

PENERAPAN ALGORITMA CAST-128 PADA PROSES ENKRIPSI DAN DEKRIPSI TEKS

Cut Armida Adhana¹, Rafli^{2*}, M.Zikri Febriansyah³, Mhd Yudya Ananda Hasibuan⁴, Mhd Sanjay Sitepu⁵, Hesty Sisy Olivia⁶, Aulia Ramadhan Hasibuan⁷, Zahrul Laina⁸

^{1,2,3,4,5,6,7,8}Universitas Malikussaleh, Indonesia

¹cut.220170207@mhs.unimal.ac.id, ²raffi.220170056@mhs.unimal.ac.id, ³zikri.220170115@mhs.unimal.ac.id,

⁴aulia.220170044@mhs.unimal.ac.id, ⁵hesty.220170096@mhs.unimal.ac.id, ⁶muhammad.220170104@mhs.unimal.ac.id,

⁷sanjay.220170179@mhs.unimal.ac.id

ABSTRACT

Penyandian yang pertama kali dibuat dengan menggunakan algoritma klasik. Algoritma ini menumpukan keamanannya pada kerahasiaan algoritma yang digunakan. Namun algoritma ini tidak efisien saat digunakan untuk berkomunikasi dengan banyak orang karena algoritmanya masih sangat sederhana dan masih sangat mudah untuk dipecahkan, sehingga informasi atau data penting yang ingin dirahasiakan dengan mudah dapat diketahui orang lain atau orang yang tidak bertanggung jawab. CAST-128 adalah sebuah algoritma kriptografi yang dikatakan mirip dengan algoritma DES dimana menggunakan 16 putaran jaringan feistel sebagai salah satu kekuatannya. Dimana pada proses enkripsi dan dekripsi teks CAST-128 menggunakan panjang blok 64 bit dan panjang kunci sampai 128 bit. Sebagai informasi, algoritma CAST-128 ini disebut sebagai salah satu algoritma kriptografi yang kuat terhadap berbagai macam kriptanalisis, termasuk differential dan linear attack. Dengan adanya penerapan algoritma CAST-128 dalam proses enkripsi dan dekripsi teks akan lebih sulit untuk dipecahkan teks yang disandikan oleh orang-orang yang tidak mengetahui kuncinya sehingga dapat menciptakan keamanan yang lebih dari teks yang telah disandikan. Jadi ketika teks tersebut ingin dikirim atau dipindahkan ke flashdisk akan lebih terjaga kerahasiaannya.

Kata Kunci: Penerapan, Enkripsi, Dekripsi, teks dan Algoritma CAST-128.

PENDAHULUAN

Kriptografi adalah suatu ilmu yang mempelajari bagaimana cara menjaga agar data atau pesan tetap aman saat dikirimkan, dari pengirim ke penerima tanpa mengalami gangguan dari pihak ketiga. Menurut Bruce Schneier dalam bukunya "Applied Cryptography", kriptografi adalah ilmu pengetahuan dan seni menjaga *message-message* agar tetap aman (*secure*). Dalam kriptografi terdapat dua konsep utama yakni enkripsi dan dekripsi[1]. Enkripsi adalah proses mengacak data sehingga tidak dapat dibaca oleh pihak lain. Gambaran sederhana tentang enkripsi, misalnya mengganti huruf a dengan n, b dengan m. Pembahasan enkripsi akan terfokus pada enkripsi *password* dan enkripsi komunikasi data. Dekripsi adalah upaya pengolahan data menjadi sesuatu yang dapat diutarakan secara jelas dan tepat dengan tujuan agar dapat dimengerti oleh orang yang tidak langsung mengalaminya sendiri[2].

Penyandian yang pertama kali dibuat dengan menggunakan algoritma klasik. Algoritma ini menumpukan keamanannya pada kerahasiaan algoritma yang digunakan. Namun algoritma ini tidak efisien saat digunakan untuk berkomunikasi dengan banyak orang karena algoritmanya masih sangat sederhana dan masih sangat mudah untuk dipecahkan, sehingga informasi atau data penting yang ingin dirahasiakan dengan mudah dapat diketahui orang lain atau orang yang tidak bertanggung jawab [3].

CAST-128 adalah sebuah algoritma kriptografi yang menggunakan 16 putaran jaringan feistel sebagai salah satu kekuatannya. CAST-128 menggunakan panjang blok 64 bit dan panjang kunci sampai 128 bit. Sebagai informasi, algoritma CAST-128 ini disebut sebagai salah satu algoritma kriptografi yang kuat terhadap berbagai macam kriptanalisis, termasuk *differential* dan *linear attack*.

METODE PENELITIAN

Keamanan

Keamanan adalah keadaan bebas dari bahaya. Istilah ini bisa digunakan dengan hubungan kepada kejahatan, segala bentuk kecelakaan, dan lain-lain. Keamanan merupakan topik yang luas termasuk keamanan nasional terhadap serangan teroris, keamanan komputer terhadap hacker atau *cracker*, keamanan rumah terhadap maling dan penyusup lainnya, keamanan finansial terhadap kehancuran ekonomi dan banyak situasi berhubungan lainnya[4].

Kriptografi

Kriptografi atau kriptologi merupakan keahlian dan ilmu dari cara-cara untuk komunikasi aman pada kehadirannya di pihak ketiga. Secara umum kriptografi ialah mengenai mengkonstruksi dan menganalisis protokol komunikasi yang dapat memblokir lawan, berbagai aspek dalam keamanan informasi seperti data rahasia, integritas data, autentikasi, dan non-repudansi merupakan pusat dari kriptografi modern. Kriptografi modern terjadi karena terdapat titik temu antara disiplin ilmu matematika, ilmu komputer, dan teknik elektro. Aplikasi dari kriptografi termasuk ATM, *password* komputer, dan *E-commerce*. Kriptografi sebelum pada termodernisasi merupakan sinonim dari *enkripsi*, konversi dari kalimat-kalimat yang dapat dibaca menjadi kelihatan tidak masuk akal. Pembuat dari pesan enkripsi membagi teknik pemecahan sandi yang dibutuhkan untuk mengembalikan informasi asli jika hanya dengan penerima yang diinginkan, sehingga dapat mencegah orang yang tidak diinginkan melakukan hal yang sama. Sejak Perang Dunia I dan kedatangan komputer, metode yang digunakan untuk mengelola kriptologi telah meningkat secara kompleks dan pengaplikasiannya telah tersebar luar[5].

Kriptografi modern sangat didasari pada teori matematis dan aplikasi komputer. Algoritma kriptografi didesain pada asumsi ketahanan komputasional, membuat algoritma ini sangat sulit dipecahkan oleh musuh. Secara teoritis, sangat sulit memecahkan sistem kriptografi, namun tidak layak melakukannya dengan cara-cara praktis. Skema ini disebut sangat aman secara komputasional, kemajuan teoritis dapat meningkatkan algoritma faktorisasi integer, dan meningkatkan teknologi komputasi yang membutuhkan solusi ini untuk diadaptasi terus-menerus. Terdapat skema keamanan informasi yang benar-benar tidak dapat ditembus bahkan dengan komputasi yang tak terbatas namun skema ini sangat sulit diimplementasikan.[3]

Algoritma CAST-128

CAST-128 didesain oleh Carlisle Adams dan Stafford Tavers dari Canada. *CAST-128* termasuk kelas algoritma enkripsi yang menggunakan jaringan *fiestel*. Secara umum algoritma ini mirip dengan algoritma *Data Encryption Standard (DES)*[6]. Input dan keluaran dari algoritma *CAST-128* adalah:

1. Input
 - a. Teks-asli $p_{1..p64}$ (blok teks-asli sepanjang 64 *bit*)
 - b. Kunci $K=k_{1..k128}$ (kunci sepanjang 128)
2. output

Teks-kode $c_{1..c64}$ (blok kode sepanjang 64 *bit*). Fungsi Enkripsi Dalam Algoritma *CAST-128* yaitu:

1. Penjadwalan kunci yaitu menentukan 16 putaran *upa-kunci (subkey)* dari masukan pengguna.
2. Bagi blok menjadi dua bagian yaitu 64 *bit* teks-asli dibagi menjadi dua bagian yang sama, yaitu bagian kiri dan bagian kanan dengan panjang 32 *bit*.
3. 16 putaran jaringan *fiestel*.
4. Konkatenasi untuk membuat teks-kode yaitu tukarkan bagian kiri dengan bagian kanan blok diputar terakhir. Setelah itu kedua bagian digabungkan menjadi satu dan menjadi teks-kode.

Langkah-langkah dekripsi identik dengan langkah-langkah algoritma enkripsi, hanya saja urutan jadwal kunci yang digunakan pada ke-16 putaran dibalik. Jadi kunci terakhir akan digunakan pada putaran pertama dan seterusnya sampai kunci pertama digunakan pada putaran terakhir. (Dony Ariyus, hal 227:2008).

Model jaringan *fiestel* dengan menggunakan:

$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$$

Dalam hal ini i adalah bilangan bulat 1..r (jumlah putaran), K_i adalah *upa kunci* pada putaran ke i sedangkan F adalah fungsi transformasi.

Algoritma enkripsi *CAST-128* didesain untuk mampu menerima berbagai macam panjang kunci yang berbeda mulai dari 40 *bit* sampai dengan 128 *bit* [7], dimana perbedaan antara nilai tersebut harus dalam kelipatan delapan. Jadi panjang kunci yang valid adalah: 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, 120, 128 *bit*. (Ade Gunawan, 2006).

HASIL DAN PEMBAHASAN

CAST-128 adalah algoritma kriptografi kunci simetri yang mengenkripsi data per blok dengan panjang 64 *bit*, dan dengan panjang kunci yang bervariasi antara 40 sampai 128 dalam kelipatan delapan. *CAST-128* menggunakan 16 putaran jaringan *fiestel*.

Langkah-langkah untuk mengenkripsi dengan menggunakan algoritma *CAST-128* yang pertama, tentukan plainteks, lalu lakukan penjadwalan kunci yaitu menentukan 16 pasang upa-kunci (*subkey*) dari masukan pengguna, kedua bagi blok menjadi dua bagian yaitu 64 *bit* teks-asli dibagi menjadi dua bagian yang sama, yaitu bagian kiri (L) dan bagian kanan (R) dengan panjang 32 *bit*, lalu lakukan 16 putaran *fiestel* dengan menggunakan:

$$L_i = R_{i-1} \quad R_i = L_{i-1}$$

⊕ $f(R_{i-1}, K_i)$ Keterangan: i : bilangan bulat 1..r (jumlah putaran)
 K_i : upa kunci pada putaran ke i F : fungsi transformasi (kombinasi)

Selanjutnya Konkatensi untuk membuat teks-kode yaitu tukarkan bagian kiri dengan bagian kanan blok diputar terakhir. Setelah itu kedua bagian digabungkan menjadi satu dan menjadi teks-kode.

Sedangkan langkah-langkah proses dekripsi identik dengan proses enkripsi, hanya saja proses penggunaan ke-16 kunci dalam perputarannya dibalik. Jadi kunci terakhir akan digunakan sebagai putaran pertama dan seterusnya sampai putaran pertama digunakan sebagai putaran terakhir.

Contoh proses enkripsi dan dekripsi menggunakan algoritma *CAST-128*:

Tabel 1. Plainteks terdiri dari 64 bit (8 karakter)

No	Plainteks	Desimal	Biner
1.	U	089	01011001
2.	N	069	01000101
3.	I	083	01010011
4.	M	083	01010011
5.	A	073	01001001
6.	L	065	01000001
7.	I	074	01001010
8.	D	065	01000001

Tabel 2. Kunci sebanyak 128 bit (16 karakter)

No	Plainteks	Desimal	Biner
1.	U	075	01001011
2.	N	069	01000101
3.	I	076	01001100
4.	V	085	01010101
5.	E	065	01000001
6.	R	085	01001011
7.	S	071	01000111
8.	I	065	01000001
9.	T	075	01001011
10.	A	085	01010101
11.	S	075	01001011
12.	A	085	01010101
13.	C	065	01000001
14.	E	084	01010100
15.	H	075	01001011
16.	I	085	01010101

Adapun proses enkripsi yang dilakukan untuk menghasilkan cipherteks adalah:

- Menentukan 16 pasang upa-kunci (*subkey*)

K1 : 01001011 01001011
K2 : 01000101 01010101
K3 : 01001100 01001011
K4 : 01010101 01010101
K5 : 01000001 01000001
K6 : 01010010 01000001
K7 : 01000111 01010100
K8 : 01000001 01010101
K9 : 01001011 01001011

K10 :01001011 01001011
K11 : 01001100 01001011
K12 : 01010101 01010101
K13: 01000001 01000001
K14 : 01010010 01000001
K15 : 01000111 01010100
K16 : 01000001 01010101

2. Bagi plainteks menjadi 2 bagian yang masing-masing menjadi 32 bit.

R0 = 01011001 01000101 01010011 01010011
L0 = 01001001 01000001 01001010 01000001

3. Proses putaran dengan jaringan *fiestel*

Putaran pertama:

L1 = 01001000 01000000 01001001 01000000
R1 = 01011000 01000100 01010010 01010010

Putaran ke-2:

L2 = 01000111 00111111 01001000 00111111
R2 = 01001000 01011100 01001110 01001110

Putaran ke-3:

L3 = 01000110 00111110 01000111 00111110
R3 = 01010000 01000100 00010110 00010110

Putaran ke-4:

L4 = 01000101 00111101 01000110 00111101
R4 = 01010110 01000010 01001100 01001100

Putaran ke-5:

L5 = 01000100 00111100 01000101 00111100
R5 = 01010101 01000000 00001010 00001010

Putaran ke-6:

L6 = 01000011 00111011 01000100 00111011
R6 = 01010001 00111001 00001011 00001011

Putaran ke-7:

L7 = 01000010 00111010 01000011 00111010
R7 = 01011011 01100000 00000010 00000010

Putaran ke-8:

L8 = 01000001 00111001 01000010 00111001
R8 = 01000110 00111010 01011101 01011101

Putaran ke-9:

L9 = 01000000 00111000 01000001 00111000
R9 = 01010001 01111101 01001000 01001000

Putaran ke-10:

L10 = 00111111 00110111 01000000 00110111
R10 = 00110000 00101010 00100111 00100111

Putaran ke-11:

L11 = 00111110 00110110 00111111 00110110
R11 = 01001001 01000011 00111000 00111000

Putaran ke-12:

L12 = 00111101 00110101 00111110 00110101
R12 = 0011000 01011100 00111001 00111001

Putaran ke-13:



L13 = 00111100 00110100 00111101 00110100

R13 = 01001101 00000011 01000100 01000100

Putaran ke-14:

L14 = 00111011 00110011 01000010 00110011

R14 = 00110100 00111110 00111000 00111000

Putaran ke-15:

L15 = 00111010 00110010 01001011 00110010

R15 = 00110111 00111101 00111110 00111110

Putaran ke-16:

L16 = 01001001 01100100 011011101 01100100

R16 = 01001001 01101000 01110110 01110110

4. Gabungkan L16 dan R16 dengan posisi R16 disisi kiri dan L16 pada bagian kanan untuk menghasilkan Cipherteks. Cipherteks yang dihasilkan adalah:

C = 01001001 01101000 01110110 01110110 01001001

01100100 01101101 01100100

(IhvvIdmd)

Untuk mengembalikannya ke bentuk semula dilakukan Proses dekripsi, dengan catatan putaran terakhir pada proses enkripsi menjadi putaran pertama pada proses dekripsi. Adapun proses tersebut adalah sebagai berikut:

1. Putaran pertama:

L1 = 01001001 01101000 01110110 01110110

R1 = 01001001 01100100 01101101 01100100

2. Putaran ke-2:

L2 = 00111011 00110011 01000010 00110011

R2 = 00110100 00111110 00111000 00111000

3. Putaran ke-3:

L3 = 00111100 00110100 00111101 00110100

R3 = 01001101 00000011 01000100 01000100

4. Putaran ke-4:

L4 = 00111101 00110101 00111110 00111101

R4 = 0011000 01011100 00111001 00111001

5. Putaran ke-5:

L5 = 00111111 00110110 00111111 00110110

R5 = 01001001 01000011 00111000 00111000

6. Putaran ke-6:

L6 = 00111111 00110111 01000000 00110111

R6 = 01011011 01100000 00000010 00000010

7. Putaran ke-7:

L7 = 01000000 00111000 01000001 00111000

R7 = 01000110 00111010 01011101 01011101

8. Putaran ke-8:

L8 = 01000001 00111001 01000010 00111001

R8 = 01010001 01111101 01001000 01001000

9. Putaran ke-9:

L9 = 01000010 00111010 01000011 00111010

R9 = 01001000 01011100 01001110 01001110

10. Putaran ke-10:

L10 = 01000011 00111011 01000100 00111011

R10 = 01010000 01000100 00010110 00010110

11. Putaran ke-11:

L11 = 01000100 00111100 01000101 00111100

R11 = 01010101 01000000 00001010 00001010



12. Putaran ke-12:

L12 = 01000101 00111101 01000110 00111101

R12 = 01010001 00111001 00001011 00001011

13. Putaran ke-13:

L13 = 01000110 00111110 01000111 00111110

R13 = 01010000 01000100 00010110 00010110

14. Putaran ke-14:

L14 = 01000111 00111111 01001000 00111111

R14 = 01010110 01000010 01001100 01001100

15. Putaran ke-15:

L15 = 01001000 01000000 01001001 01000000

R15 = 01011000 01000100 01010010 01010010

16. Putaran ke-16:

L16 = 01001001 01000001 01001010 01000001

R16 = 01011000 01000100 01010010 01010010

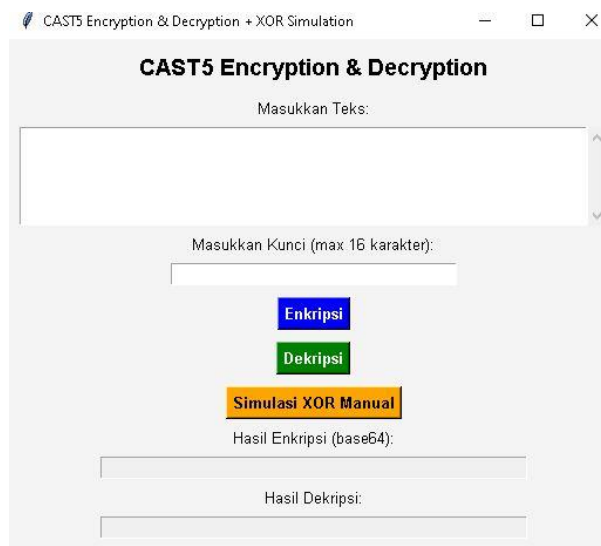
5. Gabungkan L16 dan R16 dengan posisi R16 disisi kiri dan L16 pada bagian kanan untuk menghasilkan Cipherteks. Cipherteks yang dihasilkan adalah:

C = 01011000 01000100 01010010 01010010 01001001 01000001 01001010
01000001 (UNIMALID)

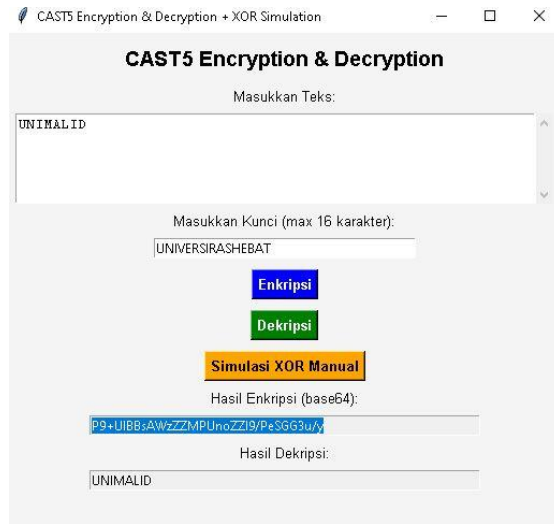
HASIL

Tampilan menu enkripsi dan dekripsi yang paling utama muncul pada saat menjalankan program yaitu seperti gambar di bawah, yang didalamnya terdapat beberapa pilihan yaitu pilih file untuk dienkripsi atau dekripsi.

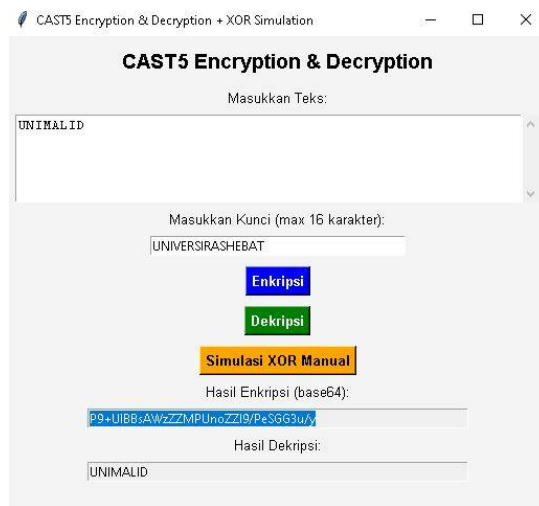
Tampilan Menu Enkripsi dan Dekripsi



Gambar 1. Tampilan Menu Enkripsi dan Dekripsi

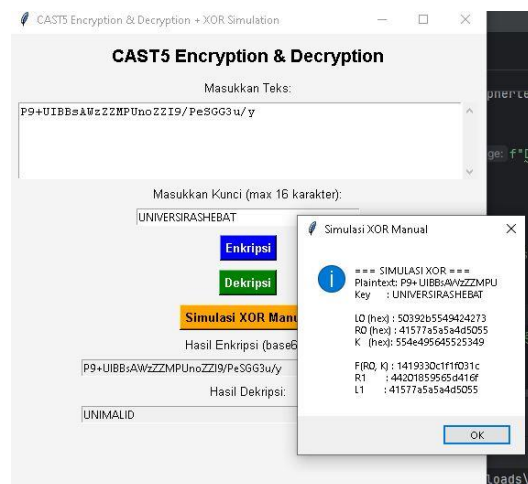


Gambar 2. Tampilan Menu Enkripsi



Gambar 3. Tampilan Menu Dekripsi

Tampilan Hasil Enkripsi dan Dekripsi



Gambar 4. Tampilan Hasil Enkripsi dan Dekripsi

KESIMPULAN

Penerapan algoritma *CAST-128* memiliki 4 tahapan pada perubahan plainteks menjadi cipherteks dan sebaliknya, yaitu penjadwalan kunci atau penentuan 16 putaran upa kunci, penentuan blok, penggunaan 16 putaran jaringan fiestel, dan proses kontaktenasi.

REFERENSI

- [1] D. Ariyus, "Kriptografi keamanan data dan komunikasi," *Yogyakarta Graha Ilmu*, 2006.
- [2] M. Qamal, "Kriptografi File Citra Menggunakan Algoritma Tea (Tiny Encryption Algorithm)," *TECHSI - J. Penelit. Tek. Inform.*, 2014.
- [3] R. Munir, "Kriptografi," *Inform. Bandung*, 2006.
- [4] J. Simarmata *et al.*, "Implementation of AES Algorithm for information security of web-based application," *Int. J. Eng. Technol.*, vol. 7, no. 3.4 Special Issue 4, 2018.
- [5] T. Limbong *et al.*, "The implementation of computer based instruction model on Gost Algorithm Cryptography Learning," in *IOP Conference Series: Materials Science and Engineering*, 2018, vol. 420, no. 1, p. 12094.
- [6] Z. Hercigonja, "Comparative Analysis of Cryptographic Algorithms," *Int. J. Digit. Technol. Econ.*, vol. 1, no. 2, pp. 127–134, 2016.
- [7] G. N. Krishnamurthy, "Encryption Quality Analysis and Security Evaluation of CAST-128 Algorithm and its Modified Version using Digital Images," *Netw. Secur.*, vol. 1, no. 1, pp. 28–33, 2009.