

# Implementasi Algoritma Government Standard (GOST) dalam Pengamanan File Dokumen

Sugiarti<sup>a,1</sup>, Mirnawati<sup>a,2</sup>

<sup>a</sup> Universitas Muslim Indonesia, Jl. Urip Sumoharjo KM.5, Makassar 90231, Indonesia

<sup>1</sup> sugiarti.sugiarti@umi.ac.id; <sup>2</sup>13020110134@umi.ac.id

INFORMASI ARTIKEL	ABSTRAK
Diterima : 10 - 04 - 2020 Direvisi : 24 - 05 - 2020 Diterbitkan : 31 - 07 - 2020	Dokumentasi dalam teknologi saat ini adalah hal yang tak dapat terlepas dari kebutuhan suatu lembaga atau instansi dimana dokumen adalah hal yang paling sering digunakan baik dalam bentuk dokumen manual maupun dalam bentuk file teks dalam komputer. Oleh karena itu, keamanan data atau dokumen rahasia sangat dibutuhkan dalam bisnis maupun pribadi. Tetapi dalam pengiriman atau pengamanan file data yang bersifat rahasia masih kurang dalam sistem keamanan data. Maka dari itu perlu keamanan tambahan untuk proses penyimpanan file baik itu file yang di rasa pribadi maupun data dalam keorganisasian dengan menggunakan proses enkripsi dan deskripsi menggunakan metode Government Standard (GOST). Maka data penting dapat lebih terjaga dan dengan adanya aplikasi ini kita dapat mengenkripsi data untuk menjaga kerahasiaan file yang kita simpan. Aplikasi ini dibangun dengan menggunakan Visual Studio 2010.
<b>Kata Kunci:</b> Keamanan data gost kriptografi	
	

## I. Pendahuluan

Informasi secara umum dapat disampaikan dalam bentuk suara, simbol-simbol teks, tetapi dalam dunia teknologi sekarang ini kita tidak dapat lepas dari sistem komputerisasi yang digunakan sehari-hari untuk menunjang kegiatan manusia baik itu dalam bentuk dokumen maupun secara umum dalam bidang kegiatan komputerisasi sehari-hari di setiap perusahaan maupun organisasi-organisasi yang secara umum sifatnya adalah privasi. Informasi yang ingin disampaikan dapat menjadi suatu hal yang sangat berharga bagi seseorang maupun badan organisasi maka dari itu dibutuhkan adanya keamanan informasi.

Keamanan informasi yang sering digunakan adalah informasi dalam bentuk dokumen dimana dokumen itu sifatnya sangat penting sekarang ini karena banyak orang maupun badan organisasi yang ingin mendapatkan informasi dari orang lain atau badan organisasi lain. Persaingan memaksa orang untuk dapat melakukan sesuatu hal yang dengan cara kejahatan yaitu mengambil dokumen tanpa sepengetahuan pihak organisasi maupun perorangan untuk mendapatkan informasi yang mereka inginkan. Perkembangan teknologi sekarang ini tidak lepas dari internet atau jaringan lokal dimana keamanan data dokumen tidak dapat dijamin keamanannya. Teknologi internet telah menjadi sarana komunikasi dan bertukar informasi bagi masyarakat. Internet merupakan fasilitas umum dimana keamanan informasinya dapat dikatakan tidak aman bagi dokumen rahasia yang ingin di sampaikan ke pihak yang dituju.

Besar kemungkinan dokumen dapat diakses atau dicuri oleh orang yang tidak berkepentingan, maka diperlukan suatu pengamanan informasi dengan menyampaikan pesan dalam bentuk cipher text dimana cipher text ini adalah hasil dari enkripsi sehingga informasi tersebut hanya orang yang berhak saja yang dapat mengetahui plain text atau hasil dari dekripsi dari informasi tersebut. Banyak algoritma yang dapat digunakan untuk melakukan keamanan dokumen salah satunya dengan mengimplementasikan algoritma Government Standard (GOST).

Berdasarkan latar belakang tersebut, maka Penulis menggunakan metode Government Standard (GOST) dikarenakan algoritma tersebut masih kurang digunakan sehingga baik di implementasikan dalam keamanan file dokumen.

## II. Metode

### A. Kriptografi

Menurut Rinaldi Munir (2006:2), kriptografi (Cryptography) berasal dari bahasa Yunani: “cryptos” artinya “secret” (rahasia), sedangkan “graphein” artinya “writing” (tulisan). Jadi, kriptografi berarti “secret writing” atau tulisan rahasia. Beberapa buku telah mendefinisikan kriptografi seperti yang telah didefinisikan oleh buku-buku lama (sebelum tahun 1980-an) menyatakan bahwa kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikan ke dalam bentuk yang tidak dapat lagi dimengerti maknanya. Pada masa lalu kriptografi digunakan untuk keamanan komunikasi penting seperti komunikasi di kalangan militer, diplomat dan mata-mata. Namun saat ini kriptografi digunakan tidak hanya untuk komunikasi penting tetapi digunakan untuk tujuan integrity, authentication dan nonrepudiation.

Sejarah awal kriptografi, setiap orang mempunyai cara sendiri yang unik untuk merahasiakan pesan. Perbedaan cara yang unik dalam merahasiakan pesan sehingga pesan rahasia mempunyai nilai estetika tersendiri. Sejak dahulu hingga sekarang teknik kriptografi terus berkembang. Pada perkembangan kedepannya, kriptografi akan menjadi sebuah disiplin ilmu sendiri karena teknik-teknik kriptografi dapat diformulasikan secara matematik sehingga menjadi sebuah metode yang formal.

### B. Algoritma Government Standard (GOST)

GOST merupakan singkatan dari “Gosudarstvennyi Standard” atau “Government Standard”. Algoritma ini merupakan suatu algoritma block cipher yang dikembangkan oleh seorang berkembangasaan Uni Soviet, dan digunakan untuk menyembunyikan data atau informasi yang bersifat rahasia pada saat berkomunikasi. Algoritma ini merupakan suatu algoritma enkripsi sederhana yang memiliki jumlah proses sebanyak 32 round (putaran) dan menggunakan 64 bit blok cipher dengan 256 bit key. Algoritma GOST juga menggunakan tabel S-box yang berbeda-beda dan operasi XOR serta Left Circular Shift (Muh Manahan PS, Andri H, Jurnal Ilmiah Universitas Kristen Indonesia Teknik Informatika, Vol 1, Nomor 2, Juli 2004).

Kelemahan GOST yang diketahui sampai saat ini adalah karena key sederhana – nya yang sederhana sehingga pada keadaan tertentu menjadi titik lemahnya terhadap metode kriptanalisis seperti Related-key Cryptanalysis. Tetapi hal ini dapat diatasi dengan melewati kunci kepada fungsi hash yang kuat secara kriptografi seperti SHA –1, kemudian menggunakan hasil hash untuk input inisialisasi kunci. Kelebihan dari GOST ini adalah kecepatannya yang cukup baik, walaupun tidak secepat Blowfish tetapi lebih cepat dari IDEA. Komponen dari algoritma GOST antara lain :

1. Key Share Unit (KSU) menyimpan 256 bit string dengan menggunakan 32 bit register (K0, K1, ..., K7).
2. Dua Buah 32 bit register (R1, R2)
3. 32 bit adder modulo 232 (CM1)
4. Bitwise adder XOR (CM2)
5. Substitution block (S) yaitu berupa 8 buah 64 bit S-Box.
6. Rotasi Left Shift register (R) sebanyak 11 bit.

#### Proses Pembentukan Kunci

Proses pembentukan kunci dapat di lihat dalam penjabaran berikut ini :

1. Input key berupa 256 bit key dengan perincian k1, k2, k3, ..., k256.
2. Input key tersebut dikelompokkan dan dimasukkan ke dalam 8 buah KSU dengan aturan seperti berikut,
 
$$K1 = (k32, \dots, k1)$$

$$K2 = (k64, \dots, k33)$$

$$K3 = (k96, \dots, k65)$$

$$K4 = (k128, \dots, k97)$$

$$K5 = (k160, \dots, k129)$$

$$K6 = (k192, \dots, k161)$$

$$K7 = (k224, \dots, k193)$$

$$K8 = (k256, \dots, k225)$$

#### Proses Enkripsi

Proses enkripsi dengan metode GOST untuk satu putaran (iterasi), dapat dilihat pada penjabaran berikut ini,

1. 64 bit plaintext dibagi menjadi 2 buah bagian 32 bit, yaitu  $L_i$  dan  $R_i$ .  
Caranya : Input  $a1(0), \dots, a32(0), \dots, b1(0), \dots(0)$   

$$R0 = a32(0), a31(0), \dots, a1(0)$$

$$L0 = b32(0), b31(0), \dots, b1(0)$$
2.  $(R_i + K_i) \bmod 232$ . Hasil dari penjumlahan modulo 232 berupa 32 bit
3. Hasil dari penjumlahan modulo 232 dibagi menjadi 8 bagian, dimana masing-masing bagian terdiri dari 4 bit. Setiap bagian dimasukkan ke dalam tabel S-Box yang berbeda, 4 bit pertama menjadi input dari S-Box kedua dan seterusnya.

4. Hasil yang didapat dari substitusi ke S-Box dan digabungkan kembali menjadi 32 bit dan kemudian dilakukan rotasi left shift sebanyak 11 bit.
5.  $R_i + I = R_i$  (hasil dari rotate left shift) XOR  $L_i$ .
6.  $L_i + I = R_i$  sebelum dilakukan proses Proses pembentukan kunci modulo, S-BOX, Rotate left Shift dilakukan sebanyak 32 kali (putaran) dengan penggunaan kunci pada masing-masing (putaran) dengan penggunaan kunci pada masing-masing putaran berbeda-beda sesuai dengan putaran berbeda-beda sesuai dengan aturan berikut ini.
 

Putaran 0 - 7	: K0, K1, K2, ..., K7
Putaran 8 - 15	: K0, K1, K2, ..., K7
Putaran 16 - 23	: K0, K1, K2, ..., K7
Putaran 24 - 31	: K7, K6, K5, ..., K0

Untuk putaran ke-31, langkah 5 dan 6 agak sedikit berbeda langkah 5 dan 6 untuk putaran 31 adalah sebagai berikut,

$R_{32} = R_{31}$  sebelum dilakukan proses  
 $L_{32} = L_{31}$  XOR  $R_{31}$  Sehingga chipertext yang dihasilkan adalah  
 $L_{32} : b(32), b(31), \dots, b(1)$   
 $R_{32} : a(32), a(31), \dots, a(1)$   
 $T = a(1), \dots, a(32), b(1), \dots, b(32)$

### Proses Dekripsi

Proses dekripsi merupakan proses kebalikan dari proses enkripsi, penggunaan kunci pada masing-masing putaran pada proses dekripsi adalah sebagai berikut ,

Putaran 0 - 7 : K0, K1, K2, ..., K7  
 Putaran 8 - 15 : K7, K6, K5, ..., K0  
 Putaran 16 - 23 : K7, K6, K5, ..., K0  
 Putaran 24 - 31 : K7, K6, K5, ..., K0

Algoritma yang digunakan untuk proses dekripsi sama dengan proses enkripsi dengan aturan untuk langkah 5 dan 6 pada putaran ke-31 adalah sebagai berikut,

$R_{32} - R_{31}$  sebelum dilakukan proses,  $L_{32} = R_{31}$  XOR  $L_{31}$ .  
 Plaintext yang dihasilkan pada proses dekripsi adalah,  
 $L_{32} - b(32), b(31), \dots, b(1)$   
 $R_{32} - a(32), a(31), \dots, a(1)$   
 $P - a(1), \dots, a(32), b(1), \dots, b(32)$ ,

### Implementasi Hardware dan Software GOST

1. GOST sudah diimplementasikan dalam bentuk perangkat keras.
2. Dalam bentuk perangkat keras, GOST diimplementasikan di dalam chip. Setiap detik chip ini dapat mengenkripsikan 16,8 juta blok (atau 1 gigabit per detik).
3. Implementasi GOST ke dalam perangkat lunak dapat melakukan enkripsi 32.000 blok per-detik (pada komputer mainframe IBM 3090)..

### III. Hasil dan Pembahasan

Analisis algoritma dibutuhkan untuk mengetahui perhitungan secara umum bagaimana perhitungan tersebut menjadi file dokumen seperti contoh berikut:

Diberikan contoh:

- Plaintext(x) = COMPUTER
- Key(k) = 13 34 57 79 9B BC DF F1

#### Langkah Pertama :

Ubahlah plaintext kedalam bentuk biner

C : 01000011  
 O : 01001111  
 M : 01001101  
 P : 01010000  
 U : 01010101  
 T : 01010100  
 E : 01000101  
 R : 01010010

Ubahlah key kedalam bentuk biner

13 : 00010011  
 34 : 00110100  
 57 : 01010111  
 79 : 01111001  
 9B : 10011011  
 BC : 10111100  
 DF : 11011111  
 F1 : 11110001

**Langkah Kedua :**

Lakukan Initial Permutation (IP) pada bit plaintext menggunakan tabel IP berikut:

Tabel 1. Initial Permutation (IP)

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Urutan bit pada plaintext urutan ke 58 ditaruh diposisi 1,  
 Urutan bit pada plaintext urutan ke 50 ditaruh di posisi 2,  
 Urutan bit pada plaintext urutan ke 42 ditaruh di posisi 3, dst  
 Sehingga hasil outputnya adalah

IP(x) : 11111111 10111000 01110110 01010111 00000000 00000000 00000110 10000011

Pecah bit pada IP(x) menjadi 2 bagian yaitu:

L<sub>0</sub> : 11111111 10111000 01110110 01010111 (tabel IP dengan warna Putih) R<sub>0</sub> : 00000000 00000000  
 00000110 10000011 (tabel IP dengan warna Merah)

**Langkah Ketiga :**

proses key yang di gunaka nuntuk mengenkripsi plaintext dengan menggunakan tabel permutasi kompresi 1, pada langkah ini terjadi kompresi dengan membuang 1 bit masing-masing blok kunci dari 64 bit menjadi 56 bit.

Tabel 2. PC-1

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	45	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

Dapat kita lihat pada tabel diatas, tidak terdapat urutan bit 8,16,24,32,40,48,56,64 karena telah dikompres.  
 Berikut hasil outputnya :

CD(k) : 1111000 0110011 0010101 0101111 0101010 1011001 1001111 0001111

Pecah CD(k) menjadi dua bagian kiri dan kanan, sehingga menjadi

$C_0$  : 1111000 0110011 0010101 0101111 (tabel PC-1 warna Putih)

$D_0$  : 0101010 1011001 1001111 0001111 (tabel PC-1 warna Merah)

Sehingga Input:

$R_{16}L_{16} =$  00011111 10010111 10100101 11100110 01101110 10100010 10101000 10110001

Menghasilkan Output:

Cipher(dalam biner) = **01010110 11110001 11010101 11001000 01010010 10101111 10000001 00111111**

Setelah didapatkan nilai biner maka bit yang telah dipecah tadi diubah mejadi file dokumen sesuai format file yang diambil.

#### IV. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan, serta pengujian perangkat lunak yang telah diuraikan pada bab-bab sebelumnya, maka dapat ditarik kesimpulan sebagai berikut:

1. Berdasarkan hasil pengujian perangkat lunak Aplikasi keamanan file menggunakan metode *government Standard* Berbasis Visual Studio.Net, telah bebas dari kesalahan program dan logika sehingga dapat mengatasi permasalahan yang ada pada sistem yang berjalan.
2. Dengan adanya aplikasi keamanan *file* menggunakan metode *government Standard* ini maka diharapkan *file* dokumen yang telah dienkripsi dapat terjaga keamanannya.

#### Daftar Pustaka

- [1] H. Azis, "Network steganography system using covert channel for LSBS stego data on VOIP communication," *Int. J. Eng. Adv. Technol.*, vol. 8, no. 5, pp. 1448–1449, 2019.
- [2] H. Azis and F. Fattah, "Analisis Layanan Keamanan Sistem Kartu Transaksi Elektronik Menggunakan Metode Penetration Testing," *Ilk. J. Ilm.*, vol. 11, no. 2, p. 167, 2019.
- [3] A. Djamililleil, M. Muslim, Y. Salim, E. I. Alwi, H. Azis, and Herman, "Modified Transposition Cipher Algorithm for Images Encryption," *Proc. - 2nd East Indones. Conf. Comput. Inf. Technol. Internet Things Ind. EIconCIT 2018*, pp. 1–4, 2018.
- [4] M. Nazeri, A. Rezai, and H. Azis, "An Efficient Architecture for Golay Code Encoder," *Proc. - 2nd East Indones. Conf. Comput. Inf. Technol. Internet Things Ind. EIconCIT 2018*, pp. 114–117, 2018.
- [5] F. Muharram, H. Azis, and A. R. Manga, "Analisis Algoritma pada Proses Enkripsi dan Dekripsi File Menggunakan Advanced Encryption Standard (AES)," *Pros. Semin. Nas. Ilmu Komput. dan Teknol. Inf.*, vol. 3, no. 2, pp. 112–115, 2018.
- [6] Y. Salim and H. Azis, "Metode Digital Watermark Pada File Penelitian Dosen," *Ilk. J. Ilm.*, vol. 9, no. 2, pp. 161–166, 2017.
- [7] H. Azis and R. Wardoyo, "Penerapan Network Steganography Menggunakan Metode Modifikasi LACK Dan Layanan Message Authentication Code Pada Voip Network Steganography System with modification of LACK and Message Authentication Code on VoIP," *Semin. Nas. Komun. dan Inform.*, pp. 13–19, 2015.