Implementasi Keamanan Data Keuangan di SMK Swasta Musda Perbaungan Menggunakan Metode RC4

Rista[1], Arjon Samuel Sitio[2]

[1,2] Program Studi Teknik Informatika, STMIK Pelita Nusantara Jl. St. Iskandar Muda No. 1 Medan, Indonesia
[1] kakarista.96@gmail.com; [2] arjonsitio@gmail.com

INFORMASI ARTIKEL

Kata Kunci:
Enkripsi, RC4, XOR.

ABSTRAK

Keamanan dan kerahasian data saat ini menjadi isu yang sangat penting dan terus berkembang. Beberapa kasus menyangkut keamanan data saat ini menjadi suatu pekerjaan yang membutuhkan biaya penanganan dan pengamanan yang sedemikian besar. Untuk menjaga keamanan dan kerahasiaan pesan, data, atau informasi agar tidak dapat di baca atau di mengerti oleh sembarang orang, kecuali untuk penerima yang berhak, maka dirancang aplikasi sistem pengaman data dengan metode enkripsi menggunakan algoritma rc4. RC4 (Rivest Cipher 4) adalah sebuah synchrone streamcipher, yaitu cipher yang memiliki kunci simetris dan mengenkripsi plainteks secara digit per digit atau byte per byte dengan cara mengkombinasikan dengan operasi biner (biasanya XOR) dengan sebuah angka semi acak.

Keywords:
Enkripsi, RC4, XOR.

ABSTRACT

Data security and confidentiality is currently a very important issue and continues to grow. Several cases concerning data security are currently a job that requires handling and security costs that are so large. To maintain the security and confidentiality of messages, data, or information so that no one can read or understand it, except for the rightful recipients, a data security system application with an encryption method using the rc4 algorithm is designed. RC4 (Rivest Cipher 4) is a synchrone streamcipher, which is a cipher that has a symmetric key and encrypts the plaintext digit by digit or byte per byte by combining binary operations (usually XOR) with a semi-random number.

I. Pendahuluan

Saat ini dengan seiring perkembangan teknologi sekarang menjadi kebutuhan penting bagi setiap orang dikarenakan setiap orang membutuhkan sebuah privasi.

Pada algoritma simetris ada beberapa metode yang dapat digunakan untuk melakukan pengamanan data diantaranya: Data Encryption Standard, RC2, RC4, Advanced Encryption Standard, On Time Pad dan lain sebagainya. Rivest Code 4 (RC4) merupakan salah satu stream chipher sehingga dapat memproses unit atau input data, pesan atau informasi pada satu saat (Irwansyah, 2018). RC4 digunakan secara luas pada beberapa aplikasi dan umumnya dinyatakan sangat aman, sebab kerahasiaan kunci harus dijaga dan dikirim di saluran komunikasi yang aman(Yuliansyah & Juliasari, 2018).

Kriptografi (cryptography) berasal dari bahasa Yunani: "cryptos" yang artinya "secret" (rahasia) dan "graphein" yang artinya "writing" (tulisan). Jadi kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan (cryptography is the art and science of keeping message secure)(Munir,2016).

SMK Swasta Musda Perbaungan merupakan Sekolah Menengah Kejuruan Swasta yang ada di Kota Perbaungan. Saat ini SMK Swasta Musda perbaungan melakukan pembayaran SPP dengan cara manual belum adanya sistem keuangan khususnya pembayaran SPP sehingga sering terjadi kesalahan dalam pencatatan transaksi oleh karena itu dibangun sebuah sistem keuangan.

Berdasarkan uraian tersebut diatas maka dibuatlah judul skripsi "Implementasi Keamanan Data Keuangan di SMK Swasta Musda Perbaungan Menggunakan Metode RC4".

II. Metode

A. Kriptografi

Kriptografi berasal dari Bahasa Yunani, crypto dan Graphia, Crypto berarti rahasia dan Graphia artinya menulis. Menurut terminologi, kriptografi adalah ilmu dan seni untuk memelihara keamanan pesan saat pesan dikirim, dari satu tempat ke temat lain. Teknik untuk data enkripsi(kriptografi) diterapkan pada data dan informasi, dilakukan dengan menyandikan atau menyembunyikan data asli(Zendrato et al., 2019).

B. Enkripsi dan Dekripsi

Enkripsi adalah proses yang dilakukan untuk mengubah file yang tidak rusak pesan (teks biasa) kedalam bentuk yang tidak terbaca (teks chip),dekripsi adalah proses yang dilakukan untuk mengubah pesan yang tidak dapat dibaca menjadi bentuk yang bisa dibaca dan dimengerti. Enkripsi dan Proses dekripsi diatur oleh satu atau lebih kriptografi kunci(Dhany et al., 2018).

C. Tujuan kriptografi

Dari paparan awal dapat dirangkumkan bahwa kriptografi bertujuan untuk member layanan keamanan. Yang dinamakan aspek-aspek keamanan:

1. Kerahasiaan (confidentiality) Adalah layanan yang ditujukan untuk menjaga agar pesan tidak dapat dibaca oleh pihak-pihak yang tidak berhak.
2. Integritas data (data integrity) Adalah layanan yang menjamin bahwa pesan masih asli atau belum pernah dimanipulasi selama pengiriman.
3. Otentikasi (authentication) Adalah layanan yang berhubungan dengan identifikasi, baik mengidentifikasi kebenaran pihakpihak yang berkomunikasi (user autehentication).
4. Non-repudiation Adalah layanan untuk menjaga entitas yang berkomunikasi melakukan penyangkalan(Pabokory et al., 2016).

III. Hasil dan Pembahasan

Algoritma kriptografi Rivest Code 4 (RC4) merupakan salah satu algoritma kunci simetris dibuat oleh RSA Data Security Inc (RSADSI) yang berbentuk stream chipper(Nugroho et al., 2016). Algoritma RC4 (Rivest code 4) ditemukan pada tahun 1987 oleh Ronald Rivest dan menjadi simbol keamanan RSA, digunakan secara luas pada beberapa aplikasi dan umumnya dinyatakan sangat aman(Yuliansyah & Juliasari, 2018). Dalam metode ini terdapat 5 langkah yang harus dilakukan untuk diterapkan pada sebuah data yang ingin diamankan(Abdurrahman et al., 2018), diantaranya :

1. **Langkah ke 1 :** Inisialisasi larik S: $S_0 = 0$, $S_1 = 1$, …, $S_{255} = 255$
2. **Langkah ke 2 :** Menentukan nilai kunci Jika panjang kunci $U < 256$, lakukan *padding* sehingga panjang kunci menjadi 256 *byte*. Contoh U =“abc” (3 byte), maka lakukan padding U = “abcabc….” hingga U mencapai 256 byte.
3. **Langkah ke 3 :** Lakukan permutasi nilai-nilai di dalam larik S
4. **Langkah ke 4 :** Bangkitkan aliran-kunci
5. **Langkah Akhir :** Lakukan enkripsi

Algoritma RC4 yang merupakan algoritma blok kode yang bisa dibilang cukup cepat, dengan parameter sebagai berikut:

1. Ukuran Blok (32/64/128 bit)
2. Panjang Kunci masukan dari user (0-2040 bit)
3. Jumlah putaran (0-255 bit)(Pengamanan et al., 2018).

Berikut adalah perhitungan manual algoritma RC4 dengan mode 5 Bit dengan plaintext **RISTA** dan kunci **25731.** Untuk penggunaan kunci dapat juga menggunakan Huruf dan dapat menggunakan mode bit 0-2040 bit.

a. Proses Ekspansi Kunci (*Key Expansion*)

Langkah pertama dalam algoritma ini adalah Inisialisasi S-Box dan kunci seperti berikut :

Tabel 4.1 Inisialisasi S-Box dan Key Inisilaisasi S-box (S)

0	**1**	**2**	**3**	**4**	**...**	**...**	**...**	**...**	**255**
S(0)	S(1)	S(2)	S(3)	S(4)	...	…	...	…	S(255)

Key S-box (K)

2	**5**	**7**	**3**	**1**	**...**	**...**	**...**	**...**	**W**
K(0)	K(1)	K(2)	K(3)	K(4)	...				(Kn)

Pembentukan tabel diatas berdasarkan pseudocode-nya, berikut pseudocode-nya :

For i from 0 to 255
S[i] = i
K[i] = K[i mod length]
Endfor

Kemudian melakukan proses pencarian nilai Array S yang terakhir dengan melakukan perulangan dengan cara menjumlahkan inisialisasi S-Box dengan inisialisasi kunci kemudian hasil keduanya di mod dengan jumlah bit plaintext yang diinputkan serta menginisialisasikan i dan j adalah 0 agar dapat menghitung pseudo-code random (r). dengan pseudcode dan perhitungan manual sebagai berikut :

```
i = 0; j = 0;
for i = 0 to 255 {
j = (j + S[i] + K[i]) mod 255
Swap S[i] dan S[j] }
```

Iterasi pertama
For i = 0; j : 0
J = (j + S(i) + K(i)) mod 5
= (j + S(0) + (K0)) mod 5
= (0 + 0 + 2) mod 5
= 2
Swap S[0] dan S[2] maka menjadi :

2	**1**	**0**	**3**	**4**	**...**	**...**	**...**	**...**	**255**
S(0)	S(1)	S(2)	S(3)	S(4)	...	…	...	…	S(255)

Iterasi kedua
For i = 1; j : 2
J = (j + S(i) + K(i)) mod 5
= (j + S(1) + (K1)) mod 5
= (2 + 1 + 5) mod 5
= 3
Swap S[1] dan S[3] maka menjadi :

2	**3**	**0**	**1**	**4**	**...**	**...**	**...**	**255**
S(0)	S(1)	S(2)	S(3)	S(4)	…	...	…	S(255)

Iterasi ketiga
For i = 2; j: 3
J = (j + S(i) + K(i)) mod 5
= (j + S(2) + (K2) mod 5
= (3 + 0 + 7) mod 5
= 0
Swap S[2] dan S[0] maka menjadi :

0	**3**	**2**	**1**	**4**	**...**	**...**	**...**	**...**	**255**
S(0)	S(1)	S(2)	S(3)	S(4)	...	…	...	…	S(255)

Iterasi keempat
For i = 3; j = 0
J = (j + S(i) + K(i)) mod 5
= (j + S(3) + (K3) mod 5
= (0 +1 + 3) mod 5
= 4
Swap S[3] dan S[4] maka menjadi :

0	**3**	**2**	**4**	**1**	**...**	**...**	**...**	**...**	**255**
S(0)	S(1)	S(2)	S(3)	S(4)	...	…	...	…	S(255)

Iterasi kelima
For i = 4; j = 4
J = (j + S(i) + K(i)) mod 5
= (j + S(4) + (K4) mod 5
= (4+ 1 + 1) mod 5

= 1

Swap S[4] dan S[1] maka menjadi :

1	**3**	**2**	**4**	**0**	**...**	**...**	**...**	**...**	**255**
S(0)	S(1)	S(2)	S(3)	S(4)	...	…	...	…	S(255)

Setelah didapatkan hasil array S yang terakhir lalu langkah selanjutnya adalah meng-*XOR* kan plaintext sebanyak jumlah karakter plaintext itu sendiri, plaintext yang ditentukan adalah **RISTA** yang terdiri dari 5 Karakter yaitu R –I – S – T – A dan array S yang dipakai adalah array S yang terakhir. Adapun inisialisasi nya yaitu :

Inisialisasi i dan j dengan i = 0; j = 0.

Untuk kunci K[0] :

i = (i + 1) mod 5

i = (0 + 1) mod 5

= 1 mod 5

= 1

j = (j + S(i)) mod 5

j = (0 + S[1])) mod 5

= (0 + 4) mod 5

= 4

Swap S(1) dan S(4) maka menjadi :

1	**0**	**2**	**4**	**3**	**…**	**...**	**...**	**...**	**255**
S(0)	S(1)	S(2)	S(3)	S(4)	…	…	...	…	S(255)

K[0] = S[(S[i] + S[j]) Mod 5]
= S((S[1] + S[4]) Mod 5]
= S((0 + 4) Mod 5)
= S(4 Mod 5)
= S(4)
= 3

K[0] = 00000011

Untuk kunci K[1] :

i = (i + 1) mod 5

i = (1 + 1) mod 5

= 2 mod 5

= 2

j = (j + S[i]) mod 5

j = (4 + S[2]) mod 5

= (4 + 2) mod 5

= 1

Swap S(2) dan S(1) maka menjadi :

1	**2**	**0**	**4**	**3**	**...**	**...**	**...**	**...**	**255**
S(0)	S(1)	S(2)	S(3)	S(4)	…	…	...	…	S(255)

K[1] = S[(S[i] + S[j]) Mod 5]
= S[(S[2] + S[1]) Mod 5]
= S[(0 + 2) Mod 5]
= S(2 Mod 5)
= S(2)
= 0

K[1] = 00000000

Untuk kunci K[2] :

i = (i + 1) mod 5

i = (2 + 1) mod 5

= 3 mod 5

= 3

j = (j + S[i]) mod 5

j = (1 + S[3]) mod 5

= (1 + 4) mod 5

= 0

Swap S(3) dan S(0) maka menjadi :

0	**2**	**1**	**4**	**3**	**...**	**...**	**...**	**...**	**255**
S(0)	S(1)	S(2)	S(3)	S(4)	…	…	...	…	S(255)

K[2] = S[(S[i] + S[j]) Mod 5]
= S[(S[3] + S[0]) Mod 5]
= S[(4 + 0) Mod 5]
= S(4 Mod 5)
= S(4)
= 3
K[2] = 00000011

Untuk kunci K[3] :

i = (i + 1) mod 5
i = (3 + 1) mod 5
= 4 mod 5
= 4
j = (j + S[i]) mod 5
j = (0 + S[4]) mod 5
= (0 + 3) mod 5
= 3

Swap S(4) dan S(3) maka menjadi :

0	**2**	**1**	**3**	**4**	**...**	**...**	**...**	**...**	**255**
S(0)	S(1)	S(2)	S(3)	S(4)	…	…	...	…	S(255)

K[3] = S[(S[i] + S[j]) Mod 5]
= S[(S[4] + S[3]) Mod 5]
= S[(4 +3) Mod 5]
= S(7 Mod 5)
= S(2)
= 1
K[3] = 00000001

Untuk kunci K[4] :

i = (i + 1) mod 5
i = (4 + 1) mod 5
= 5 mod 5
= 0
j = (j + S[i]) mod 5
j = (3 + S[0]) mod 5
= (3 + 0) mod 5
=3

Swap S(0) dan S(3) maka menjadi :

3	2	1	0	4	...	...	...	...	255
S(0)	S(1)	S(2)	S(3)	S(4)	…	…	...	…	S(255)

K[4] = S[(S[i] + S[j]) Mod 5]
= S[(S[0] + S[3]) Mod 5]
= S[(3 + 0) Mod 5]
= S(3 Mod 5]
= S(3)
= 0
K[4] = 00000000

Berdasarkan perhitungan *key expassion* diatas, maka didapatkan kunci untuk proses enkripsi dan deskripsi adalah :

Tabel 4.2 Index kunci enkripsi dan deskripsi

Index Kunci	**Biner Kunci**
K[0]	00000011

K[1]	00000000
K[2]	00000011
K[3]	00000001
K[4]	00000000

a. Proses Enkripsi

Proses enkripsi dimulai dengan mengkonversi plaintext ke biner, karena biner dari masing-masing plaintext inilah nantinya yang akan di XOR dengan masing-masing kunci, untuk lebih jelasnya bisa dilihat seperti berikut ini :

Plaintext : **R**
Biner Plaintext : 01010010
Key [0] : 000000011
Biner Chipertext : 01010001
Chipertext : S
Plaintext : **I**
Biner Plaintext : 01001001
Key [1] : 00000000
Biner Chipertext : 01001001
Chipertext : I
Plaintext : **S**
Biner Plaintext : 01010011
Key [2] : 00000011
Biner Chipertext : 01010000
Chipertext : P
Plaintext : **T**
Biner Plaintext : 01010100
Key [3] : 00000001
Biner Chipertext : 01010101
Chipertext : U
Plaintext : **A**
Biner Plaintext : 01000001
Key [4] : 00000000
Biner Chipertext : 01000001
Chipertext : A

Sehingga dari proses diatas didapatkan chipertext adalah **SIPUA.**

a. Proses deskripsi

Sedangkan proses deskripsi adalah kebalikan dari proses enkripsi, yaitu mengubah ciphertext menjadi plaintext kembali. Untuk lebih jelas dapat dilihat seperti berikut ini :

Chipertext : S
Biner Chipertext : 01010001
Key [0] : 00000011
Biner Plaintext : 01010010
Plaintext : **R**
Chipertext : I
Biner Chipertext : 01001001
Key [1] : 00000000
Biner Plaintext : 01001001
Plaintext : **I**
Chipertext : P
Biner Chipertext : 01010000
Key [2] : 00000011
Biner Plaintext : 01010011
Plaintext : **S**
Chipertext : U
Biner Chipertext : 01010101
Key [3] : 00000001
Biner Plaintext : 01010100
Plaintext : **T**
Chipertext : A

Biner Chipertext : 01000001
Key [4] : 00000000
Biner Plaintext : 01000001
Plaintext : **A**
Sehingga dari proses diatas didapatkan chipertext adalah **RISTA.**

IV. Kesimpulan

Berdasarkan penelitian yang dilakukan mengenai penerapan metode RC4 untuk mengamankan data keuangan di SMK Musda Perbaungan, maka dapat diambil kesimpulan sebagai berikut :

1. Menjalankan rencana atau rancangan yang telah dibuat dalam keamanan data keuangan di SMK Musda Perbaungan.
2. Penerapan metode RC4 dalam keamanan data keuangan di SMK Musda Perbaungan dapat mempermudah Kepala sekolah melakukan keamanan data keuangan dan menjaga kerahasian data dari orang lain.
3. Dengan menggunakan aplikasi keamanan data ini orang yang tidak di kehendaki yang ingin mengetahui tentang informasi data keuangan di SMK Musda Perbaungan tidak akan mudah mengetahui informasi data keuangan tersebut .
4. Pengguna harus *login* terlebih dahulu, dan untuk input data atau bendahara harus mengetahui kunci private yang di berikan oleh kepala sekolah.

Daftar Pustaka

Abdurrahman, F. N., Informatika, T., Informasi, F. T., Luhur, U. B., Utara, P., & File, E. O. (2018). *Aplikasi Pengamanan File Email Dengan Metode Kriptografi Rivest Code 4 (Rc4) Dan Steganografi End of File (Eof) Pada Pt . Pratama Indomitra Konsultan*. *1*(3), 1143–1147.

Dhany, H. W., Izhari, F., Fahmi, H., Tulus, M., & Sutarman, M. (2018). *Encryption and Decryption using Password Based Encryption, MD5, and DES*. *141*(ICOPOSDev 2017), 278–283. https://doi.org/10.2991/icoposdev-17.2018.57

Irwansyah, D. (2018). Pengamanan Data Teks dengan Algoritma Modifikasi RC4. *Pelita Informatika Budi Darma*, *17*, 55–58.

Munir, R. (2016). Pengantar Kriptografi. *Departemen Teknik Informatika Institut Teknologi Bandung*, *Buku*, 12.

Nugroho, N. B., Azmi, Z., & Arif, S. N. (2016). Aplikasi Keamanan Email Menggunakan Algoritma Rc4. *Jurnal SAINTIKOM*, *15*(3), 81–88. https://lppm.trigunadharma.ac.id/public/fileJurnal/hpO91 Jurnal Nurcahyo.pdf

Pabokory, F. N., Astuti, I. F., & Kridalaksana, A. H. (2016). Implementasi Kriptografi Pengamanan Data Pada Pesan Teks, Isi File Dokumen, Dan File Dokumen Menggunakan Algoritma Advanced Encryption Standard. *Informatika Mulawarman : Jurnal Ilmiah Ilmu Komputer*, *10*(1), 20. https://doi.org/10.30872/jim.v10i1.23

Pengamanan, A., Keuangan, D., Desktop, B., & Algoritma, M. (2018). *Aplikasi Pengamanan Database Keuangan Berbasis Dekstop Menggunakan Algoritma RC4 Dan Vigenere*. *1*(1), 237–242.

Yuliansyah, A., & Juliasari, N. (2018). Aplikasi kriptografi menggunakan algoritma rc4 dan des untuk mengamankan pesan email. *Skanika*, *1*(2), 491–497.

Zendrato, N., Zarlis, M., Efendi, S., Barus, E. S., Sulindawaty, & Fahmi, H. (2019). Increase Security of IoT Devices Using Multiple One Time Password. *Journal of Physics: Conference Series*, *1255*(1). https://doi.org/10.1088/1742-6596/1255/1/012030