Journal of Informatics Management and Information Technology
ISSN 2774-4744 (Media Online)
Vol 1, No 4, Oktober 2021
Hal 144-152
https://hostjournals.com/jimat

Modifikasi Algoritma XTEA dengan Pembangkitan Kunci Menggunakan Metode Linear Congruential Untuk Pengamanan File Dokumen

Berliana Oktaviani Sinaga, Sinar Sinurat, Taronisokhi Zebua

Fakultas Ilmu Komputer dan Teknologi Informasi, Prodi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia
Email: [1]berlianaoktavianisinaga@gmail.com

Abstrak−File dokumen yang bersifat rahasia tentu harus memiliki keamanan baik saat proses file itu dikirimkan atau disimpan kedalam media stroge. Hal ini dilakukan agar file dokumen rahasia tidak diketahui seseorang yang memang tidak memiliki wewenang atas file tersebut, atau menghindari kerugian apabila file tersebut jatuh ketangan yang salah. Demi mewujudkan hal tersebut, maka dibutuhkanyalah sebuah teknik pengaman file dokumen yang dapat merubah data file dokumen menjadi data yang tidak dikenali. Salah satu teknik yang dapat digunakan adalah kriptografi. Kriptografi memanfaatkan algoritma untuk menyandikan file dokumen dengan proses enkripsi, dan mengembalikan file dokumen dengan proses dekripsi. Salah satu algoritma yang dapat digunakan adalah algoritma XTEA. XTEA membagi string file dokumen menjadi 2 bagian yaitu AL 32 bit dan BL 32 bit. Kelemahan dari algoritma XTEA pada saat melakukan proses enkripsi atau dekripsi dengan panjadwalan kunci yang berulang sehingga tidak efektif dalam mengoptilkan keamanan dari file dokumen hasil enkripsi, oleh sebab itu pada penelitian ini ditambahkanya metode Linear Congruential Generator (LCG) untuk membangkitkan kunci XTEA. LCG memiliki nilai a,c,m dan X0 yang menjadi pemicu untuk mendapatkan kunci XTEA. Sehingga pada proses pendistribusian file dokumen hasil enkripsi, kunci yang dikirimkan adalah nilai pemicu LCG berupa a,c,m dan X0, apabila nilai tersebut diketahui oleh pihak attacker, maka file dokumen hasil enkripsi tidak dapat dikembalikan. Sehingga dari hasil enkripsi algoritma XTEA dengan pengoptimal kunci berdasarkan metode LCG menghasilkan cipher file dokumen yang lebih optimal keamananya.

Kata Kunci: Kriptografi; XTEA; LCG; File Dokumen

Abstract−Confidential document files must have security both when the file process is sent or stored in the stroge media. This is done so that the confidential document file is not known by someone who does not have authority over the file, or to avoid loss if the file falls into the wrong hands. In order to achieve this, a document file security technique is needed that can convert document file data into unrecognized data. One of the techniques that can be used is cryptography. Cryptography makes use of algorithms to encode document files with an encryption process, and restore document files with a decryption process. One of the algorithms that can be used is the XTEA algorithm. XTEA divides the document file string into 2 parts, namely AL 32 bit and BL 32 bit. The weakness of the XTEA algorithm when carrying out the encryption or decryption process with repeated key scheduling so that it is not effective in optimizing the security of encrypted document files, therefore in this study the Linear Congruential Generator (LCG) method is added to generate XTEA keys. LCG has values a, c, m and X0 which triggers to get the XTEA key. So that in the process of distributing encrypted document files, the key sent is the LCG trigger value in the form of a, c, m and X0, if the value is known by the attacker, the encrypted document file cannot be returned. So that the results of the XTEA encryption algorithm with key optimizer based on the LCG method produce a document file cipher that is more optimal in security.

Keywords: Cryptography; XTEA; LCG; Document Files

1. PENDAHULUAN

Keamanan dan kerahasiaan sebuah informasi maupun dokumen berharga menjadi suatu perhatian yang sangat penting pada zaman modern saat ini. Berbagai macam teknik saat ini banyak digunakan untuk melindungi informasi yang rahasia agar orang yang tidak memiliki hak tidak dapat mengakses informasi tersebut, maka diperlukan suatu cara untuk mengamankan data dan informasi. Salah satunya adalah dengan cara merubah data tersebut ke dalam bentuk data yang lain yang tidak dapat dimengerti dalam bentuk penyandian data dengan teknik kriptografi. Data yang digunakan untuk proses pembagian informasi dapat berbentuk berbagai macam, seperti teks, gambar, tabel, maupun grafik. Data file teks yang digunakan kerap dijadikan sebuah file yang dapat dikirimkan untuk mendapat suatu informasi atau membagikan informasi tersebut kepada orang lain. Salah satu teknik yang umum digunakan untuk mengamankan data adalah teknik kriptografi.

Kriptografi merupakan salah satu solusi untuk menjamin keamanan dari suatu informasi. Teknik kriptografi merupakan metode dengan menyandikan isi informasi (*plaintext*) menjadi isi yang sulit atau bahkan tidak dipahami melalui proses enkripsi [1]. Informasi yang asli didapakan dengan melakukan proses dekripsi, yang tentunya dengan menggunakan kunci yang benar. Agar dapat dilakukan dengan baik, dibutuhkan suatu algoritma untuk enkripsi dan dekripsi. salah satu algoritma dari teknik kriptografi yang dapat digunakan dalam mengamankan data adalah algoritma XTEA.

Algoritma *eXtended Tiny Encryption Algorithm* (XTEA) adalah algoritma yang termasuk kedalam kriptografi berbasis blok *cipher*, dan merupakan turunan dari TEA. XTEA memiliki prinsip yang menonjol yaitu *small, secure, simple, dan fast*, dan salah satu alasan yang membuat algoritma ini dianggap aman karena dalam penerapannya tidak menggunakan fungsi *s-boxes* dan permutasian, sehingga terbebas dari analisis frekuensi [2].

Algoritma XTEA merupakan algoritma kunci simetri, di mana untuk proses enkripsi dan dekripsinya menggunakan kunci yang sama. Keamanan dari algoritma simetri adalah tergantung pada kunci yang digunakan, hal ini dikarenakan untuk melakukan proses pendistribusian kunci, pengirim biasanya akan mengirim kunci asli untuk proses dekripsi [3]. Apabila kunci yang digunakan dan dikirim kepada publik dapat diketahui oleh pihak lain, maka hal ini sangat

ISSN 2774-4744 (Media Online)
Vol 1, No 4, Oktober 2021
Hal 144-152
https://hostjournals.com/jimat

berbahaya bagi *ciphertext* yang akan dikirim. Salah satu solusi permasalahan ini adalah melakukan pembangkitan kunci yang lebih acak. Salah satu teknik yang dapat dilakukan untuk membangkitkan kunci acak memanfaatkan pembangkitan bilangan acak berdasarkan metode *Linear Congruential Generator* (LCG).

Metode LCG adalah salah satu jenis pembangkit bilangan acak semu. LCG menggunakan metode *linear* dalam pembangkitan bilangan acak dalam jumlah yang besar dan waktu yang cepat [4]. Berdasarkan penelitian terdahulu, menjelaskan bahwa metode LCG dapat memberikan tingkat keamanan kunci yang cukup baik daripada menggunakan kunci langsung tanpa proses pembangkitan, sehingga dapat meminalisir kebocoran kunci asli[5].

2. METODOLOGI PENELITIAN

2.1 Kriptografi

Kriptografi berasal dari Bahasa Yunani, *crypto* dan *graphia*. *Crypto* berarti *secret* (rahasia) dan *graphia* berarti *writing* (tulisan). Kriptografi adalah sebuah teknik penyandian pesan yang dilakukan agar pesan dapat dikirim dan diterima dengan aman. Kriptografi bertujuan untuk menjaga kerahasiaan data dan informasi agar tidak disalah gunakan oleh pihak yang tidak sah [6]. Agar kriptografi dapat berjalan dengan baik haruslah terdapat empat elemen utama didalamnya, yang paling berkait satu sama lain [7], yaitu :

1. *Plain Text*
   Merupakan sebagai pesan awal atau pesan asli yang dikirim pada proses komunikasi. *Plaintext* inilah yang kemudian dienkripsi dan dideskripsi.
2. *Cipher text*
   Merupakan pesan yang tersembunyi, yaitu pesan asli (*plaintext*) yang telah dienkripsi pada proses kriptografi. *Ciphertext* ini dapat diubah kembali kebentuk aslinya (*plaintext*) memanfaatkan *key* yang telah disediakan.
3. *Cryptography Key*
   Merupakan kunci yang digunakan untuk melakukan enkripsi dan deskripsi pada proses kriptografi. Tanpa adanya kunci (*key*) yang sama maka proses enkripsi dan deskripsi tidaka dapat dilakukan dengan baik. Kunci (*key*) merupakan informasi yang padat menjadi kendali terhadap proses terjadinya kriptografi.
4. *Encryption Decryption Algorithm*
   Komponen terakhir yang juga sama pentingnya dalam proses kriptografi adalah algoritma yang di gunakan untuk enkripsi dan dekripsi.

2.3 Algoritma eXtended Tiny Encryption Algorithm (XTEA)

Algoritma *eXtended Tiny Encryption Algorithm (XTEA*) adalah salah satu algoritma yang dapat digunakan untuk melakukan enkripsi data sehingga data asli hanya dapat dibaca oleh seseorang yang memiliki kunci enkripsi tersebut. Algoritma ini merupakan pengembangan dari *Tiny Encryption Algorithm (TEA)* dan dikembangkan untuk memperbaiki kelemahan dari algoritma tersebut. Perbedaan dengan algoritma sebelumnya adalah penggunakan *key* yang lebih kompleks dan pengaturan urutan dari operasi shift, XOR, dan penambahan [8].

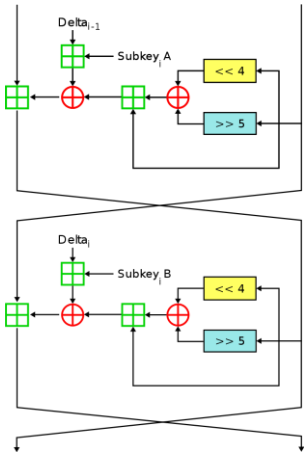
Wheeler dan Needham menciptakan XTEA pada tahun 1997 untuk menutupi kelemahan pada TEA [9]. Sama seperti TEA, XTEA juga beroperasi dalam ukuran blok 64 bit dan panjang kunci 128 bit yang dibagi menjadi 4 blok masik-masing blok 32 bit [9], seperti berikut:

Kunci [0] = Kunci blok 1
Kunci [1] = Kunci blok 2
Kunci [2] = Kunci blok 3
Kunci [3] = Kunci blok 4

Bentuk jaringan *feistel* nya pun masih sama, yang membedakan adalah fungsi *feistel* dan penjadwalan kunci yang digunakan yaitu pada algoritma XTEA, ronde ganjil digunakan K[sum AND 3], sedangkan pada ronde genap digunakan K[sum >> 11 AND 3] [14]. Adapun setiap penjadwalan kunci untuk setiap ronde pada putaran enkripsi tetap menggunakan nilai Delta $9E3779B9_{(16)}$. Berikut rumus dalam menentukan jadwal kunci untuk setiap ronde pada satu putaran enkripsi dan dekripsi XTEA [8]:

Ronde ganjil menggunakan sub kunci dengan rumus :
Kunci [sum + Delta AND 3]
Ronde genap menggunakan sub kunci dengan rumus :
Kunci [sum + Delta >> AND 3]
Keterangan:
Sum = Jumlah putaran enkripsi.
Delta = Nilai konstan dalam algoritma XTEA yaitu $9E3779B9_{(16)}$.
Delta >> = Nilai Delta yang dirubah kedalam biner dan dilakukan pergeseran 11 bit biner delta kekanan.
AND = operator.
Adapun satu putaran algoritma XTEA memiliki 2 ronde enkripsi yang dapat diliat pada gambar 1 di bawah ini :

Journal of Informatics Management and Information Technology
ISSN 2774-4744 (Media Online)
Vol 1, No 4, Oktober 2021
Hal 144-152
https://hostjournals.com/jimat



Gambar 1. Satu Putaran pada Jaringan Fiestel dalam Algoritma XTEA

2.4 Linear Congruential Generator (LCG)

Linear Congruential Generator (LCG) mewakili salah satu algoritma *Pseudo Random Number* yang tertua dan paling populer. Teori dari algoritma ini mudah dipahami dan dapat diimplementasikan secara cepat. Keuntungan dari LCG adalah operasinya yang sangat cepat. LCG dapat diterapkan untuk menghasilkan sekumpulan nilai acak ataupun dapat digunakan untuk mengacak posisi dari sekumpulan nilai [4]. Adapun proses pembangkitan nilai Algoritma *Linear Congruential Generator* (LCG) didefinisikan dalam relasi berulang [4], sebagai berikut:

$X_{n+1} = (aX_n + c) \bmod m$

Keterangan :
Xn = bilangan bulat ke-n
a = bilangan pengali
m = modulus
Jika nilai a = 21, c = 3, dan m = 16, maka perhitungannya adalah:

$$X1 = (21 \times 2 + 3) \bmod 16 = 13$$
$$X2 = (21 \times 13 + 3) \bmod 16 = 4$$
$$X3 = (21 \times 4 + 3) \bmod 16 = 7$$
$$X4 = (21 \times 7 + 3) \bmod 16 = 6$$
$$X5 = (21 \times 6 + 3) \bmod 16 = 1$$

Demikian seterusnya untuk X6, X7, X8, X9,….. Xn.

3. HASIL DAN PEMBAHASAN

Keamanan data file dokumen rahasia sangat rentang terhadap penyerangan untuk membocorkan file dokumen tersebut. Apalagi file dokumen rahasia didistribusikan melalui jaringan *internet*, dimana *internet* merupakan jaringan public yang dapat diakses oleh siapa saja. Berdasarkan tersebut dibutuhkan sebuah teknik pengamanan data file dokumen sebelum dikirimkan kepada penerima, pengamanan data file dokumen rahasia dapat mengandalkan teknik kriptografi.

Berdasarkan rumusan masalah pada bab sebelumnya dan paparan di atas, masalah yang terjadi adalah bagaimana sebuah file dokumen rahasia yang belum disandikan dapat diamankan dengan teknik kriptografi sebelum didistribusikan kepada penerima yang berhak. Teknik kriptografi akan mengacak data file dokumen rahasia menjadi data yang tidak dapat dipahami ketika di baca. Metode yang digunakan dalam pembahasan ini adalah sebuah algoritma kriptografi simetri yaitu algoritma XTEA dengan metode pembangkitan kunci acak *Linear Congruential Generator* (LCG). Pengamanan data file dokumen rahasia dilakukan dengan proses enkripsi menggunakan algoritma XTEA dengan mendapatkan kunci melalui *generate* kunci menggunakan metode *Linear Congruential Generator* (LCG).

3.1 Penerapan Algoritma XTEA

Kasus proses hitungan manual menggunakan algoritma XTEA dengan pengoptimalkan kunci menggunakan LCG. Hal pertama adalah menentukan file dokumen yang akan di enkripsi. File dokumen yang akan dienkripsi secara menual adalah:

Gambar 2. File dokumen Sampel

Berdasarkan pada gambar di atas, file dokumen dengan ekstensi .docx dengan nama biodata.docx. Adapun file dokumen tersebut jika dibuka terlihat seperti pada gambar di bawah:

ISSN 2774-4744 (Media Online)
Vol 1, No 4, Oktober 2021
Hal 144-152
https://hostjournals.com/jimat

Gambar 3. Isi Sampel File Dokumen

Berdasarkan pada gambar 3 di atas, terdapat isi file dokumen yang akan dijadikan sampel dalam perhitungan manual yang terdiri dari beberapa karakter dan gambar. Demi mempermudah proses hitungan manual, maka pada tahap ini diambil beberapa karakter saja yang digunakan untuk keperluan hitungan. Karakter yang digunakan pada isi file dokumen tersebut adalah karakter "BERLIANA". Selanjutnya adalah melakukan tahapan pembangkitan kunci menggunakan metode *Linier Congruential Generator* (LCG), tahapan enkripsi XTEA dan tahapan dekripsi XTEA.

Pembangkitan kunci menggunakan LCG ini bertujuan untuk mendapatkan nilai bilangan acak yang akan menjadi *key* enkripsi menggunakan algoritma XTEA. XTEA bekerja pada kunci 128 bit atau 16 karakter, sehingga proses LCG hanya akan dilakukan sebanyak 16 purulangan untuk menghasilkan 16 deret bilangan acak (128 bit) yang dijadikan sebagai kunci. Berikut adalah tahapan-tahapan pada proses perhitungan manual.

1. Pembangkitan Kunci LCG

Adapun rumus proses pembangkitan kunci menggunakan metode *Linier Congruential Generator* (LCG) adalah sebagai berikut :

$X_{n+1} = (aX_n + c) \bmod m$

Keterangan :
Xn = bilangan bulat ke-n
a = bilangan pengali
c = nilai penambah
m = modulus

Sebelum melakukan proses pembangkitan kunci terlebih dahulu menentukan inputan nilai dari a, c, m dan X_0. Adapun proses membangkitkan bilangan acak sebanyak 16 kali sesuai dengan panjang kunci XTEA dengan ketentuan:

a = 21, c = 3, dan m = 16, $X_0 = 2$.

Nilai a,c,m dan X_0 inilah yang akan diberikan kepada penerima untuk sebagai acuan dalam pembangkitan kunci dekripsi. Adapun proses pembangkitan nilai berdasarkan nilai a,c,m dan X_0 sebagai berikut:

$X_1 = (21 \times 2 + 3) \bmod 16 = 13$
$X_2 = (21 \times 13 + 3) \bmod 16 = 4$
$X_3 = (21 \times 4 + 3) \bmod 16 = 7$
$X_4 = (21 \times 7 + 3) \bmod 16 = 6$
$X_5 = (21 \times 6 + 3) \bmod 16 = 1$
$X_6 = (21 \times 1 + 3) \bmod 16 = 8$
$X_7 = (21 \times 8 + 3) \bmod 16 = 11$
$X_8 = (21 \times 11 + 3) \bmod 16 = 10$
$X_9 = (21 \times 10 + 3) \bmod 16 = 5$
$X_{10} = (21 \times 5 + 3) \bmod 16 = 12$
$X_{11} = (21 \times 12 + 3) \bmod 16 = 15$
$X_{12} = (21 \times 15 + 3) \bmod 16 = 14$
$X_{13} = (21 \times 14 + 3) \bmod 16 = 9$
$X_{14} = (21 \times 9 + 3) \bmod 16 = 0$
$X_{15} = (21 \times 0 + 3) \bmod 16 = 3$
$X_{16} = (21 \times 3 + 3) \bmod 16 = 2$

Sehingga didapatkan keseluruhan nilai dari pembangkitan kunci sebanyak 16 nilai yaitu 13,4,7,6,1,8,11,10,5,12,15,14,9,0,3,2

2. Proses Enkripsi Algoritma XTEA

Adapun proses enkripsi algoritma XTEA terdiri dari 16 *cycle* dan 1 *cycle* terdiri dari 2 ronde dengan ketentuan penjadwalan kunci. Pada algoritma XTEA, ronde ganjil digunakan K[sum AND 3], sedangkan pada ronde genap

digunakan K[sum >> 11 AND 3]. Nilai sum adalah nilai putara ke [i], missalkan berada pada putaran pertama maka nilai sum adalah 0 atau indeks ke [0], jika berada pada putaran kedua maka nilai suma adalah 1 begitu seterusnya. Sedangkan nilai konstan delta sama dengan algoritma TEA yaitu dalam bentuk hexadesimal = 9E3779B9. Tahapan enkripsi algoritma XTEA dengan membentuk setiap *plain* menjadi 64 bit atau 8 karakter dan kunci yang panjangnya 128 bit atau 16 angka. Adapun file dokumen sampel yang menjadi *plain* sudah ditetapkan pada pembahsan sebelumnya dan pembangkitan kunci LCG sudah didapatkkan, kemudian dilanjutkan dengan proses enkripsi dengan ketentuan sebagai berikut:

Sampel isi *plaindokumen* = BERLIANA
Kunci hasil pembangkitan = 13,4,7,6,1,8,11,10,5,12,15,14,9,0,3,2

Plaindokumen dibagi menjadi 2 bagian yaitu block A dan block B :
Block A = **BERL**
Block B = **IANA**

Kemudian kunci = 128 bit dibagi menjadi 4 blok masing-masing 32 bit
Kunci [0] = 13,4,7,6
Kunci [1] = 1,8,11,10
Kunci [2] = 5,12,15,14
Kunci [3] = 9,0,3,2

Kemudian karakter *plaindokumen* "BERLIANA" dirubah kedalam bentuk biner dalam kode ASCII sehingga menjadi :
B = 01000010
E = 01000101
R = 01010010
L = 01001100
I = 01001001
A = 01000001
N = 01001110
A = 01000001

Kemudian dirubah nilai karakter kunci "13,4,7,6,1,8,11,10,5,12,15,14,9,0,3,2" ke dalam bentuk desimal dan biner dengan kode ASCII sehingga menjadi :
13 = 00001101
4 = 00000100
7 = 00000111
6 = 00000110
1 = 00000001
8 = 00001000
11 = 00001011
10 = 00001010
5 = 00000101
12 = 00001100
15 = 00001111
14 = 00001110
9 = 00001001
0 = 00000000
3 = 00000011
2 = 00000010

Biner *plaindokumen* digabungkan menjadi seperti berikut :
AR_0 = 01000010010001010101001001001100
BL_0 = 01001001010000010100111001000001

Kemudian kunci digabungkan kedalam biner kunci menjadi 32 bit perkelompok sehingga menjadi empat bagian kelompok yaitu :
Kunci [0] = 00001101000001000000011100000110
Kunci [1] = 00000001000010000000101100001010
Kunci [2] = 00000101000011000000111100001110
Kunci [3] = 00001001000000000000001100000010

Adapun rumus 1 putaran yang terdiri dari 2 ronde dari algoritma XTEA adalah sebagai berikut :
BLi = ((ARi<<4 XOR ARi>>5)+ARi) XOR (Delta+(Kunci [sum AND 3])) XOR BLi
ARi = ((BLi<<4 XOR BLi>>5)+BLi) XOR (Delta+(Kunci[sum>>11 AND 3])) XOR ARi

Proses XOR dan pejumlahan (OR) bilangan biner mengacu pada bentuk tabel kebenaran Gerbang logika. Dimana tabel kebenaran gerbang logika XOR adalah

ISSN 2774-4744 (Media Online)
Vol 1, No 4, Oktober 2021
Hal 144-152
https://hostjournals.com/jimat

Tabel 1. Nilai Kebenaran XOR

Input X	Input Y	Output Z
1	1	0
0	1	1
1	0	1
0	0	0

Sedangkan tabel kebenaran logika OR (penjumlahan) adalah :

Tabel 2. Nilai Kebenaran OR

Input X	Input Y	Output Z
1	1	1
0	1	1
1	0	1
0	0	0

a. Proses enkripsi putaran pertama

Proses pertama adalah mencari sub kunci untuk setiap ronde dengan ketentuan ronde ganjil menggunakan sub kunci dengan rumus :
Kunci [sum AND 3]
sedangkan untuk ronde genap menggunakan sub kunci dengan rumus :
Kunci [sum>>11 AND 3]
sum adalah nilai jumlah putaran ditambahkan dengan nilai delta. Sehingga untuk putaran pertama sub kunci untuk ronde ganjil adalah :
Kunci = [0 + 9E3779B9 AND 3]
Kunci = [1]

Sedangkan untuk ronde genap putaran pertama adalah :
Kunci = [0 + 9E3779B9>>11 AND 3]
Kunci = [3]
Ronde 1
Plaindokumen 32 bit AR_0 digeser ke kiri 4 bit, kemudian 5 bit ke kanan
AR_0 = 01000010010001010101001001001100
AR_0(kiri) = 00100100010101010010010011000100
AR_0(kanan) = 00100001001000101010100100100110

AR_0(kiri) XOR dengan AR_0(kanan)
AR_0(kiri) = 00100100010101010010010011000100
AR_0(kanan) = 00100001001000101010100100100110
_____XOR
AR_0(kirikanan) = 00000101011101111000110111100010

Kemudian AR_0(kirikanan) OR dengan AR_0
AR_0(kirikanan) = 00000101011101111000110111100010
AR_0 = 01000010010001010101001001001100
_____OR
AR_0(kirikanan) = 01000111011101111101111111101110

Sub kunci enkripsi pertama ronde ganjil di OR dengan nilai nilai bit delta, dimana nilai delta adalah konstan, dan diubah ke bentuk hexadesimal = 9E3779B9 dan diubah menjadi biner. Adapun sub kunci enkripsi ronde ganjil adalah K [1].

Delta = 10011110001101110111100110111001
Kunci [1] = 00000001000010000000101100001010
_____OR
DeltaKunci[1] = 10011111001111110111101110111011

Kemudian DeltaKunci[1] diXORkan dengan AR_0(kirikanan)
DeltaKunci[1] = 10011111001111110111101110111011
AR_0(kirikanan) = 01000111011101111101111111101110
_____XOR

DeltaKunci[1]AR_0(kirikanan)= 11011000010010001010010001010101
Hasil akhir adalah XORkan DeltaKunci[1]AR_0(kirikanan) dengan nilai BL_0 awal
DeltaKunci[1]AR_0(kirikanan) = 11011000010010001010010001010101
BL_0 awal = 01001001010000010100111001000001
_____XOR
BL_1 = 10010001000010011110101000010100

Ronde 2
Hasil enkripsi ronde 1 BL_1 32 bit digeser ke kiri 4 bit, kemudian 5 bit ke kanan
BL_1 = 10010001000010011110101000010100
BL_1(kiri) = 00010000100111101010000101001001
BL_1(kanan) = 01001000100001001111010100001010

BL_1(kiri) XOR dengan BL_1(kanan)
BL_1(kiri) = 00010000100111101010000101001001
BL_1(kanan) = 01001000100001001111010100001010
_____XOR
BL_1(kirikanan) = 01011000000110100101010001000011

Kemudian BL_1(kirikanan) OR dengan BL_1
BL_1(kirikanan) = 01011000000110100101010001000011
BL_1 = 10010001000010011110101000010100
_____OR
BL_1(kirikanan) = 11011001000110111111111001010111

Sub kunci enkripsi pertama ronde genap di OR dengan nilai bit delta, dimana nilai delta adalah konstan, dan diubah ke bentuk hexadesimal = 9E3779B9 dan diubah menjadi biner. Adapun sub kunci enkripsi ronde genap adalah K [3].
Delta = 10011110001101110111100110111001
Kunci [3] = 00001001000000000000001100000010
_____OR
DeltaKunci[3] = 10011111001101110111101110111011
Kemudian DeltaKunci[3] diXORkan dengan BL_1(kirikanan)
DeltaKunci[3] = 10011111001101110111101110111011
BL_1(kirikanan) = 11011001000110111111111001010111
_____XOR
DeltaKunci[3]BL_1(kirikanan) = 01000110001011001000010111101100
01000110001011001000010111101100

Hasil akhir adalah XORkan DeltaKunci[3]BL_1(kirikanan) dengan nilai AR_0
DeltaKunci[3]BL1(kirikanan) = 01000110001011001000010111101100
AR_0 = 01000010010001010101001001001100
_____XOR
AR_1 = 00000100011010011101011110100000

Sehingga didapatkan nilai AR_1 dan BL_1 untuk putaran pertama sebagai berikut:
AR_1 = 00000100011010011101011110100000
BL_1 = 10010001000010011110101000010100
Proses yang sama dilakukan dari putaran kelima hingga putaran ke enambelas (16 *cycle*) atau sampai 32 *round* (1 *cycle* = 2 *round*). Adapun hasil keseluruhan putaran dapat dilihat pada tabel di bawah ini :

Tabel 3. Hasil Putaran Enkripsi

Putaran	AR	BL
1	00000100011010011101011110100000	10010001000010011110101000010100
2	11010001100000010110111011000100	01001010111011111010000100101 1011
3	10111011010000011101001010000000	00100100001111110100011000001111
4	11001011100010110101011111000100	01000000111101011110101001111011
5	01010111100000011110101110000000	00000000001100010100011000101110
6	00100001010000110100101001001001	11001000110100011101011001010100
7	10101101110110010101101111111000	11110010001100011110011000010000
8	11111100000110011110001000110010	11000010111111111100001001010001

Journal of Informatics Management and Information Technology
ISSN 2774-4744 (Media Online)
Vol 1, No 4, Oktober 2021
Hal 144-152
https://hostjournals.com/jimat

9	100101011111101011100001001110101	10100010010110110100101111011100
10	011000101100000001000010111101011	10101000100100111111001000010100
11	00010101000100001010000000111110	1100100000100110000001010001000100
12	100000001100001000000001111010110	00001000111010011001100000000101
13	0100000000010001010000010010000100	01011011000101011101110001101001
14	01101001000011101001010011001001	10100110000100010010110101010000
15	10001000110110010000111001101111	01010100010010001100000000010000
16	001111000000010001000110110110000	00000010100000101101000001000000

Gabungkan kembali blok AR16 dengan BL16 sehingga menjadi *chiperdokumen* biner:
Chiperdokumen :
001111000000010001000110110110000000000010100000101101000001000000
Hasil *chiperdokumen* biner dirubah kedalam bentuk karakter dengan kode ASCII seperti table di bawah ini :

Tabel 4. *Chiperdokumen*

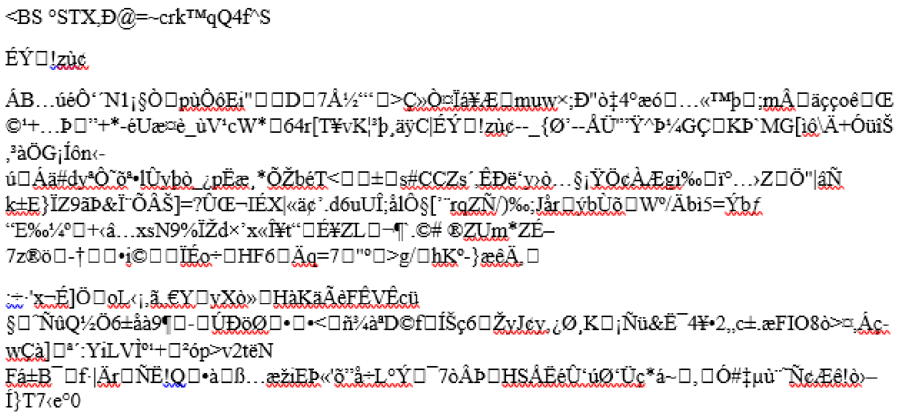
***Chiperdokumen* Biner**	**Karakter *Chiperdokumen***
00111100	<
00001000	BS *(backspace)*
10001101	
10110000	°
00000010	STX
10000010	‚
11010000	Ð
01000000	@

Sehingga didapat karakter *chiperdokumen* adalah "<BS °STX‚Ð@". Berdasarkan hasil enkripsi karakter *plaindokumen* mengalami perubahan menjadi karakter yang tidak dapat dikenali dan dipahami artinya seperti tabel perbandingan di bawah :

Tabel 5. Perbandingan Nilai Hex *Plaindokumen*

Karakter *Plaindokumen*	Karakter *Chiperdokumen*
BERLIANA	<BS °STX‚Ð@

Setelah dilakukanya proses enkripsi, kemudian file dokumen hasil enkripsi disimpan kembali dengan format yang sama, akan tetapi dengan isi file yang sudah tidak dapat dibaca seperti pada contoh di bawah ini:


<BS °STX‚Ð@=~crk™qQ4f^S

ÉÝ□!zùç

ÁB…úéÔ‘´N1¡§Ò□pùÔôEi"□□D□7Å½“‘□>Ç»Ò¤Ïá¥Æ□muw×;Ð"ò‡4°æó□…«™þ□;mÂ□äççoê□Œ
©¹+…Þ□"+*-éUæ¤è_ùV¹cW*□64r[T¥vK¦³þ,äÿC|ÉÝ□!zùç--_{Ø'--ÅÜ"'Ÿ^Þ¼GÇ□KÞ`MG[ìô\Ä+ÓüîŠ
‚³àÖG¡Íôn‹-
ú□Áä#dyªÔ˜õª•lÛybò_¿pËæ,*ÕŽbéT<□□±□s#CCZs´ÊÐë‘y›ò…§¡ŸÖ¢ÀÆgi‰□ï°…›Z□Ö"|âÑ
k±E}ÏZ9ãÞ&Ï¨ÕÂŠ]=?ÛŒ¬IÉX|«ä¢ˆ.d6uUÎ;ålÔ§[ˆ¨rqZÑ/)‰;Jår□ýbÙõ□Wº/Äbi5=Ýbƒ
"E‰¼°□+‹â…xsN9%ÏŽd×ˆx«Î¥t"□É¥ZL□¬¶.©# ®ZUm*ZÉ–
7z®ö□-†□□•¡©□□ÏÉo÷□HF6□Äq=7□"°□>g/□hKº-}æêÄ.□

;÷·'x¬É]Ö□oL‹¡.ã.€Y□yXò»□HàKäÃèFÊVÊcü
§□˜ÑûQ½Ö6±åà9¶□-□ÚÐöØ□•□•<□ñ¾àªD©f□ÍŠç6□ŽyJ¢y,¿Ø,K□¡Ñü&Ë¯4¥•2„c±.æFIO8ò>¤.Åç-
wÇà]□ª´:YìLVÌº¹+□²óp>v2tëN
Fá±B¯□f·|Är□ÑË!Q□•à□ß…æžìEÞ«'õ"å÷L°Ý□¯7òÂÞ□HSÅËéÛ‘úØ‘Üç*á~□,□Ó#‡µù¨˜Ñ¢Æê!ò›–
Í}T7‹e°0

Gambar 4. Sampel File Dokumen Hasil Enkripsi

4. KESIMPULAN

Berdasarkan pengujian yang telah dapat disimpulkan bahwa file dokumen dapat diamankan dengan teknik kriptografi menggunakan algoritma XTEA dan menghasilkan *cipherdokumen* yang tidak dapat dipahami maknanya apabila file dibuka. Hasil modifikasi kunci XTEA dengan teknik LCG berhasil mengamankan file *cipherdokumen* dari kebocoran kunci enkripsi, hal ini dikarenakan kunci yang dikirim kepada penerima adalah bilangan pembangkit a,c,m dan X0 dari LCG.

ISSN 2774-4744 (Media Online)
Vol 1, No 4, Oktober 2021
Hal 144-152
https://hostjournals.com/jimat

REFERENCES

[1] Murdani, "Perancangan Aplikasi Keamanan Data Teks Menggunakan Algoritma Merkle Hellman Knapsack," *Jurnal Pelita Informasi*, vol. 16, no. 3, pp. 302-305, 2017.
[2] Fathima'ruf,"Konsep Penggabungan Alogritma Vigenere Dengan XTEA Blok Chiper Untuk Meningkatkan Keamanan Dokomen Dalam Matakuliah Kriptografi",Jurnal Ilimiah Ilmu Pendidikan, Vol.2, No.1, pp.158-165, 2019
[3] Sistem Pengkodean Data Pada File teks Untuk Keamanan Informasi Dengan Menggunakan Metode Skipjack, Jurnal Computech & Bisnis, Vol.12, No.1, pp.59-72, 2018.
[4] D. Biantara, "Modifikasi Metode Linear Congruential Generator Untuk Optimalisasi Hasil Acak," *Seminar Nasional Informatika*, pp. 182-186, 2015.
[5] R.Aulia, A.Zakir dan M.Zulhafiz, "Penerapan Algoritma One Time Pad& Linear Congruential Generator Untuk Keamanan Pesan Teks", Jurnal Nasional Informatika, Vol.4, No.1, pp.38-41, 2019
[6] E. Setyaningsih, Kriptografi & Implementasi Menggunakan Matlab, Yogyakarta: Andi. 2015
[7] A.A. Ibrahim, "Perancangan Pengamanan Data Menggunakan Algoritma AES (*Advanced Encryption Standard*)", Teknik Informatika STMIK Antar Bangsa, vol. III, pp.53-60, 2017
[9] K.William,"Studi Mengenai *Tiny Encryption Algorithm* (TEA) dan Turunan Turunannya (XTEA dan XXTEA), Institut Teknologi Bandung, 2015.
[9] Wikipedia, (2018,nov.2). XTEA [online]. https://en.wikipedia.org/wiki/XTEA