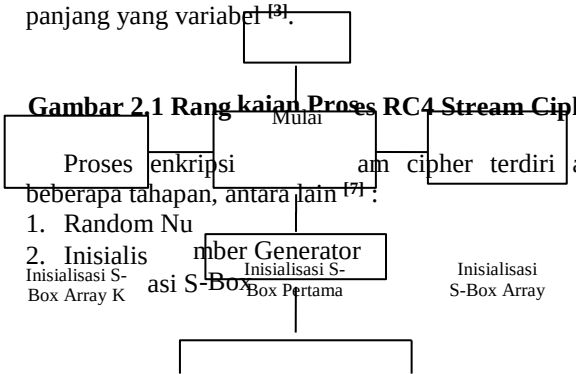
2.1 Algoritma RC4

Algoritma Rivest Code (RC4) merupakan salah satu algoritma kunci simetris yang dibuat oleh RSA Data Security Inc (RDADSI) yang berbentuk stream cipher. Algoritma stream cipher adalah Algoritma kriptografi beroperasi pada plainteks/cipherteks dalam bentuk bit tunggal, yang dalam hal ini rangkaian bit dienkripsikan/didekripsikan bit per bit [8]. RC4 menggunakan panjang key dari 1 sampai 256 byte yang digunakan untuk menginisialisasi tabel sepanjang 256 byte. Tabel ini dugunakan untu generasi berikut dari pseudorandom yang menggunakan XOR dengan plaintext untuk menghasilkan ciphertext [13]. Masing-masing elemen dalam tabel saling ditukarkan minimal sekali. RC4 merupakan salah satu jenis stream cipher sehingga RC4 memproses unti atau input data, pesan atau informasi pada satu saat. Unit atau data pada umumnya sebuah byte atau bahkan dalam bit (byte dalah hal RC4) sehingga dengan cara ini enkripsi atau dekripsi dapat dilaksanakan pada panjang yang variable [2].

Tidak seperti cipher aliran yang memproses data dalam bit, RC4 memproses data dalam ukuran byte (1 byte = 8 bit). Untuk membangkitkan aliran kunci, *chipper* menggunakan status internal yang terdiri dari dua bagian :

1. Permutasi angka 0 sampai 255 di dalam larik S_0, $S_{1, \dots}$, S_{255}. Permutasi merupakan fungsi dari kunci U dengan panjang variable.
2. Dua buah pencacah indeks, i dan j [6]

Algoritma RC4 menggunakan dua buah S-Box yaitu array sepanjang 256 yang berisi permutasi dari bilangan 0 sampai 255, dan S-Box kedua, yang berisi permutasi yang merupakan fungsi dari kunci dengan panjang yang variabel [3].



Gambar 2.1 Rangkaian Proses RC4 Stream Cipher

Proses enkripsi am cipher terdiri atas beberapa tahapan, antara lain [7] :

1. Random Nu
2. Inisialis mber Generator
   asi S-Box

Pengacakkan S-

Pembuatan Pseudo-Random Byte

3. Menyimpan Key Dalam Key Byte Array
4. Permutasi Pada S-Box
5. Pengoperasian XOR untuk menghasilkan ciphertext atau plaintext.

2.1.1 Mekanisme Kerja RC4

1 Penginisialisasi Array State

Dalam penginisialisasian *state-array*, terdapat 2 *statearray* yang harus diinisialisasi, S dan K. Array S sebesar 256 bit diinisialisasi dengan angka dari 0 sampai dengan 255. Sedangkan array K sebesar 256 bit diisi dengan key dengan panjang 1-256 bit secara berulang sampai seluruh array K terisi penuh. Setelah itu, dilakukan *Key Scheduling-Algorithm* untuk menghasilkan permutasi dari array S berdasarkan key yang tersedia.

2. Penghasil Kunci Enkripsi dan Pengenkripsian

Setelah memiliki state array yang telah teracak, maka kita akan menginisialisasi kembali i dan j dengan 0. Setekah itu, kita lakukan pseudo-random generation algorithm atau PRGA untuk menghasilkan kunci enkripsi (dalam hal ini cipher_byte) yang akan di XOR-kan dengan plainteks. Untuk menghasilkan kunci enkripsi, PRGA meng-*incremen* i, menambahkan nilai S[i] dan S[j] menukar nilai keduanya, dan nilai kunci yang dihasilkan adalah S dengan indeks yang sama dengan jumlah S[i] dan S[j] di-modulo dengan 256 [4].

2.1.2 Key Scheduling Algoritma (KSA)

Algoritma key scheduling digunakan untuk menginisialisasi permutasi di array "S". panjang kunci didefinisikan sebagai jumlah byte di kunci dan mempunyai rentang panjang kunci dari 1 sampai 256, khususnya antara 5-16 tergantung dari panjang kunci 40- 128bit. Pertama-tama array "S" diinisialisasi untuk identitas permutasi. S kemudian diproses ke 256 iterasi dengan cara yang sama dengan PRGA utama, tapi juga dikombinasikan dalam byte dari kunci dalam waktu yang bersamaan.

2.1.3 Pseudo-Random Generation Algoritma (PRGA)

PRGA (Pseudo-Random Generation Algortihm) memodifikasi state dan output sebuah byte dari keystream. Hal ini penting karena banyaknya dibutuhkan iterasi. Dalam setiap iterasi, PRGA menginkremen **i**, menambahkan nilai S yang ditunjuk oleh **i** sampai **j,** kemudian menukar nilai S[i] dan S[j], lalu mengembalikan elemen dari S di lokasi S[i] + S[j] (modulo 256). Setiap elemen S ditukar dengan elemen lainnya paling tidak satu kali setiap 256 iterasi.

2.2 Bit Flipping Attack (BFA)

Bit Flipping Attack merupakan serangan pada algoritma stream cipher dengan tujuan untuk mengubah hasil dekripsi dengan cara mengubah bit ciphertext tertentu. Proses pengubahan tersebut dilakukan dengan melakukan proses flip (membalikkan) bit tertentu pada ciphertext. Maksud dari proses flip tersebut adalah mengubah 0 jadi 1 atau bit 1 jadi 0.

Kelemahan ini dapat dimanfaatkan pada bit flipping attack yang dijelaskan sebagai berikut [1] :

1. Attacker melakukan sniffing terhadap jaringan
2. Attacker melakukan intersepsi terhadap paket yang dienkripsi menggunakan WPA
3. Attacker melakukan serangan bit-flipping terhadap paket tersebut dan melakukan modifikasi cheksum (ICV)
4. Attacker mengirimkan frame yang telah dimodifikasi kepada receiver
5. Receiver (client maupun access point) menerima frame dan melakukan perhitungan ulang ICV berdasarkan pada konten frame
6. Receiver melakukan proses dekapsulasi frame dan memproses paket layer 3
7. Karena bit pada layer 3 ter-flip, proses checksum pada layer 3 gagal
8. Stack ip pada receiver membangkitkan pesan error yang mudah diprediksi
9. Pesan error di-enkripsi dan dikirim oleh receiver kepada sender
10. Attacker melakukan sniffing untuk menangkap pesan error yang ternekripsi
11. Berdasarkan pada pesan error yang diketahui dan pesan error yang terenkripsi, attacker dapat menurunkan keystream

2.3 Cyclic Redudancy Check (CRC)

CRC (Cyclic Redudancy Check) adalah teknik untuk mendeteksi kesalahan-kesalahan di dalam data digital, tetapi tidak bisa memperbaiki kesalahan apabila terdeteksi [15]. CRC mempunyai beberapa varian tergantung pada bilangan polinom yang digunakan dalam proses komputasinya, misal CRC-8, CRC-16, CRC-32 dan CRC-64. Varian tersebut yang menentukan jumlah total bit yang dapat diproses untuk menghasilkan nilai CRC.

Data yang hendak ditransmisikan atau disimpan ke sebuah media penyimpanan rentan sekali mengalami kesalahan. Untuk memastikan integritas data yang hendak ditransmisikan atau disimpan [16].

Dalam CRC-32, berarti menggunakan poli 32 bit (4 byte). Jika melakukan operasi dalam basis byte bukan bit, maka poli yang digunakan akan dioperasikan dalam bentuk byte, sehingga harus mempunyai panjang kelipatan 8 bit (1 byte).

3. Pemodelan Dan Perancangan Sistem

3.1 Analisis Kebutuhan Sistem

Pada perangkat lunak yang dibutuhkan pada tugas akhir ini, program dapat bekerja di berbagai macam platform dan sistem operasi. Dibutuhkan Java Development Kit untuk menjalankan program tugas akhir tersebut. Dan nantinya program yang akan digunakan berbentuk ekstensi *.jar.

3.2 Analisis Kebutuhan Fungsional Sistem

Yang dimaksud dengan kebutuhan fungsional adalah kebutuhan yang berkaitan dengan fungsi atau proses transformasi atau perubahan yang dilakukan oleh sistem.

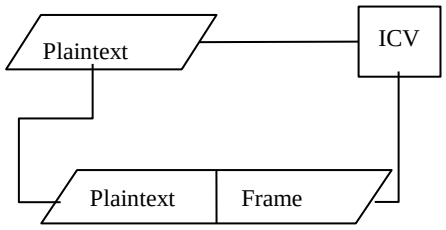
a. User dapat memilih file yang akan dijadikan input file dengan ukuran 0 sampai dengan 102400 bit

untuk dilakukan enkripsi / dekripsi. Lalu user memasukkan nilai seed 1 byte (8 bit) sebagai input awal sistem yang menghasilkan Initialization Vector 3 byte (24 bit) dan secret key 13 byte (104 bit) dan kemudian dibangkitkan dan akan digunakan sebagai key.

b. Sistem akan membaca file input per byte
c. Sistem akan melakukan perhitungan CRC untuk plaintext sebanyak dua kali, yaitu sebelum dilakukan proses enkripsi dan sebelum dikirim ke penerima
d. Sistem melakukan enkripsi / dekripsi dengan algoritma RC4, yang kemudian menghasilkan file yang berupa ciphertext / plaintext. Selanjutnya sistem akan melakukan perhitungan waktu komputasi atau lamanya waktu proses eksekusi
e. Sistem melakukan Bit Flipping Attack (BFA) yaitu merusak ciphertext dengan mengubah 1 bit-nya misalkan dari 0 menjadi 1 atau sebaliknya. Selanjutnya sistem akan melakukan perhitungan nilai CRC terbaru.

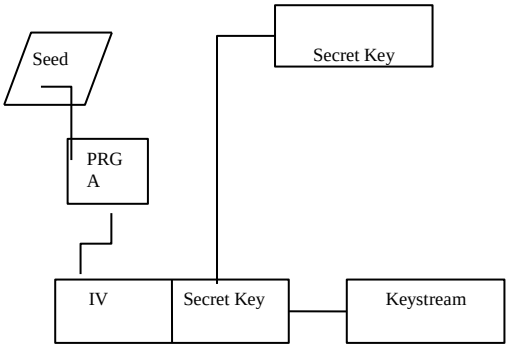
3.3 Perancangan Sistem

Langkah pertama yang dilakukan adalah proses penyisipan CRC-32 bit pada plaintext sebelum dilakukan enkripsi, untuk kunci input nilai seed yang berfungsi untuk membangkitkan Initialization Vector (IV). IV ini akan digunakan bersama secret key sebagai keystream

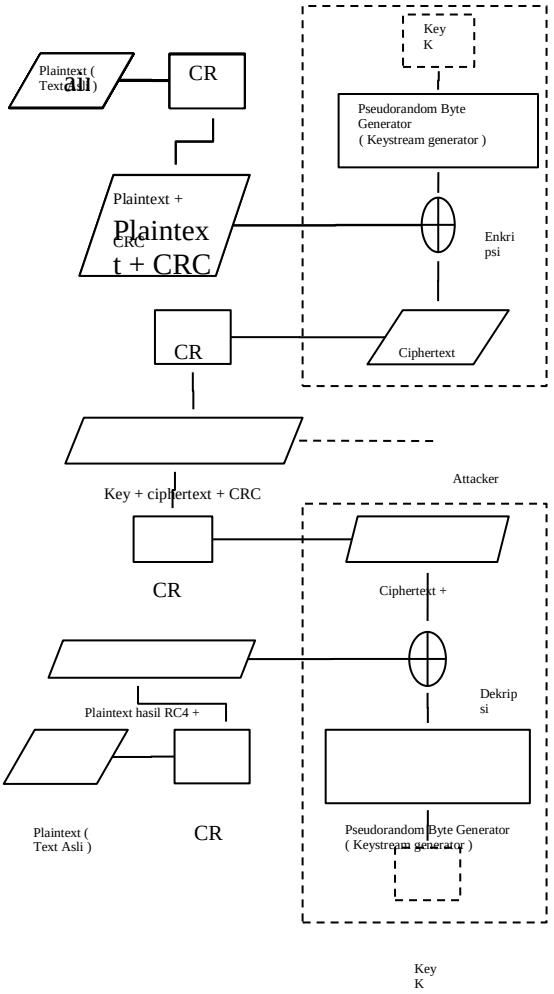


Gambar 3.1 Skema Proses ICV Pada Plaintext

Pada gambar di bawah ini, seed adalah nilai awal yang digunakan untuk menghasilkan IV. User memasukkan secret key yang akan digabungkan dengan IV dan berubah menjadi keystream untuk proses enkripsi. Seed dan secret key yang tidak di enkripsi akan dikirim ke penerima sebagai key generator.



Gambar 3.2 Skema Pembuatan Kunci RC4



Gambar 3.3 Gambaran Umum Sistem RC4 Enkripsi dan Dekripsi

4. Hasil dan Analisa

4.1 Pengujian

4.1.1 Pengujian Skenario 1

Tabel 4.1 Hasil Pengujian Pengecekkan Ciphertext

No	Data Uji	Key	Cek CRC	Dekripsi	Cek CRC
1.	01.txt	berhasil	cocok	berhasil	cocok
2.	02.txt	berhasil	cocok	berhasil	cocok
3.	03.txt	berhasil	cocok	berhasil	cocok
4.	04.txt	berhasil	cocok	berhasil	cocok
5.	05.txt	berhasil	cocok	berhasil	cocok
6.	06.txt	berhasil	cocok	berhasil	cocok
7.	07.txt	berhasil	cocok	berhasil	cocok
8.	08.txt	berhasil	cocok	berhasil	cocok
9.	09.txt	berhasil	cocok	berhasil	cocok
10.	10.txt	berhasil	cocok	berhasil	cocok

Dari hasil tabel 4.1 di atas terbukti bahwa algoritma RC4 tahan terhadap serangan *Bit Flipping Attack* selama proses enkripsi

4.1.2 Pengujian Skenario 2

Tabel 4.2 Hasil Pengujian AE Dengan Key Sama Plaintext Berbeda Pada RC4

4

5

No	Plaintext	Key	Avalanche Effect (%)
1.	01A.txt	87350	32%
2.	02A.txt		
3.	03A.txt	77552	41%
4.	04A.txt		
5.	05A.txt	11111	50%
6.	06A.txt		
7.	07A.txt	00000	79%
8.	08A.txt		
9.	09A.txt	08090	83%
10	10A.txt		

Tabel 4.3 Hasil Pengujian AE Dengan Key Sama Plaintext Berbeda Pada RC4+CRC-32

No	Plaintext	Key	Avalanche Effect (%)
1.	01A.txt	87350	97 %
2.	02A.txt		
3.	03A.txt	77552	42 %
4.	04A.txt		
5.	05A.txt	11111	57 %
6.	06A.txt		
7.	07A.txt	00000	98 %
8.	08A.txt		
9.	09A.txt	08090	76 %
10	10A.txt		

Dari hasil tabel 4.2 dan tabel 4.3 diatas menunjukkan nilai *Avalanche Effect* pada proses enkripsi RC4 dengan RC4+CRC-32 menghasilkan beberapa AE yang ideal dan tidak ideal.

Tabel 4.4 Hasil Pengujian AE Dengan Plaintext Sama Key Berbeda Pada RC4

No	Plaintext	Key	Avalanche Effect (%)
1	01B.txt	87350	33%
		87351	
2	02B.txt	77552	40%
		77553	
3	03B.txt	11111	64%
		11112	
4	04B.txt	00000	46%
		00001	
5	05B.txt	08090	71%
		08091	

Tabel 4.5 Pengujian AE Dengan Plaintext Sama Key Berbeda Pada RC4+CRC-32

No	Plaintext	Key	Avalanche Effect (%)
1	01B.txt	87350	73 %
		87351	
2	02B.txt	77552	80 %
		77553	
3	03B.txt	11111	69 %
		11112	
4	04B.txt	00000	92 %
		00001	
5	05B.txt	08090	60 %
		08091	

Dari hasil tabel 4.4 dan tabel 4.5 diatas menunjukkan nilai *Avalanche Effect* pada proses enkripsi RC4 dengan RC4+CRC-32, dengan plaintext sama dan key yang berbeda sama baiknya.

Gambar 4.1 Grafik Perbedaan Nilai AE Pada RC4

Gambar 4.2 Grafik Perbedaan Nilai AE Pada RC4+CRC-32

Dari grafik gambar 4.1 dan gambar 4.2 diatas menunjukkan bahwa *Avalanche Effect* yang di hasilkan dari percobaan skenario 2 di atas, ada masih dalam range AE yang ideal, ada pula yang nilainya di luar nilai AE ideal.

Tabel 4.6 Hasil Pengujian Waktu Eksekusi Algoritma RC4

No	Jumlah Karakter	Waktu Eksekusi (Dalam Second)	
		Enkripsi	Dekripsi
1	512	0,107	0,003
2.	1024	0,031	0,008
3.	10240	0,223	0,268
4.	20480	0,793	1,108
5.	30720	1,884	2,354
6.	40960	3,386	4,928
7.	50200	4,702	5,117
8.	61440	6,897	8,459
9.	71680	9,061	9,612
10.	81920	12,323	12,276
11.	92760	16,653	17,054
12.	102400	19,598	19,825

Tabel 4.7 Hasil Pengujian Waktu Eksekusi Algoritma RC4+CRC-32

No	Jumlah Karakter	Waktu Eksekusi (Dalam Second)	
		Enkripsi	Dekripsi
1	512	0,120	0,016
2.	1024	0,046	0,023
3.	10240	0,238	0,283
4.	20480	0,808	1,123
5.	30720	1,899	2,369
6.	40960	3,403	4,945
7.	50200	4,719	5,134
8.	61440	6,914	8,476
9.	71680	9,078	9,629
10.	81920	12,340	12,293

11.	92760	16,673	17,074
12.	102400	19,618	19,845

Dari tabel 4.6 dan tabel 4.7 di atas menunjukkan waktu komputasi untuk algoritma RC4 lebih cepat dibandingkan dengan RC4 dengan penambahan CRC-32.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Dari hasil pengujian algoritma RC4 pada tugas akhir ini, dapat diambil kesimpulan bahwa :

1. Dengan penambahan CRC-32 pada RC4 dapat menguatkan plaintext saat proses enkripsi dan teruji dapat memperkuat ketahanan algoritma RC4 terhadap serangan *Bit Flipping Attack*. Namun hal ini juga berpengaruh pada waktu komputasi RC4 dalam melakukan proses enkripsi dan dekripsi. Pada RC4, rata-rata waktu komputasi untuk enkripsi adalah 6,304833 detik dan untuk dekripsi adalah 6,751 detik. Sedangkan untuk RC4 dengan penambahan CRC-32, waktu komputasi yang dibutuhkan untuk enkripsi adalah 6,3213333 detik dan untuk dekripsi adalah 6,7675 detik. Dengan penambahan CRC-32 pada RC4, waktu komputasinya lebih lama dibanding RC4.
2. Nilai *Avalanche Effect* yang dihasilkan, baik dengan plaintext sama dan key berbeda maupun plaintext berbeda dengan key sama, pada RC4 dengan penambahan CRC-32 menunjukkan ada beberapa yang masih dalam batas *Avalanche Effect* yang ideal, yaitu 45 % - 60 % dan ada juga yang nilainya di luar batas AE ideal. Hal ini dipengaruhi oleh nilai key dan seed yang diinputkan, jumlah karakter yang berbeda setelah dilakukan proses enkripsi dan banyaknya percobaan yang dilakukan.
3. Pada *Bit Flipping Attack*, *attacker* tidak perlu mengetahui nilai key yang diinputkan untuk proses enkripsi. Dengan mengubah 1 bit saja pada data akan menghasilkan nilai AE yang berbeda.

5.1 Saran

Saran untuk tugas akhir berikutnya yang berkaitan dengan RC4 meliputi :

1. Menganalisis dan menguji algoritma RC4 sebagai metode enkripsi dengan serangan yang berbeda dengan penambahan metode yang berbeda.
2. Menganalisis dan menguji algoritma RC4 sebagai metode enkripsi pada jenis file yang berbeda, misalkan image (*.jpg, *.jpeg, *.bmp, *.png, dll), audio (*.mp3, *.wav, dll), video (*.mp4, *.flv, *.mpeg, *.cam, dll)

DAFTAR PUSTAKA

[1] http://www.cisco.com/en/us/docs/solutions/enterprise/mobility/design/guide Diakses pada tanggal 2 Oktober 2012.

[2] Ariyanto, Yuri. 2009. Algoritma RC4 Dalam Proteksi Transmisi Dan Hasil Query Untuk ORDBMS POSTGRESQL. Fakultas Teknologi Informatika Institut Teknologi Adhitama Surabaya, Jurnal Informatika Vol.10.No.1.

[3] Bastari, A Thoriq Abrawi. 2010. Analisis Perbandingan Stream Cipher RC4 Dan SEAL. Makalah IF3058 Kriptografi Institut Teknologi Bandung.

[4] Cristian, Glen. Studi Terhadap Jaringan Nirkabel Dengan Protokol Keamanan WEP. Teknik Informatika Institut Teknologi Bandung.

[5] Dogan, Mithat C. 2004. On Security Issues In Wireless Communication System. Diakses pada tanggal 7 Juli 2013

[6] Nicholas, R.K dan Lekkas, P.C. 2002. Wireless Security : Model, Threats, And Solutions. Penerbit The McGraw-Hill Companies Publisher. New York, Chicago, San Fransisco, Lisbon, London, Madrid, Mexico City, Milan, New Delhi, San Juan, Seoul, Singapore, Sidney, Toronto.

[7] Puspitasari, A. 2012. Penanganan Bit Flipping Attack (BFA) Pada Sistem Kriptografi RC4. Tugas Akhir. Program Studi Teknik Informatika Institut Teknologi Telkom Bandung.

[8] Munir, Rinaldi. 2004. Tipe Dan Mode Algoritma Simetri. Program Studi Teknik Informatika Institut Teknologi Bandung.

[9] Maryono, Slamet. 2003. Keamanan Jaringan Komputer RC4 Stream Cipher. Diakses pada tanggal 7 Juli 2013.

[10] Passa, Fitriana. 2010. Studi Analisis Implementasi Algoritma RC4 Dengan Modifikasi Key Menggunakan Fungsi SHA-1. Makalah IF3058 Kriptografi Institut Teknologi Bandung.

[11] Suryani, Karina Novita. 2009. Algoritma RC4 Sebagai Metode Enkripsi. Program Studi Teknik Informatika Institut Teknologi Bandung.

[12] Sutiono, Arie Pratama. 2010. Algoritma RC4 Sebagai Perkembangan Metode Kriptografi. Program Studi Teknik Informatika Institut Teknologi Bandung.

[13] W, I.Y.B Aditya Eka Prabowo. Analisis Perbandingan Dan Pengujian Rivest Cipher 4 Dan Ciphersaber 2. Program Studi Teknik Informatika Institut Teknologi Bandung.