RESOLUSI : Rekayasa Teknik Informatika dan Informasi
ISSN 2745-7966 (Media Online)
Vol 2, No 3, Januari 2022
Hal 96-105
https://djournals.com/resolusi

Penerapan Algoritma Xtea Dengan Metode Pembangkitan Kunci Linear Congruential Generator Untuk Pengamanan Teks Rahasia

Nusra Anwar, Sinar Sinurat, Imam Saputra

Fakultas Ilmu Komputer dan Teknologi Informasi, Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia
Jalan Sisingamangaraja No. 338, Medan, Sumatera Utara, Indonesia
Email: nusraanwar@gmail.com

Abstrak−Data teks yang bersifat rahasia sangat rentan terhadap penyadapan oleh pihak yang tidak bertanggung jawab. Teks tersebut masih bersifat plaintext sehingga memudahkan pihak attacker untuk melakukan pemanfaatan. Data teks dapat diamankan dengan beberapa teknik. Salah satunya adalah teknik kriptografi. Teknik kriptografi dapat mengamankan data teks rahasia dengan mengacak nilai hexa data tersebut menjadi sebuah data yang tidak dapat dikenali dan dibaca. Salah satu algoritma yang dapat diandalkan adalah algoritma simetri eXtended Tiny Encryption Algorithm (XTEA). Algoritma XTEA adalah algoritma sederhana yang ringan dan cepat dalam proses enkripsi dan dekripsinya. Akan tetapi konsep dari algoritma simetri adalah proses enkripsi dan dekripsi menggunakan kunci yang sama. Keamanan konsep ini sangat rentan apabila kunci bocor dan bersifat publik. Sehingga hal ini dapat dioptimalkan dengan membangkitkan kunci menggunakan metode Linear Congruential Generator (*LCG*). Pembangkitan kunci digunakan dengan bilangan prima. Kunci yang dikirim kepada penerima yang berhak hanya berupa bilangan prima awal. Sehingga penerima harus melakukan pembakitan kunci terlebih dahulu untuk melakukan proses dekripsi. Hal ini dapat mengoptimalkan ciphertext yang dienkripsi menggunakan algoritma XTEA.

Kata Kunci: Kriptografi, Teks, Xtea, LCG

Abstract−Confidential text data is very vulnerable to wiretapping by irresponsible parties. The text is still plaintext, making it easier for the attacker to make use of it. Text data can be secured by several techniques. One of them is cryptographic techniques. Cryptographic techniques can secure secret text data by randomizing the hex value of the data into a data that cannot be recognized and read. One of the reliable algorithms is the eXtended Tiny Encryption Algorithm (XTEA) symmetry algorithm. The XTEA algorithm is a simple algorithm that is lightweight and fast in the encryption and decryption process. However, the concept of the symmetry algorithm is the encryption and decryption process using the same key. The security of this concept is very vulnerable if the key is leaked and is public. So that this can be optimized by generating keys using the Linear Congruential Generator (LCG) method. Key generation is used with prime numbers. The key sent to an eligible recipient is only the initial prime number. So that the recipient has to do key assembly first to carry out the decryption process. This can optimize the encrypted ciphertext using the XTEA algorithm.

Keywords: Cryptography, Text, Xtea, LCG

1. PENDAHULUAN

Saat ini berbagai macam teknik digunakan untuk melindungi informasi yang dirahasiakan dari orang yang tidak berhak terhadap hak akses 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 halnya sebuah data file teks. File teks kerap dijadikan sebuah file yang dikirim untuk mendapatkan informasi atau membagikan informasi kepada orang lain. Semakin rahasia data yang dikirimkan semakin banyak tindak kejahatan atau pencurian yang akan dilakukan oleh pihak attacker.

Berdasarkan pada masalah keamanan data yang dapat merugikan pihak yang memiliki otoritas, solusi yang diberikan adalah dengan memanfaatkan sebuah teknik kriptografi. Teknik kriptografi adalah sebuah teknik yang dapat mengacak atau meyandikan sebuah informasi menjadi informasi yang sulit bahkan tidak dipahami melalui sebuah proses yang di namakan dengan enkripsi [1]. Proses enkripsi pada teknik kriptografi mengandalkan sebuah perhitungan algoritma yang telah ditentukan setiap jenisnya. Salah satunya adalah algoritma eXtended Tiny Encryption Algorithm (XTEA).

Algoritma eXtended Tiny Encryption Algorithm (XTEA) adalah pengembangan dari algoritma TEA. Algoritma TEA adalah salah satu algoritma yang memiliki kesederhanaan dalam implementasinya serta kecepatan proses

enkripsi dan dekripsi yang tinggi [2]. Algoritma TEA beroperasi dalam ukuran blok 64 bit dan panjang kunci 128 bit [2]. TEA berbasiskan jaringan Feistel dan memiliki 32 putaran. Karena memiliki kesederhanaan dalam penjadwalan kuncinya, maka algoritma TEA memiliki kerentanan terhadap equivalent key [3]. Untuk menutupi kelemahan algoritma TEA tersebut maka diciptakan algoritma XTEA dengan penjadwalan kunci dan jaringan feistel yang lebih kompleks dibandingkan dengan TEA. Akan tetapi algoritma XTEA ini dapat diserang jika round-nya dikurangi. XTEA yang paling umum digunakan adalah XTEA 32 ronde. Jadi, hingga saat ini XTEA masih dapat dianggap sebagai algoritma kriptografi yang aman asalkan ronde yang digunakan tidak kurang dari 32 [3].

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 dikarenakn untuk melakukan proses pendistribusian kunci, pengirim biasanya akan mengirim kunci asli untuk proses dekripsi. Apabila kunci yang digunakan dan dikirim bocor kepada publik maka hal ini sangat berbahaya bagi ciphertext yang akan dikirim. Sehingga dalam hal ini dapat diatasi dengan menambahkan proses pembangkitan kunci dengan metode Linear Congruential Generator (LCG). Metode LCG adalah salah satu jenis pembangkit bilangan acak semu. LCG

ISSN 2745-7966 (Media Online)
Vol 2, No 3, Januari 2022
Hal 96-105
https://djournals.com/resolusi

menggunakan metode linier dalam pembangkitan bilangan acak dalam jumlah yang besar dan waktu yang cepat [4], sehingga keamanan chipertext dapat dioptimalkan dengan kunci yang dibangkitkan dengan metode LCG.

Penelitian ini akan menerapkan algoritma XTEA dengan pembangkitan kunci enkripsi menggunakan metode LCG dalam mengenkripsi sebuah file dokumen teks pada aplikasi yang akan dibangun. Hasil yang diharapkan penulis nantinya untuk memaksimalkan peran algoritma XTEA dan metode LCG dalam mengenkripsi file dokumen berupa teks.

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 kerahasian data dan informasi agar tidak disalah gunakan oleh pihak yang tidak sah [5]. Agar kriptografi dapat berjalan dengan baik haruslah terdapat empat elemen utama didalamnya, yang paling berkait satu sama lain [6]

2.2 Algoritma XTEA (eXtended Tiny Encryption Algorithm)

Wheeler dan Needham menciptakan XTEA pada tahun 1997 untuk menutupi kelemahan pada TEA[12]. Sama seperti TEA, XTEA juga beroperasi dalam ukuran blok 64 bit dan panjang kunci 128 bit. Bentuk jaringan *feistel* nya pun masih sama, yang membedakan adalah fungsi *feistel* dan penjadwalan kunci yang digunakan. Pada XTEA, pada ronde ganjil digunakan K[sum & 3], sedangkan pada ronde genap digunakan K[sum>> 11 & 3] [12]. Algoritma XTEA (*eXtended Tiny Encryption Algorithm*) 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 *TEA (Tiny Encryption Algorithm)* 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.

2.3 Metode 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 [14]. Adapun proses pembangkitan nilai Algoritma *Linear Congruential Generator*(LCG) didefinisikan dalam relasi berulang sebagai berikut [16]:

$$Zn = (A .Zn - 1 + C) \bmod M$$

Dimana:
Z_n = bilangan acak ke-n dari deretnya
Z_{n-1} = bilangan acak sebelumnya
A = faktor pengali
C = increment
M=modulus
Z_0 adalah kunci pembangkit atau disebut juga umpan (*seed*).

3. HASIL DAN PEMBAHASAN

3.1 Analisa Masalah

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

Berdasarkan rumusan masalah pada bab sebelumnya dan paparan di atas, masalah yang terjadi adalah bagaimana sebuah teks rahasia yang belum disandikan dapat diamankan dengan teknik kriptografi sebelum didistribusikan kepada penerima yang berhak. Teknik kriptografi akan mengacak data teks 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 teks rahasia dilakukan dengan proses enkripsi menggunakan algoritma XTEA dengan mendapatkan kunci melalui *generate* kunci menggunakan metode *Linear Congruential Generator* (LCG).

Pembangkitan kunci dengan metode *Linear Congruential Generator* (LCG) dilakukan dengan menentukan nilai dari a, c dan m. Nilai a adalah nilai faktor pengali, nilai c adalah nilai *increment* atau nilai penambah dan nilai m adalah nilai akan di modulus. Sedangkan enkripsi menggunakan algoritma XTEA dimulai dengan menentukan *plaintext* sebanyak 64 bit atau 8 karakter, dimana 8 karakter tersebut dibagi menjadi 2 bagian yaitu bagian AR dan bagian BL.

ISSN 2745-7966 (Media Online)
Vol 2, No 3, Januari 2022
Hal 96-105
https://djournals.com/resolusi

Hasil enkripsi algoritma XTEA merupakan *ciphertext* yang akan dikirim kepada penerima. Sedangkan proses dekripsi dilakukan dengan cara yang sama saat proses enkripsi.

3.1.1 Penerapan Pembangkitan Kunci LCG dan Enkripsi Algoritma XTEA

Hal pertama adalah melakukan pembangkitan kunci menggunakan metode *Linear Congruential Generator* (LCG). Pembangkitan kunci menggunakan LCG ini bertujuan untuk mendapatkan nilai bilangan acak yang akan menjadi key enkripsi menggunakan algoritma XTEA

1. Pembangkitan Kunci LCG

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

   $Z_i = (aZ_{i-1} + c) \bmod m$

   Dimana :

   Z_i = bilangan acak ke-i dari deretnya
   Z_{i-1} = bilangan acak sebelumnya
   a = faktor pengali
   c = increment
   m = modulus

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

   a = 4, c = 7, m = 15, dan $Z_0 = 3$

   $Z_1 = (4.3+7) \bmod 15 = 4$
   $Z_2 = (4.4+7) \bmod 15 = 8$
   Z3 = (4.8+7) mod 15 = 5
   $Z_4 = (4.5+7) \bmod 15 = 12$
   $Z_5 = (4.12+7) \bmod 15 = 10$
   $Z_6 = (4.10+7) \bmod 15 = 2$
   $Z_7 = (4.2+7) \bmod 15 = 0$
   $Z_8 = (4.0+7) \bmod 15 = 7$
   $Z_9 = (4.7+7) \bmod 15 = 9$
   $Z_{10} = (4.9+7) \bmod 15 = 13$
   Z11 = (4.13+7) mod 15 = 14
   $Z_{12} = (4.14+7) \bmod 15 = 3$
   $Z_{13} = (4.3+7) \bmod 15 = 4$
   $Z_{14} = (4.4+7) \bmod 15 = 8$
   $Z_{15} = (4.8+7) \bmod 15 = 5$
   $Z_{16} = (4.5+7) \bmod 15 = 12$

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

2. Proses Ekripsi Algoritma XTEA

   Adapun proses enkripsi algoritma XTEA terdiri dari 16 *cycle* dan 1 *cycle* terdiri dari 2 ronde. proses enkripsi algoritma XTEA hampir sama dengan algoritma TEA, perbedaan disni adalah jaringan *fiestel* yang sudah berubah dan kententuan penjadawalan *key*. Pada algoritma XTEA, ronde ganjil digunakan K[sum & 3], sedangkan pada ronde genap digunakan K[sum >> 11 & 3]. Sedangkan nilai konstan delta sama dengan algoritma TEA yaitu dalam bentuk hexadesimal = 9E3779B9. Adapun proses enkripsi ditentukan terlebih dahulu karakter yang akan di enkripsi, pada algoritma XTEA sekali enkripsi membutuhkan 64 bit atau 8 karakter paintext dan 128 bit atau 16 karakter kunci sebagai berikut :

   Plaintext **= NUSRAANW**
   Kunci = 4,8,5,12,10,2,0,7,9,13,14,3,4,8,5,12.
   Plaintext dibagi menjadi 2 bagian yaitu block A dan block B :
   Block A = **NUSR**
   Block B = **AANW**
   Kemudian kunci = 128 bit dibagi menjadi 4 blok masing masing 32 bit
   Kunci [0] = 4,8,5,12
   Kunci [1] = 10,2,0,7
   Kunci [2] = 9,13,14,3
   Kunci [3] = 4,8,5,12
   Kemudian karakter *plaintext* " **NUSRAANW"** dirubah kedalam bentuk biner dalam kode ASCII sehingga menjadi :
   N = 01001110
   U = 01010101
   S = 01010011
   R = 01010010
   A = 01000001

ISSN 2745-7966 (Media Online)
Vol 2, No 3, Januari 2022
Hal 96-105
https://djournals.com/resolusi

A = 01000001
N = 01001110
W = 01010111
Kemudian dirubah nilai karakter kunci "4,8,5,12,10,2,0,7,9,13,14,3,4,8,5,12" kedalam bentuk desimal dan biner dengan kode ASCII sehingga menjadi :
4 = 00000100
8 = 00001000
5 = 00000101
12 = 00001100
10 = 00001010
2 = 00000010
0 = 00000000
7 = 00000111
9 = 00001001
13 = 00001101
14 = 00001110
3 = 00000011
4 = 00000100
8 = 00001000
5 = 00000101
12 = 00001100
Biner *plaintext* digabungkan menjadi seperti berikut :
AR0 = 01001110010101010101001101010010
BL0 = 01000001010000010100111001010111
Kemudian kunci digabungkan kedalam biner kunci:
Kunci [0] = 00000100000010000000010100001100
Kunci [1] = 00001010000000100000000000000111
Kunci [2] = 00001001000011010000111000000011
Kunci [3] = 00000100000010000000010100001100
Adapun rumus 1 putaran yang terdiri dari 2 ronde dari algoritma XTEA adalah sebagai berikut :
Sum = 0 << Posisi putaran XTEA
Delta = 9E3779B9
Sum = sum + delta
Sum = 0 + 9E3779B9 = 9E3779B9 << Putaran ke 0 dan seterusnya
BLi = BLi XOR((ARi<<4 XOR ARi>>5)+ARi) XOR(sum+(Kunci [sum AND 3]))
ARi= ARiXOR((BLi<<4 XOR BLi>>5)+BLi) XOR(sum+(Kunci[sum>>11 AND 3]))
Proses XOR dan pejumlahan (OR) bilangan biner mengacu pada bentuk tabel kebenaran Gerbang logika. Dimana tabel kebenaran gerbang logika XOR adalah

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

1. Proses enkripsi putaran pertama (ke 0)
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]

ISSN 2745-7966 (Media Online)
Vol 2, No 3, Januari 2022
Hal 96-105
https://djournals.com/resolusi

sum adalah nilai jumlah putaran ditambahkan dengan nilai delta. Sehingga untuk putaran pertama (ke 0) 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

Plaintext 32 bit AR0 digeser ke kiri 4 bit, kemudian 5 bit ke kanan
AR0 = 01001110010101010101001101010010
AR0(kiri) = 11100101010101010011010100100100
AR0(kanan) = 00100111001010101010100110101001
AR0(kiri) XOR dengan AR0(kanan)
AR0(kiri) = 11100101010101010011010100100100
AR0(kanan) = 00100111001010101010100110101001
_____XOR
AR0(kirikanan) = 11000010011111111100111001000110 1

Kemudian AR0(kirikanan) ditambahkan dengan AR0
AR0(kirikanan) = 11000010011111111100111001000110 1
AR0 = 01001110010101010101001101010010
_____+
AR0(kirikanan) =11001110011111111101111111011111

Sub kunci enkripsi pertama (ke 0) ronde ganjil di tambahkan dengan nilai nilai bit delta, dimana nilai delta adalah konstan, dan diubah ke bentuk hexadesimal = 9E3779B9 dan diubah menjadi biner. Adapun sub kunci enkripsi ke 0 ronde ganjil adalah K [1].
Delta = 10011110011101110111100110111001
Kunci [1] = 00001010000000100000000000000111
_____+
DeltaKunci[1] = 10011110011101110111100110111111

Kemudian DeltaKunci[1] diXORkan dengan AR0(kirikanan)
DeltaKunci[1] = 10011110011101110111100110111111
AR0(kirikanan) = 11001110011111111101111111011111
_____XOR
DeltaKunci[1]AR0(kirikanan)= 01010000000010001010011001100000

Hasil akhir adalah XORkan DeltaKunci[1]AR0(kirikanan) dengan nilai BL0 awal
DeltaKunci[1]AR0(kirikanan) = 01010000000010001010011001100000
BL0 awal = 01000001010000010100111001010111
_____XOR
BL1 = 00010001010010011110100000110111

Ronde 2

Hasil enkripsi ronde 1 BL1 32 bit digeser ke kiri 4 bit, kemudian 5 bit ke kanan
BL1 = 00010001010010011110100000110111
BL1(kiri) = 00010100100111101000001101110001
BL1(kanan) = 10001000101001001111010000011011

BL1(kiri) XOR dengan BL1(kanan)
BL1(kiri) = 00010100100111101000001101110001
BL1(kanan) = 10001000101001001111010000011011
_____XOR
BL1(kirikanan) = 10011100001110100111011101101010

Kemudian BL1(kirikanan) ditambahkan dengan BL1
BL1(kirikanan) = 10011100001110100111011101101010
BL1 = 00010001010010011110100000110111
_____+
BL1(kirikanan) = 10011101011110111111111101111111

ISSN 2745-7966 (Media Online)
Vol 2, No 3, Januari 2022
Hal 96-105
https://djournals.com/resolusi

Sub kunci enkripsi pertama (ke 0) ronde genap di tambahkan dengan nilai bit delta, dimana nilai delta adalah konstan, dan diubah ke bentuk hexadesimal = 9E3779B9 dan diubah menjadi biner. Adapun sub kunci enkripsi ke 0 ronde genap adalah K [3].

Delta = 10011110011101110111100110111001
Kunci [3] = 00000100000010000000010100001100
_____+
DeltaKunci[3] = 10011110011111110111110110111101

Kemudian DeltaKunci[3] diXORkan dengan BL1(kirikanan)

DeltaKunci[3] = 10011110011111110111110110111101
BL1(kirikanan) = 10011101011110111111111101111111
_____XOR
DeltaKunci[3]BL1(kirikanan) = 00000011000001001000001011000010

Hasil akhir adalah XORkan DeltaKunci[3]BL1(kirikanan) dengan nilai AR0 awal

DeltaKunci[3]BL1(kirikanan) = 00000011000001001000001011000010
AR0 awal = 01001110010101010101001101010010
_____XOR
AR1 = 01001101010100011101000110010000

Sehingga didapatkan nilai AR1 dan BL1 untuk putaran pertama sebagai berikut:
AR1 = 01001101010100011101000110010000
BL1 = 00010001010010011110100000110111

2. Proses enkripsi putaran kedua (ke 1)

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 (ke 0) sub kunci untuk ronde ganjil adalah :
Kunci = [1 + 9E3779B9 AND 3]
Kunci = [2]

Sedangkan untuk ronde genap putaran pertama adalah :
Kunci = [1 + 9E3779B9>>11 AND 3]
Kunci = [3]

Ronde 3

AR1 32 bit digeser ke kiri 4 bit, kemudian 5 bit ke kanan
AR1 = 01001101010100011101000110010000
AR1(kiri) = 11010101000111010001100100000100
AR1(kanan) = 00100110101010001110100011001000
AR1(kiri) XOR dengan AR1(kanan)
AR1(kiri) = 11010101000111010001100100000100
AR1(kanan) = 00100110101010001110100011001000
_____XOR
AR1(kirikanan) = 11110011101101011111000111001100

Kemudian AR1(kirikanan) ditambahkan dengan AR1 awal
AR1(kirikanan) = 11110011101101011111000111001100
AR1 = 01001101010100011101000110010000
_____+
AR1(kirikanan) = 11111111111101011111000111011100

Sub kunci enkripsi kedua (ke 1) ronde ganjil di tambahkan dengan nilai nilai bit delta, dimana nilai delta adalah konstan, dan diubah ke bentuk hexadesimal = 9E3779B9 dan diubah menjadi biner. Adapun sub kunci enkripsi ke 1 ronde ganjil adalah K [2].

Delta = 10011110011101110111100110111001
Kunci [2] = 00001001000011010000111000000011
_____+
DeltaKunci[2] = 10011111011111110111111110111011

ISSN 2745-7966 (Media Online)
Vol 2, No 3, Januari 2022
Hal 96-105
https://djournals.com/resolusi

Kemudian DeltaKunci[2] diXORkan dengan AR1(kirikanan)
DeltaKunci[2] = 10011111011111110111111110111011
AR1(kirikanan) = 11111111111010111110000111011100
_____XOR
DeltaKunci[2]AR1(kirikanan) = 01100000100010101000111001100111

Hasil akhir adalah XORkan DeltaKunci[2]AR1(kirikanan) dengan nilai BL1
DeltaKunci[2]AR1(kirikanan) = 01100000100010101000111001100111
BL1 awal = 00010001010010011110100000110111
_____XOR
BL2 = 01110001110000110110011001010000

Ronde 4
Hasil enkripsi ronde 3 BL2 32 bit digeser ke kiri 4 bit, kemudian 5 bit ke kanan
BL2 = 01110001110000110110011001010000
BL2(kiri) = 00011100001101100110010100000111
BL2(kanan) = 00111000111000011011001100101000
BL2(kiri) XOR dengan BL2(kanan)
BL2(kiri) = 00011100001101100110010100000111
BL2(kanan) = 00111000111000011011001100101000
_____XOR
BL2(kirikanan) = 00100100110101111101011000101111

Kemudian BL2(kirikanan) ditambahkan dengan BL2
BL2(kirikanan) = 00100100110101111101011000101111
BL2 = 01110001110000110110011001010000
_____+
BL2(kirikanan) = 01110101110101111111011001111111

Sub kunci enkripsi kedua (ke 1) ronde genap di tambahkan dengan nilai bit delta, dimana nilai delta adalah konstan, dan diubah ke bentuk hexadesimal = 9E3779B9 dan diubah menjadi biner. Adapun sub kunci enkripsi ke 1 ronde genap adalah K [3].
Delta = 10011110011101110111100110111001
Kunci [3] = 00000100000010000000010100001100
_____+
DeltaKunci[3] = 10011110011111110111110110111101

Kemudian DeltaKunci[3] diXORkan dengan BL2(kirikanan)
DeltaKunci[3] = 10111111011111110111110110111111
BL2(kirikanan) = 01110101110101111111011001111111
_____XOR
DeltaKunci[3]BL2(kirikanan) = 11101011101010001000101111000010

Hasil akhir adalah XORkan DeltaKunci[3]BL2(kirikanan) dengan nilai AR1 awal
DeltaKunci[3]BL2(kirikanan) = 11101011101010001000101111000010
AR1 awal = 01001101010100011101000110010000
_____XOR
AR2 = 10100110111110010101101001010010

Sehingga didapatkan nilai AR2 dan BL2 untuk putaran kedua sebagai berikut:
AR2 = 10100110111110010101101001010010
BL2 = 01110001110000110110011001010000

Proses yang sama dilakukan dari putaran ketiga hingga putaran ke enambelas (16 *cycle*) atau sampai 32 round (1 *cycle* = 2 *round*). Adapun hasil keseluruhan putaran didapatkan AR16 dan BL16 sebagai berikut :
AR16 = 01111110 00001101 01101011 10011001
BL16 = 01010001 01110111 00110100 00011100
Gabungkan kembali block AR16 dengan BL16 sehingga menjadi *chipertext* biner:
***chipertext*:**
01111110000011010110101110011001010100010111011100110100000011100
Hasil *chipertext* biner dirubah kedalam bentuk karakter dengan kode ASCII seperti tabel di bawah ini :

ISSN 2745-7966 (Media Online)
Vol 2, No 3, Januari 2022
Hal 96-105
https://djournals.com/resolusi

Tabel 3. *Chipertext*

Chipertext Biner	Karakter Chipertext
01111110	~
00001101	CR
01101011	k
10011001	™
01010001	Q
01110111	W
00110100	4
00011100	FS

Sehingga didapat karakter *chipertext* adalah "**~CRk™QW4FS**". Berdasarkan hasil enkripsi karakter *plaintext* mengalami perubahan menjadi karakter yang tidak dapat dikenali dan dipahami artinya seperti tabel perbandingan di bawah :

Tabel 4. Perbandingan Nilai Hex *Plaintext*

Karakter Plaintext	Karakter Chipertext
NUSRAANW	**~CRk™QW4FS**

3. Dekripsi Algoritma XTEA

Pada dekripsi, proses pembangkitan kunci dengan metode *Linear Congruential Generator* (LCG) dilakukan dengan cara yang sama yaitu penerima harus tau dan menginputkan nilai dari a, c, dan m, sehingga didapatkan kunci awal yaitu "4,8,5,12,10,2,0,7,9,13,14,3,4,8,5,12**".** Sedangkan untuk proses dekripsi menggunakan algoritma XTEA dilakukan dengan cara yang sama saat proses enkripsi, tetapi proses eksekusi pertama dimulai dari putaran terakhir (16) dan proses untuk setiap ronde dimulai dari chipertext BL lalu dilanjutkan dengan AR. Adapun proses dekripsi secara singkat pada skripsi ini diasumsikan sudah berada pada putaran pertama yaitu ronde 2 dan ronde 1 dengan nilai AR1 dan BL1 sebagai berikut :

AR1 = 01001101010100011101000110010000
BL1 = 00010001010010011110100000110111

Untuk mencari nilai AR0 dan BL0 makan dilakukan proses dekripsi dimulai dengan eksekusi BL1.

Ronde 2

Hasil enkripsi ronde 1 BL1 32 bit digeser ke kiri 4 bit, kemudian 5 bit ke kanan

BL1 = 00010001010010011110100000110111
BL1(kiri) = 00010100100111101000001101110001
BL1(kanan) = 10001000101001001111010000011011
BL1(kiri) XOR dengan BL1(kanan)
BL1(kiri) = 00010100100111101000001101110001
BL1(kanan) = 10001000101001001111010000011011
_____XOR
BL1(kirikanan) = 10011100001110100111011101101010
Kemudian BL1(kirikanan) ditambahkan dengan BL1
BL1(kirikanan) = 10011100001110100111011101101010
BL1 = 00010001010010011110100000110111
_____+
BL1(kirikanan) = 10011101011110111111111101111111

Sub kunci enkripsi pertama (ke 0) ronde genap di tambahkan dengan nilai bit delta, dimana nilai delta adalah konstan, dan diubah ke bentuk hexadesimal = 9E3779B9 dan diubah menjadi biner. Adapun sub kunci enkripsi ke 0 ronde genap adalah K [3].

Delta = 10011110011101110111100110111001
Kunci [3] = 00000100000010000000010100001100
_____+
DeltaKunci[3] = 10011110011111110111110110111101

Kemudian DeltaKunci[3] diXORkan dengan BL1(kirikanan)
DeltaKunci[3] = 10011110011111110111110110111101
BL1(kirikanan) = 10011101011110111111111101111111
_____XOR
DeltaKunci[3]BL1(kirikanan) = 00000011000001001000001011000010

Hasil akhir adalah XORkan DeltaKunci[3]BL1(kirikanan) dengan nilai AR1 untuk mendapatkan nilai AR0
DeltaKunci[3]BL1(kirikanan) = 00000011000001001000001011000010

ISSN 2745-7966 (Media Online)
Vol 2, No 3, Januari 2022
Hal 96-105
https://djournals.com/resolusi

AR1 = **01001101010100011101000110010000**
_____XOR
AR0 = 01001110010101010101001101010010

Ronde 1
Plaintext 32 bit AR0 digeser ke kiri 4 bit, kemudian 5 bit ke kanan
AR0 = 01001110010101010101001101010010
AR0(kiri) = 11100101010101010011010100100100
AR0(kanan) = 00100111001010101010100110101001
AR0(kiri) XOR dengan AR0(kanan)
AR0(kiri) = 11100101010101010011010100100100
AR0(kanan) = 00100111001010101010100110101001
_____XOR
AR0(kirikanan) = 11000010011111111001110010001101
Kemudian AR0(kirikanan) ditambahkan dengan AR0
AR0(kirikanan) = 11000010011111111001110010001101
AR0 = 01001110010101010101001101010010
_____+
AR0(kirikanan) =11001110011111111101111111011111

Sub kunci enkripsi pertama (ke 0) ronde ganjil di tambahkan dengan nilai nilai bit delta, dimana nilai delta adalah konstan, dan diubah ke bentuk hexadesimal = 9E3779B9 dan diubah menjadi biner. Adapun sub kunci enkripsi ke 0 ronde ganjil adalah K [1].
Delta = 10011110011101110111100110111001
Kunci [1] = 00001010000000100000000000000111
_____+
DeltaKunci[1] = 10011110011101110111100110111111

Kemudian DeltaKunci[1] diXORkan dengan AR0(kirikanan)
DeltaKunci[1] = 10011110011101110111100110111111
AR0(kirikanan) = 11001110011111111101111111011111
_____XOR
DeltaKunci[1]AR0(kirikanan)= 01010000000010001010011001100000

Hasil akhir adalah XORkan DeltaKunci[1]AR0(kirikanan) dengan nilai BL1 untuk mendapatkan nilai BL0
DeltaKunci[1]AR0(kirikanan) = 01010000000010001010011001100000
BL1 = **00010001010010011110100000110111**
_____XOR
BL0 = 01000001010000010100111001010111

Sehingga didapatkan nilai AR0 dan BL0 untuk putaran pertama dekripsi XTEA sebagai berikut:
AR0 = 01001110 01010101 01010011 01010010
BL0 = 01000001 01000001 01001110 01010111

Gabungkan kembali block AR0 dengan BL0 sehingga menjadi *plaintext* biner:
plaintext:
01001110 01010101 01010011 01010010 01000001 01000001 01001110 01010111
Hasil *plaintext* biner dirubah kedalam bentuk karakter dengan kode ASCII seperti tabel di bawah ini :

Tabel 5. *Plaintext*

Plaintext Biner	Karakter Plaintext
01001110	N
01010101	U
01010011	S
01010010	R
01000001	A
01000001	A
01001110	N
01010111	W

Sehingga didapat karakter *plaintext* awal adalah **"NUSRAANW".**

4. KESIMPULAN

Berdasarkan hasil dari implementasi sistem yang telah dilakukan pada bab sebelumnya, dapat diambil kesimpulan Penerapan pembangkitan kunci LCG dapat mengamankan file *chipertext* dari kebocoran kunci enkripsi, hal ini dikarenakan kunci yang dikirim kepada penerima adalah kunci pembangkitan. Dengan algoritma XTEA file *plaintext* dapat diubah menjadi file *chipertext* yang tidak dapat diketahui maknanya oleh manusia.

REFERENCES

[1] Murdani, "Perancangan Aplikasi Keamanan Data Teks Menggunakan Algoritma *Merkle Hellman Knapsack*", Jurnal Pelita Informatika, vol. 16, pp284-302, 2017
[2] M.K. Astawa, "Desain Algoritma *Block Cipher* Kadek", *Hacking and Digital Forensics Exposed*, pp.29-33, 2018
[3] K.William, "Studi Mengenai *Tiny Ecryption Algorithm* (TEA) dan Turunan-Turunannya (XTEA dan XXTEA)", Makalah Institut Teknologi Bandung, 2009
[4] D.Biantara et al, "Modifikasi *Metode Linear Congruential Generator* Untuk Optimalisasi Hasil Acak", Seminar Nasional Informatika, pp.182-186, 2015
[5] E. Setyaningsih, Kriptografi & Implementasi Menggunakan Matlab, Yogyakarta: Andi. 2015
[6] A.A. Ibrahim, "Perancangan Pengamanan Data Menggunakan Algoritma AES (*Advanced Encryption Standard*)", Teknik Informatika STMIK Antar Bangsa, vol. III, pp.53-60, 2017
[7] E.R. Agustina and A. Kurniati, "Pemanfaatan Kriptografi Dalam Mewujudkan Keamanan Informasi Pada *e-Voting* di Indonesia", Seminar Nasional Informatika, pp.22-28, 2009
[8] S. Prabowo, (2018, Jan,2). Kriptografi-Jenis Jenis Serangan dalam Kriptografi [online]. Available: http://www.sigitprabowo.id/2013/01/kriptografi-jenis-jenis-serangan-dalam.html
[9] N. Aleisa, "*A Comparison of the 3DES and AES Encryption Standards*", *International Jurnal of Security and Its Applications*, vol. 7, pp.241-246, 2015
[10] J. Thakur and N. Khumar, "DES, AES ad Blowfish: *Symmetric Key Cryptography Algorithm Simulation Based Performance Analysis*, vol. 1, pp.6-12, 2011
[11]Mohtashim, (2015,apr.9). *Cryptografy Hash Functions* [online]. https://www.tutorialspoint.com/cryptography/cryptography.htm
[12] K.William,"Studi Mengenai *Tiny Encryption Algorithm* (TEA) dan Turunan-Turunannya (XTEA dan XXTEA), Institut Teknologi Bandung, 2015.
[13]Wikipedia, (2018,nov.2). XTEA [online]. Available: https://en.wikipedia.org/wiki/XTEA
[14] Biantara et al, "Modifikasi Metode *Linear Congruential Generator* Untuk Optimasi Hasil Acak", Seminar Nasional Informatika, pp.182-186, 2015
[15] Kiaja, (2019,mar.14). Struktur dan Contoh Teks [online]. Available: https://kiajar.com/pengertian-teks-editorial/
[16] B.T Nguyen, (2019,mar.13). *ASCII(), CHAR- String Function* [online]. Available: https://sqljunkieshare.com/2012/01/11/ascii-char-string-functions/
[17]Mohtashim, (2014, jun.12). UML Tutorial [online]. Available: https://www.tutorialspoint.com/uml/index.htm
[18] R. A.S and M. Shalahudin, Modul Pembelajaran: Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek), Bandung: Modula. 2011
[19] Rahmat Priyanto, Langsung Bisa *Visual Basic.Net 2008*. 2009.