SinkrOn
JURNAL & PENELITIAN TEKNIK INFORMATIKA
Publikasi Jurnal & Penelitian Teknik Informatika
Volume 2 Nomor 2, April 2018
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Implementasi Algoritma *Vigeneere Cipher* dan *GOST* dalam Keamanan Data

Irfan Anas
STMIK Budi Darma Medan
Jl. SM. Raja No. 338 Sp. Limun Medan
uzumakirfan9@gmail.com

Putra Arya Nanda
STMIK Budi Darma Medan
Jl. SM. Raja No. 338 Sp. Limun Medan
putraaryananda4@gmail.com

Abdul Hidayat
STMIK Budi Darma Medan
Jl. SM. Raja No. 338 Sp. Limun Medan
abdulsungkar0@gmail.com

Abstract **—** Pertukaran data informasi di dunia maya sangat banyak dilakukan sekarang ini dan sangat rentan terhadap bahaya akan penyalahgunaan serta penyadapan oleh pihak-pihak yang tidak bertanggung jawab dikarenakan lemahnya proses pengamanan. Tindakan penyalahgunaan pada saat bertukar informasi dapat teratasi dengan menggunakan teknik kriptografi. Kriptografi sendiri memiliki berbagai macam algoritma, misalnya *vigeneee cipher, hill cipher*, *GOST, AES* dan lainnya. Teknik Kriptografi dapat mengamankan data berupa teks dengan proses enkripsi dan deskrpsi sehingga data akan aman. Algoritma *GOST* juga yang penggunaanya sama dengan algoritma *vigenere cipher*. Penelitian ini menguraikan pengkombinasian dua algoritma ini dalam mengamankan data berjenis teks yang sifatnya rahasia atau penting. Pengamanan data berjenis teks berdasarkan kombinasi dua metode ini dilakukan dengan mengenkripsi data terlebih dahulu berdasarkan algoritma *vigeneere cipher*, kemudian *cipher vigeneere* dienkripsi kembali berdasarkan algoritma *GOST*, sehingga dihasilkan *cipher* akhir yang benar-benar acak. Kombinasi dua metode ini diharapkan dapat dijadikan sebagai salah satu solusi dalam pengoptimalan pengaman data khususnya data yang sifatnya rahasia.

Keywords— *kriptografi, algoritma, vigeneere, GOST, data.*

I. PENDAHULUAN

Sekarang ini pertukaran informasi sudah semakin berkembang pesat, teknologi informasi juga telah terbukti mendorong kinerja pada berbagai bidang salah satunya di bidang keamanan, Bidang keamanan sendiri digunakan untuk mengamankan suatu data. Data merupakan bentuk jamak dari *datum* yang berarti fakta atau bagian dari peristiwa yang memiliki arti yang dihubungkan dengan, simbol-simbol, gambar-gambar, angka, huruf, maupun simbol-simbol yang menunjukkan berbagai ide, objek dan lain-lain [1]. Data memiliki berbagai kategori, ada yang sifatnya rahasia maupun tidak rahasia, data yang bersifat rahasia memiliki informasi yang didalamnya sangat dibutuhkan oleh pemilik, sehingga data tersebut perlu diamankan agar tidak disalahgunakan oleh orang yang tidak bertanggung jawab[2]. Penyadapan data yang bermuara pada penyalahgunaan sering terjadi terhadap data yang sering didistribusikan melalui media *online*, karena jalur ini dapat memberikan keleluasaan bagi para penyerang untuk mendapatkan dan mempelajari data yang berhasil didapatkan.

Salah satu solusi yang dapat diimplementasikan untuk megatasi permasalah keamanan data adalah teknik kriptografi. Kriptografi digunakan untuk mengamankan data yang penting, seperti data teks. Data tersebut disandikan berdasarkan algoritma teknik kriptografi untuk merubah data tersebut menjadi data yang tidak dapat lagi dipahami oleh orang lain[2][3].

Vigenere cipher sangat dikenal karena mudah dipahami dan di implementasikan. *Cipher* menggunakan bujursangkar *vigenere* untuk melakukan enkripsi[4]. Namun algoritma ini memiliki kelemahan yaitu kuncinya yang pendek dan penggunaanya yang berulang-ulang, kunci yang berulang ini menimbulkan berbagai celah berupa penggeseran yang sama untuk setiap plainteks yang disubtitusikan oleh teks pada kunci yang sama[5][6].

GOST sebagai salah satu algoritma yang dikembangkan pada era modern, dapat beroperasi pada ukuran blok pesan yang panjangnya 64 bit, sedangkan panjang kuncinya 256 bit. Jumlah putaran di dalam GOST adalah 32 putaran, Kunci *Gost* menggunakan 8 buah S-Box yang berbeda-beda, Gost juga menggunakan operasi XOR dan left *circular*

shift 11 bit atau *rotate left shift* dan menggunakan modulo 2^{32}[1].

Penelitian ini menguraikan proses pengamanan data dalam berjenis teks berdasarkan kombinasi antara algoritma *vigeneere cipher* dengan GOST. Kombinasi yang dilakukan adalah menyandikan data teks secara berlapis (ganda) yang dimulai dari penyandian berdasarkan *vigeneere cipher*, kemudian hasil *vigeneere cipher* dienkripsi kembali berdasarkan GOST, sehingga dihasilkan *cipher* yang benar-benar acak dan sulit untuk dipecahkan oleh penyerang.

II. LANDASAN TEORI

A. *Kriptografi*

Secara umum kriptografi merupakan salah satu teknik pengamanan suatu data atau informasi yang rahasia atau pribadi yang dilakukan dengan cara mengolah *plaintext* dengan suatu kunci tertentu menggunakan suatu proses enkripsi, sehingga menghasilkan *chipertext* yang tidak dapat dipahami maknanya oleh orang lain. *Ciphertext* tersebut dikembalikan menjadi plainteks setelah melalui proses deskripsi pada algoritma[7].

Hal yang harus tercapai dalam menggunakan kriptografi untuk mengamankan data [8][9] adalah :

1. Kerahasiaan, adalah layanan digunakan untuk menjaga informasi dari setiap pihak yang tidak berwenang untuk mengaksesnya.
2. Integritas data, adalah layanan yang bertujuan untuk mencegah terjadinya pengubahan informasi oleh pihak-pihak yang tidak berwenang.
3. Autentikasi, adalah layanan yang terkait dengan identifikasi terhadap pihak-pihak yang ingin mengakses system informasi ataupun keaslian data dari sistem itu sendiri.
4. Ketiadaan penyangkalan, adalah layanan yang berfungsi untuk mencegah terjadinya penyangkalan terhadap suatu aksi yang dilakukan oleh pelaku system informasi

B. *Algoritma Vigeneere Cipher*

Algoritma *vigenere cipher* ini dipublikasikan oleh diplomat Prancis, *blaine de vigenere* pada abad 16 tahun 1586. *Vigenere cipher* sangat dikenal karena mudah dipahami dan diimplementasikan. *Cipher* menggunakan bujursangkar vigenere untuk melakukan enkripsi, Bujursangkar *vigenere* digunakan untuk memperoleh *ciphertext* dengan menggunakan kunci yang sudah ditentukan, jika panjang kunci lebih pendek daripada panjang plainteks maka kunci diulang penggunaanya[6].

Adapun kelebihan kode *vigenere cipher* ialah sulitnya melakukan penyerangan dengan menggunakan analisis frekuensi karena dua huruf yang sama dalam teks kode belum bisa di dekripsikan menjadi dua huruf yang sama dalam plainteks asli. Kelemahan dari *vigenere cipher* ialah panjang kuncinya yang pendek dan penggunaanya yang berulang-ulang. Kunci yang berulang ini menimbulkan berbagai celah berupa penggeseran yang sama untuk setiap plainteks yang disubtitusikan oleh teks pada kunci yang sama[5][8].

Berikut ini adalah penggunaan algoritma *vigenere cipher* [10]:

1. Siapkan plainteks dan kunci
2. Konversi plainteks dan kunci ke decimal
3. Bila jumlah karakter kunci kurang dari jumlah karakter plain, maka tambahkan karakter kunci dengan mengulang dengan karakter kunci itu sendiri
4. Gunakan formula enkripsi $Ci = (Pi + Ki) \bmod 256$
5. Hasil enkripsi konversi menjadi karakter *ciphertext*.
6. Formula untuk dekripsi $Pi = (Ci-Ki) \bmod 256$

Keterangan :
Ci : nilai desimal karakter *ciphertext* ke-i
Pi : nilai desimal karakter *plaintext* ke-i
Ki : nilai desimal karakter kunci ke-i

C. *Algoritma GOST*

Gosudarstvennyi Standard (GOST) atau standard pemerintah, adalah salah satu algoritma enkripsi dari Negara uni soviet yang dikembangkan pada tahun 1970. *GOST* beroperasi pada ukuran blok pesan yang panjangnya 64 bit, sedangkan panjang kuncinya 256 bit. Jumlah putaran pada GOST adalah 32 putaran, setiap putaran menggunakan kunci internal. Kunci GOST menggunakan 8 buah S-Box yang berbeda-beda, GOST juga menggunakan operasi XOR dan *left circular shift* 11 bit atau *rotate left shift* dan menggunakan modulo 2^{32}[1].

Adapun proroses enkripsi algoritma GOST [1][11], adalah :

a. Konversi plainteks dan kunci ke biner
b. Kelompokkan biner menjadi 8 kelompok dengan jumlah bit kunci setiap kelompok adalah 32 bit :
   K[0] = bit ke 32..1,
   K[1] = bit ke 64..33,
   K[2] = bit ke 96 …65,
   K[3] = bit ke 128 … 97,
   K[4] = bit ke 160 … 129,
   K[5] = bit ke 192 … 161,
   K[6] = bit ke 224 …193
   K[7] = bit ke 256….225
c. Kelompokkan plainteks menjadi 2 bagian yaitu R[0] dan L[0] dengan jumlah tiap kelompok adalah 32bit. 32 bit bagian kiri menjadi R[0] dan

mulai dari bit 33 sampai 64 bit (sebelah kanan) menjadi L[0].
R[0] = bit[32], bit[31]……bit[1]
L[0] = bit[64], bit[63]……bit[33].

d. (Ri + Kj) mod 2^{32} hasil dari penjumlahan modulo 2^{32} adalah 32 bit.
e. Hasil dari penjumlahan mod 2^{32} dibagi menjadi 8 kelompok (4bit perkelompok), kemudian permutasikan dengan SBOX
f. Gabungkan biner-biner hasil permutasi SBox dan lakukan proses *rotate left shift* sebanyak 11 bit
g. Rj+1 = RLS xor Li
h. Lj+1= Ri sebelum dilakukan proses

Proses dekripsi berdasarkan GOST tidak berbeda dengan cara enkripsi, hanya saja susunan kunci yang digunakan adalah :
Putaran ke 0 s/d 7 = K[0],K[1],…K[7]
Putaran ke 8 s/d 15 = K[7],K[6].. K[0]
Putaran ke 16 s/d 23 = K[7],K[6].. K[0]
Putaran ke 24 s/d 31 = K[7],K[6],.. K[0]

Penjadwalan kunci yang sederhana pada algoritma GOST menjadi salah satu utama algoritma ini, sehingga pada keadaan tertentu terdapat titik lemahnya pada metode kriptanalisis sebagai kunci terkait pembacaan sandi.

D. *Data*

Data merupakan bentuk jamak dari *datum* yang berarti fakta atau bagian dari peristiwa yang memiliki arti yang dihubungkan dengan, simbol-simbol, gambar-gambar, angka, huruf yang menunjukkan berbagai ide, objek dan lain-lain[1]. Data dapat berupa *file* dan berbentuk string yang secara umum dibagi menjadi dua, yaitu data yang bersifat rahasia dan tidak rahasia[2].

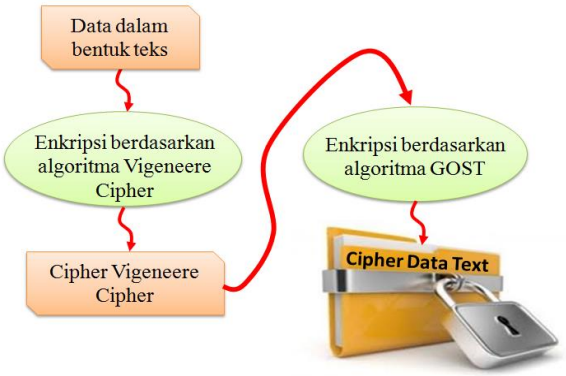
III. PEMBAHASAN

Pemanfaatan algoritma *vigenere cipher* dan *algoritma* GOST untuk penyandian data yang sifatnya rahasia dan atau penting memberi manfaat untuk mengamankan data yang akan didistribusikan kepada si penerima. Pemanfaatan satu algoritma kriptografi dalam proses penyandian data bukan tidak memungkinkan, namun akan lebih optimal *cipher* yang didapatkan bila dikombinasikan dengan algoritma yang lain. Salah satu kelebihan yang didapatkan bila terjadi pengkombinasian algoritma dalam proses penyandian data adalah terjadinya proses enkripi/penyandian data asli secara berlapis (ganda/dua kali), sehingga dihasilkan *cipher* yang benar-benar acak dan tidak lagi memperlihatkan korelasi dengan data asli.

Pengkombinasian dua algoritma ini dalam menyandikan data asli yang berjenis teks, dilakukan dengan menyandikan teks asli terlebih dahulu berdasarkan algoritma *vigeneere cipher*. *Cipher* yang dihasilkan oleh proses enkripsi *vigeneere cipher* dienkripsi kembali berdasarkan algoritma GOST.

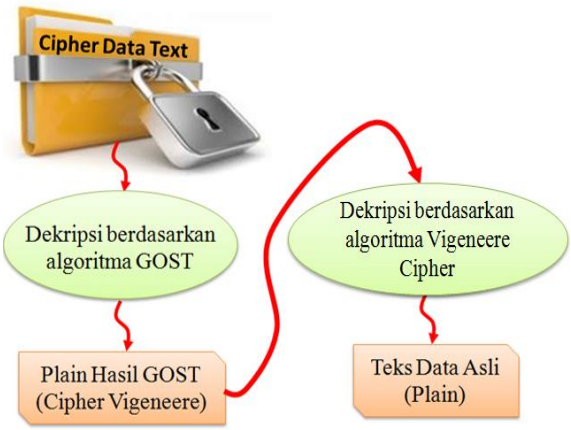
Berdasarkan pengkombinasian dua algoritma ini, maka dibutuhkan dua kunci enkripsi dan dekripsi yaitu kunci yang digunakan pada proses enkripsi dan dekripsi berdasarkan vigeneere *cipher* serta kunci yang digunakan pada proses enkripsi dan dekripsi GOST. Kedua kunci yang digunakan disini bersifat simetris, karena algoritmanya juga bersifat simetris. Artinya kunci yang digunakan pada proses enkripsi dan dekripsi adalah sama.

Adapun skema pengkombinasian dua algoritma ini dalam menyandikan data berjenis teks ditunjukkan pada gambar di bawah ini.



Gambar 1. Skema Proses Enkripsi Data

Berdaasarkan gambar 1 di atas, diketahui bahwa *cipher* data teks yang dihasilkan telah melalui dua kali proses penyandian. Penyandian diawali dengan algoritma *vigeneere cipher*, kemudian dienkripsi kembali berdasarkan algoritma GOST. Berikut ini akan disajikan skema proses dekripsi.



Gambar 2. Skema Proses Dekripsi Data

Gambar 2 menunjukkan bahwa proses dekripsi merupakan kebalikan dari proses enkripsi, yaitu diawali dengan proses dekripsi berdasarkan algoritma GOST, kemudian *plain* hasil dekripsi GOST didekripsi kembali berdasarkan algoritma *vigeneere* hingga didapatkan *plaintext* dari data asli.

Berikut ini akan dilakukan proses penyandian data teks berdasarkan kombinasi dua algoritma ini. Data teks yang dijadikan contoh dalam penelitian ini adalah data teks **VIGENERE** dengan kunci algoritma *vigeneere* adalah **ABDULABD** dan kunci algoritma GOST : **Algoritma_GOST_Irfan_AnasZZ_2017.**

1. Proses Penyandian Data Teks berdasarkan Algoritma *Vigeneere Cipher*

   Plaintext = VIGENERE
   Key Vigeneere = ABDULABD

   Konversi karakter *plaintext* ke nilai desimal ASCII :

   Tabel 1. Nilai ASCII Plain dan Kunci

V	I	G	E	N	E	R	E
86	73	71	69	78	69	82	69
A	B	D	U	L	A	B	D
65	66	68	85	76	65	66	68

   Lakukan proses enkripsi berdasarkan formula enkripsi *vigeneere cipher* :

   Ci=(P1+K) mod 256
   C1=(V+A)mod 256
   C1=86+65 = 151 = —

   C4=(P4+K) mod 256
   C4=(E+U) mod 256
   C4=69+85 = 154 =š

   C2=(P2+K)mod 256
   C2=(I+B) mod 256
   C2=73+66 = 139 = ‹

   C5=(P5+K) mod 256
   C5=(N+L) mod 256
   C5=78+76 = 154 =š

   C3=(P3+K) mod 256
   C3=(G+D) mod 256
   C3=71+68 = 139 =‹

   C6=(P6+K) mod 256
   C6=(E+A) mod 256
   C6=69+65=134 = †

   C7=(P7+K) mod 256
   C7=(R+B) mod 256
   C7=82+66 =148 = “

   C8=(P8+K) mod 256
   C8=(E+D) mod 256
   C8=69+68= 137 =‰

   Hasil dari keseluruhan proses di atas, *plaintext* VIGENERE dengan kunci ABDUL maka menghasilkan *Ciphertext* —‹‹ š š†“‰
   Cipher hasil proses enkripsi *vigeneere cipher* inilah yang menjadi input selanjutnya untuk dienkripsi kembali berdasarkan algoritma GOST.

2. Proses penyandian *Cipher* Hasil *Vigeneere* Berdasarkan Algoritma GOST

   Proses enkripsi GOST diawali dengan membangkitkan sub kunci yang digunakan pada proses enkripsi maupun dekripsi berdasarkan kunci utama yang diinput oleh pengguna.

   Plaintext : —‹‹ š š†“‰
   Kunci : Algoritma_GOST_Irfan_AnasZZ_2017

   Konversi kunci ke nilai desimal ASCII, sehingga dihasilkan :

   Tabel 2. Nilai Biner Kunci GOST

Char	dec	Biner	char	dec	biner
A	65	01000001	a	97	01100001
l	108	01100100	n	110	01101110
g	103	01100111	_	95	01011111
o	111	01101111	A	65	01000001
r	114	01110010	n	110	01101110
i	105	01101001	a	97	01100001
t	116	01110100	s	115	01110011
m	109	01101101	Z	90	01011010
a	97	01100001	Z	90	01011010
_	95	01011111	_	95	01011111
G	71	01000111	2	50	00110000
O	79	01001111	0	48	00110000
S	83	01010011	1	49	00110001
T	84	01010100	7	55	00110111
_	95	01011111			
I	73	01110010			
r	114	01100110			
f	102	01100110			

   Kelompokkan biner-biner kunci, sesuai aturan kuncinya, sehingga dihasilkan :

   Tabel 1. Sub Kunci GOST

Kunci	Posisi bit	Biner yang diambil
K[0]	32..1	11110110111001100011011010000010
K[1]	64..33	10110110001011101001011001001110
K[2]	96..65	11110010111000101111101010000110
K[3]	128..97	10010010111110100010101011001010
K[4]	160..129	01110110100001100110011001001110
K[5]	192..161	10000110011101101000001011111010
K[6]	224..193	11111010010110100101101011001110
K[7]	256..225	11101100100011000000110001001100

   Konversi *plaintext* ke bilangan biner kemudian lakukan proses pengelompokkan menjadi 2 kelompok L[0] dan R[0].
   L[0] = 10010001001010010110000101011001
   R[0] = 01011001110100011101000111101001

   Proses Enkripsi Round-0 :
   $R[0] = (R[0] + K[0]) \text{ Mod } 2^{32}$
   $R[0] = (1506922985 + 4142282370) \text{ Mod } 2^{32}$
   R[0] = 01010000101110000000100001101011

   Lakukan S-Box dan RLS tehadap hasil R[0], sehingga dihasilkan :
   R[0] = 11110011000110101111010001110011

Agar nilai R selanjutnya didapatkan, maka digunakan formula R[i hasil S-BOX dan RLS] XOR L[i]. Sedangkan untuk mencari nilai L selanjutnya diambil dari nilai R[sebelumnya]. Sehingga dihasilkan nilai :

R[1] = 01100010001100111001010100101010
L[1] = 01011001110100011101000111101001

Langkah ini dilakukan hal yang sama sebanyak 32 iterasi, hanya saja kunci yang digunakan pada setiap iterasi disesuaikan dengan aturan.

Iterasi ke-32 memberikan hasil *cipher* sebagai berikut :
L[32] = 11010111101011111101110011000001
R[32] = 01110100001110000101011100110101

Kemudian biner L[32] dan R[32] digabungkan kemudian dikonversi menjadi karakter *cipher*, sehingga dihasilkan *cipher* dari proses enkripsi GOST adalah **t8W5×¯ÜÁ**

3. Proses Dekripsi

Berdasarkan diagaram skema dekripsi (gambar 2), bahwa proses dekripsi dilakukan secara terbalik yaitu dimulai dari proses dekripsi *cipher* pesan berdasarkan algoritma GOST, kemudian *plain* yang didapatkan didekripsi kembali berdasarkan algoritma *vigeneere cipher* untuk menghasilkan *plain* data teks.

Dekripsi berdasarkan algoritma GOST dilakukan dengan cara yang sama seperti proses enkripsi, hanya saja susunan kunci yang digunakan berbeda. Hal yang sama dilakukan juga pada proses dekripsi berdasarkan algoritma *vigeneere cipher* hingga dihasilkan kembali data teks asli seperti semua.

IV. KESIMPULAN

Berdasarkan uraian pembahasan penelitian ini, maka disimpulkan bahwa pemanfaatan kombinasi dua algoritma ini dalam pengamanan data berjenis teks sangat baik karena data asli akan disandikan secara ganda (dua kali) dan akan menghasilkan *cipher* yang cukup acak serta rumit diketahui makna aslinya. Proses enkripsi maupun dekripsi yang dilakukan secara ganda menyebabkan waktu proses enkripsi maupun dekripsi menjadi lama.

REFERENSI

[1] T. Sustabri, *Konsep Sistem Informasi*, Yogyakarta:Andi Offset, 2012.
[2] A.H. Hasugian, "Implementasi Algoritma Hill Cipher Dalam Penyandian Data," Pelita Informatika Budi Darma, vol.4, no.2, Agustus 2013
[3] T. Zebua, "Penerapan Metode LSB-2 untuk Menyembunyi-kan Ciphertext Pada Citra Digital," *Pelita Inform. Budi Darma*, vol. 10, no. 3, pp. 135–140, 2015.
[4] I.P. Herryawan, "Aplikasi keamanan data menggunakan metoda kriptografi GOST", Jurnal TSI, vol 1, no 2, 138-149, Juli 2010.
[5] A. Pardede, M. H.,Manurung, H. and D. Filina, "Algoritma Vigenere Cipher Dan Hill Cipher Dalam Aplikasi Keamanan Data Pada File Dokumen", JTIK, vol 1, no 1. Januari 2017.
[6] R. Sadikin, *Kriptografi Untuk Keamanan Jaringan dan Implementasinya dalam Bahasa Java*, Yogyakarta: Andi, 2012.
[7] T. Zebua and E. Ndruru, "Pengamanan Citra Digital Berdasarkan Modifikasi Algoritma RC4," *J. Teknol. Infomasi dan Ilmu Komput.*, vol. 4, no. 4, pp. 275–282, Desember 2017.
[8] E. Setyaningsih E, Kriptografi & Implementasinya Menggunakan MATLAB, Yogyakarta: Andi Offset, 2015.
[9] T. Zebua, "Analisa dan Implementasi Algoritma Triangle Chain Pada Penyandian Record Database," *Pelita Inform. Budi Darma*, vol. 3, no. 2, pp. 37–49, April 2013.
[10] Priyono, "Penerapan algoritma Caesar Cipher dan Algoritma Vigenere Cipher Dalam Pengamanan Pesan Teks, JURIKOM, vol.3, no.5. Oktober 2016.
[11] M. Iqbal M, Y. Saputra Y and A.P.U. Siahaan,"The understanding of GOST Cryptography Techinque", IJETT, vol.39, no.3, September 2016.