Penggunaan Kriptografi One Time Pad (Algoritma Vernam) dalam Pengamanan Informasi

Nama Penulis : Firman Rickson Saragih[1)]
1) Jurusan Teknik Informatika ITB, NIM 13506096, email: if16096@if.itb.ac.id

Abstract – *Kerahasiaan data sangat diperlukan dalam hal komunikasi data.Untuk menjamin keamanan dan kerahasiaan data tersebut diperlukan teknik tertentu untuk menyandikan data atau informasi yang disebut kriptografi. Ada berbagai jenis algoritma kriptografi seperti DES, RSA dan sebagainya yang berusaha untuk menciptakan suatu algoritma yang benar-benar dapat mengamankan data atau informasi yang ditransmisikan. Untuk itu para kriptografer semakin lama semakn berusaha menciptakan algoritma yang rumit untuk lebih menjamin keamanan informasi yang dienkripsikan. Namun sebenarnya semakin sederhana algoritma pengenkripsian yang dibuat akan semakin baik, karena dengan demikian proses komputasinya akan semakin sedikit sehingga akan memakan waktu lebih sedikit untuk mengeksekusinya.*

Makalah ini akan membahas salah satu bentuk kriptografi yang relatif sederhana namun cukup sulit untuk dipecahkan yaitu kriptografi one-time pad.

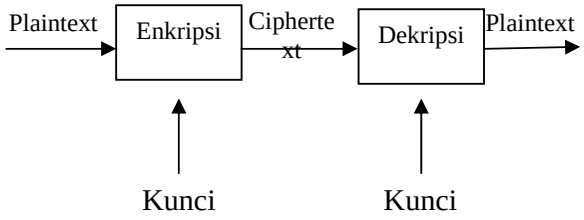
Kata Kunci – *kerahasiaan data, kriptografi, one time pad*

1. PENDAHULUAN

Suatu instansi atau organisasi sangat membutuhkan keamanan infrastruktur teknologi informasi yang baik untuk melindungi aset-asetnya terutama informasi-informasi dan data-data yang penting dan sensitif. Sehingga saat ini dibutuhkan system keamanan yang lebih kompleks dengan kesanggupan untuk mengikuti perkembangan yang ada agar dapat melindungi system dari berbagai ancaman yang mungkin timbul.

Salah satu cara yang digunakan untuk menjamin keamanan dari infrastruktur teknologi adalah dengan menjamin keamanan komunikasi. Komunikasi yang aman dimaksudkan untuk melindungi data ataupun informasi ketika dikirimkan atau ditransmisikan kepada pihak lain, sehingga data atau informasi yang ditransmisikan itu tidak dapat disadap dimanipulasi ataupun dirusak oleh pihak-pihak yang tidak bertanggungjawab. Salah satu cara untuk mengamankan komunikasi adalah dengan menerapkan teknik penyandian/ kriptografi.

Kriptografi adalah ilmu sekaligus seni untuk menjaga kerahasiaan pesan (data atau informasi) dengan cara menyamarkannya menjadi bentuk tersandi yang tidak mempunyai makna. Pesan yang dirahasiakan dalam kriptografi disebut plainteks (plaintext) dan hasil penyamaran disebut chiperteks (chipertext). Proses penyamaran dari palinteks ke chiperteks disebut enkripsi (dari kata encryption) dan proses pembalikan dari chiperteks menjadi plainteks kembali disebut dekripsi (decryption). Baik proses enkripsi maupun proses dekripsi melibatkan satu atau beberapa kunci kriptografi. Dalam suatu system di mana terdapat algoritma kriptografi, ditambah seluruh kemungkinan plaintext, ciphertext dan kunci-kuncinya disebut kriptosistem (cryptosystem atau cryptographic system) Proses tersebut dapat digambarkan secara sederhana sebagai berikut :



Gambar 1.Enkripsi/Dekripsi Sederhana

Ada banyak jenis metode kriptografi yang umum digunakan. Salah satu metode kriptografi yang tertua adalah dengan menggunakan scytale yang digunakan oleh tentara Sparta di Yunani pada permulaan tahun 400 SM [1]. Alat ini merupakan sebuah pita panjang dari daun papyrus yang dililitkan pada sebatang silinder. Pesan yang akan dikirmkan ditulis secara horizontal, kemudian pita dilepas. Untuk dapat membaca pesan itu kembali, orang yang menerima pesan harus melilitkan kembali pita tersebut pada silinder yang diameternya sama dengan silinder yang dipergunakan pengirim pesan tersebut. Teknik kriptografi ini disebut metode transposisi yang merupakan metode enkripsi tertua.

Aritmetika modular dipergunakan secara meluas dalam metode-metode kriptografi. Yang paling banyak dipergunakan yaitu aritmetika modulo 2 yang hanya melibatkan 0 dan 1 saja sehingga mirip dengan bit pada computer. Selain aritmetika modulo, operasi

logika exclusive or atau XOR juga sangat banyak dipergunakan dalam kriptografi.

Operasi XOR dapat dilihat pada tabel di bawah ini :

Tabel 1. Operasi XOR

a	b	a XOR b
0	0	0
0	1	1
1	0	1
1	1	0

Di dunia teknologi informasi terdapat banyak skema pengenkripsian yang diketahui bekerja dengan baik. Namun bagaimanapun juga semakin banyak yang akhirnya dapat “dikalahkan” oleh para encoder, dan dengan semakin cepatnya pertumbuhan dan perkembangan baik perangkat keras (hardware) dan perangkat lunak (software), pengguna teknologi informasi semakin dicemaskan akan keamanan informasinya dan terkesan bahwa tidak ada satu skema pengenkripsian pun yang tidak dapat di-“crack” oleh pihak-pihak yang tidak bertanggung jawab.

Dalam makalah ini penulis akan berkonsentrasi dalam pembahasan mengenai salah satu teknik kriptografi saja, yakni kriptografi dengan metode Algoritma Vernam yang lebih dikenali dengan istilah kriptografi *“one-time pad”* yang merupakan salah satu jenis teknik kriptografi yang tergolong sederhana namun cukup handal dan bahkan oleh beberapa orang dianggap “mustahil” untuk dipecahkan dalam kondisi tertentu.

2. PENDALAMAN TOPIK

Kriptografi one-time pad merupakan salah satu jenis teknik kriptografi yang menggunakan metode substitusi dengan cara memberikan syarat-syarat khusus terhadap kunci yang digunakan yaitu terbuat dari karakter atau huruf yang acak (kunci acak atau pad), dan pengacakannya tidak menggunakan rumus tertentu. Dengan kata lain one-time pad adalah suatu sistem di mana suatu kunci rahasia yang dibuat acak digunakan hanya sekali untuk mengenkripsi pesan yang kemudian didekripsi lagi dengan kunci yang sama. Metode ini ditemukan oleh Gilbert Vernam pada perang dunia pertama.[2]

Dalam kriptografi klasik, teks sandi dari suatu pesan diperoleh dengan menjumlahkan ataupun mengurangkan teks aslinya terhadap kunci. Sedangkan untuk mendapatkan kembali teks aslinya tersebut dilakukan pengurangan atau penjumlahan teks sandi terhadap kunci tersebut sebagai kebalikan dari proses menyandi. Atau dengan kata lain proses enkripsi data dengan metode one time pad pada zaman dulu belum menggunakan operator logika XOR.

Jika kunci yang digunakan benar – benar acak dan digunakan hanya sekali serta terjaga kerahasiaannya dengan baik, metode one-time pad ini sangat kuat dan tidak dapat dipecahkan. Setiap proses enkripsi adalah unik dan tidak memiliki hubungan satu dengan yang lainnya sehingga tidak ada pola yang dapat dideteksi oleh orang yang ingin mendekripsi pesan yang telah disandikan tersebut

Walaupun demikian, dengan menggunakan one-time pad ini, suatu organisasi atau instansi yang menggunakannya haruslah memiliki akses kepada kunci yang sama yang digunakan untuk mengenkripsi pesan. Hal ini menimbulkan masalah baru yaitu bagaimana caranya agar pihak penerima dapat menerima kunci tersebut atau menyimpan kunci tersebur dengan aman. Biasanya kunci one time pad di-generate ketika suatu kelompok berkumpul pada tempat yang sama dan digunakan setelah pertemuan berakhir. Kunci yang digunakan disebut kunci rahasia atau secret key, karena apabila telah diketahui, pesan sandi yang telah dienkripsikan dapat dengan mudah didekripsikan.

Suatu one-time pad diciptakan dengan men-generate suatu string yang terdiri dari karakter-karakter atau angka-angka yang panjangnya harus minimal sama dengan kata terpanjang dalam pesan yang akan dienkripsikan.[3] String ini di-generate secara acak / random, misalnya dengan menggunakan random number generator pada computer. String tersebut kemudian dituliskan pada suatu pad (atau peralatan apapun yang dapat digunakan atau dibaca). Pad-pad tersebut kemudian diberikan kepada siapapun yang ingin menggunakannya untuk mengirim ataupu menerima pesan. Suatu pad juga dapat dibuat menjadi sekumpulan kunci yang berlaku untuk setiap hari dalam satu bulan, misalnya, di mana setiap kunci akan aus (tidak dapat dipergunakan lagi) pada setiap akhir dari hari tersebut. Dengan demikian keamanan kunci dapat lebih terjamin, sehingga kemungkinan pesan dapat diketahu oleh pihak ketiga akibat kunci yang “bocor” dapat diminimalkan.

Saat suatu pesan akan dikirimkan, pengirim pesan akan menggunakan kunci rahasia / secret key tersebut untuk mengenkripsi setiap karakter dalam pesan yang akan dikirimkan satu per satu. Jika menggunakan computer, setiap bit dalam karakter di-“exclusive-or”kan (di-XOR-kan) dengan bit karakter yang bersesuaian dalam kunci rahasia/secret key. Untuk one-time pad algoritma pengenkripsian data yang digunakan adalah operasi exclusive or (XOR) saja. One time-pad kadang-kadang digabungkan denan algoritma pengenkripsian lainnya seperti RSA atau MD5 untuk memastikan bahwa kunci rahasia yang dipergunakan adalah benar-benar random / acak. Hal ini mengakibatkan semakin sulitnya informasi atau plainteks diperoleh oleh pihak ketiga. Seorang penulis

pernah menyebutnya sebagai “100 % noise source” yang digunakan untuk menutupi pesan yang sebenarnya. Hanya pengirim dan penerima pesan sajalah yang dapat menghilangkan “noise” tersebut. Jika suatu one-time pad telah digunakan, maka tidak dapat dipergunakan lagi karena jika dipergunakan kembali memiliki resiko adanya pihak ketiga yang “mencuri” pesan dan membandingkan pengkodean yang sama untuk kata-kata yang sama dalam pesan-pesan sandi tersebut.

Proses pengenkripsian dengan Algoritma Vernam atau Kriptografi One-time Pad pada dasarnya adalah algoritma exclusive-or yang sedarhana dengan pengimplementasian yang tidak terlalu rumit.

3. HASIL DAN PEMBAHASAN

Konsep terpenting dalam penggunaan kriptografi one-time pad adalah meng-XOR-kan plainteks dengan kunci yang telah dipersiapkan untuk menghasilkan cipherteks.

Secara sederhana dapat dituliskan sebagai berikut :

(1) $c = p \; XOR \; k$ [4]

sedangkan proses pendekripsian dituliskan sbb. :

(2) $p = c \; XOR \; k$ [5]

dengan c : chiperteks
p : plainteks
k : kunci rahasia yang digunakan

Lebih lanjutnya akan dijelaskan di bawah ini :

3.1 Penggunaan exclusive or (XOR) dalam biner

Operator logika XOR akan menghasilkan T (benar) apabila salah satu dari kedua operand (tetapi tidak keduanya) bernilai T.
Apabila diaplikasikan dalam bit maka operator XOR akan menghasilkan 1 jika dan hanya jika salah satu operand bernilai 1.
Contoh :

x	00111010	10101011
y	10100100	01010101
hasil	10011110	11111110

Sedangkan suatu bilangan dalam biner apabila di-XOR-kan dengan dirinya sendiri akan menghasilkan 0.
Contoh :

X	01010101	10101010
Y	01010101	10101010
Hasil	00000000	00000000

Apabila suatu bilangan biner x di XORkan sebanyak 2 kali dengan suatu bilangan biner yang sama maka akan diperoleh bilangan x tersebut kembali.

Contoh :

X	11010101	10001011
Y	01010110	11101010
Hasil1	10000011	01100001

Apabila hasil1 di-XOR-kan kembali dengan Y maka diperoleh :

Hasil1	10000011	01100001
Y	01010110	11101010
Hasil2	11010101	10001011

Ternyata diperoleh hasil2 sama dengan x.

Hal ini merupakan salah satu dasar dalam penerapan algoritma Vernam dalam kriptografi, yaitu suatu string yang diterjemahkan ke dalam biner dapat dienkripsikan dengan suatu kunci tertentu dan dapat pula dengan mudah diperoleh kembali dari pesan sandi dengan menggunakan operator XOR pada kunci yang sama.

3.2 Contoh penggunaan one-time pad

Misalkan pesan yang akan dikirimkan yaitu FIRMAN dengan kata kunci GLORIA.
Langkah yang dilakukan sbb :

a. Ubah menjadi kode ASCII dan biner

Berikut ini diberikan table kode ASCII yang digunakan :

Tabel 2 . ASCII

	0	1	2	3	4	5	6	7
00	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL
01	BS	HT	LF	VT	FF	CR	SO	SI
02	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB
03	CAN	EM	SUB	ESC	FS	GS	RS	US
04	Blank	!	“	#	$	%	&	‘
05	(	)	*	+	,	-	.	/
06	0	1	2	3	4	5	6	7
07	8	9	:	;	<	=	>	?
10	@	A	B	C	D	E	F	G
11	H	I	J	K	L	M	N	O
12	P	Q	R	S	T	U	V	W
13	X	Y	Z	[	\	]	^	_
14	`	a	b	c	d	e	f	G
15	h	i	j	k	l	m	n	O
16	p	q	r	s	t	u	v	W
17	x	y	z	{	|	}	~	DEL

F	106	0001 0000 0110
I	111	0001 0001 0001
R	122	0001 0010 0010
M	115	0001 0001 0101
A	101	0001 0000 0001
N	116	0001 0001 0110

Hal yang sama dilakukan pada kunci

G	107	0001 0000 0111
L	114	0001 0001 0100
O	117	0001 0001 0111
R	122	0001 0010 0010
I	111	0001 0001 0001
A	101	0001 0000 0001

b. Pesan di-XORkan dengan kunci
   Akan diperoleh
   F → 0000 0000 0001→001
   I → 0000 0000 0101→005
   R → 0000 0011 0101→035
   M → 0000 0011 0100→034
   A → 0000 0001 0000→010
   N → 0000 0001 0111→017

c. Kode ASCII tersebut diterjemahkan lagi menjadi karakter
   Diperoleh : NUL ENQ GS FS BS SI
   Untuk memperoleh plainteks kembali, penerima pesan cukup mengubah lagi plainteks menjadi ASCII dan meng-XORkan kembali dengan kunci.

3.3 Contoh Program One-Time Pad dalam bahasa C

Penulis menemukan contoh program ini dalam situs internet, program bekerja untuk mengenkripsikan suatu file teks. Program disalin sesuai dengan code aslinya dalam situs.

Programnya adalah sebagai berikut :

```
/* CRYPTIC.C  V 1.0 Copyright 1998 by Glen E.
Gardner, Jr.  */
/* Encrypts a file using a random key and saves the
key.    */
/* Automatically generates a new key when needed.
The new  */
/* key is deleted on the second use (decryption) to
prevent */
/* accidental reuse of the same key for encryption.
*/

/* This program is freeware, use it freely and enjoy!
*/
/* Be sure to cite the author and include the original
*/
/* source in all distributions.                 */

/* Written and compiled in ANSI C using Borland
C++ V 5.02  */
/* Tested on Windows NT 4.0 and FreeBSD 2.2.5
(using gcc).  */

/* Run this program once to encrypt and agin, using
the     */
/* same key, to decrypt.                        */
/* Any file can be used as a key provided it is the
*/
/* same size (or larger) as the file being encrypted.
*/
/* (small,repeating keys are for whimps)
*/
/* You need to be careful what you use as a key. If
you     */
/* don't believe this, try encrypting a file using a
windows*/
/* dll file as a key and looking at the output with a text
*/
/* editor.                                      */

/* The encrypted output is binary.  You can use
cryptic to  */
/* encrypt any file.                            */


#include<stdio.h>
#include<stdlib.h>
#include<time.h>


/* Use this include with GCC on FreeBSD machines.
*/
/*  #include</usr/include/sys/stat.h>  */

/* Use this include instead of the one above for
Windows NT. */
#include<sys\stat.h>

void makekey(long int,char *);

int main(int argc,char **argv)
{
struct stat statbuf;

time_t  t;
int key;
int data;
int output;
int count=0;
int FLAG=0;
FILE * mykeyfile;
FILE * sourcefile;
FILE * destfile;

if(argc<3)
            {
  printf("CRYPTIC Coyright 1998 by Glen E.
Gardner, Jr.\n");
  printf("USE: CRYPTIC
 <DESTINATION> <KEY>\n");
  return(0);
  }

/* Note that if no key name is given, the program
generates and uses a new key. */
/* Be sure the right key is present when decrypting
(duh).  The program does not*/
/* know if it is encrypting or decrypting.  It just
crunches the source file with*/
/* whatever key it has and spits out the result. */
```

```
/* Bail out if the wrong number of arguments are used.
*/

if(argc>4){printf("Too many arguments.");return(1);}


/* Seed the random number generator for later use. */
 srand((unsigned) time(&t));


 /* get the size of the source file */
   if ((sourcefile = fopen(argv[1], "rb"))== NULL)
   {
   printf("Can't open source file.\n");
   return(4);
   }
   fflush(sourcefile);
   fstat(fileno(sourcefile), &statbuf);
   fclose(sourcefile);

/* Look for default key file if none is given */
if(argv[3]==NULL){argv[3]="newkey";}

/* If the key is not found make a new one. */
if ((mykeyfile = fopen(argv[3], "r"))== NULL)
   {
   FLAG=1;
   printf("Can't open key file.\n");
   printf("Making a new key...\n");
   makekey(statbuf.st_size,"newkey");
   }else{fclose(mykeyfile);}

 /* open all the necessary files. */
   mykeyfile=fopen(argv[3],"rb");
   sourcefile=fopen(argv[1],"rb");
   destfile=fopen(argv[2],"wb");

 /* Use the key to encrypt/decrypt the source file. */
while (count < (statbuf.st_size))
   {
   key=fgetc(mykeyfile);
   data=fgetc(sourcefile);

/* This is all there is to it. */
   output=(key^data);
   /* XOR the data byte once with a byte from a key
and it encrypts. */
   /* XOR the resultant byte again with the same byte
from the same key, and it decrypts. */

   /* write the result to the output file. */
   fputc(output,destfile);
   count++;
   }

/* close the files. */
   fclose(mykeyfile);
   fclose(sourcefile);
   fclose(destfile);

/* Delete the default key on the second time around to
prevent it being reused. */
/* The key is deleted only if a key was not specified
and if the default */
/* key is not new.  */

   if(FLAG==0)
   {
   /* use this for Windows NT */
   system("erase newkey");

   /* use this for FreeBSD */
   /* system("rm newkey"); */
   }
   return(0);
}

/* MAKEKEY() makes a key using random numbers.
*/
/* The random number generator is seeded from the
real time clock.  */
/* It is fairly random, but the nature of the
pseudorandom generator is not */
/* completely random.  This means that a clever
programmer will */
/* eventually crack your key.  */
/* Don't reuse keys, and consider investing time in a
better way of generating */
/* random number strings to use as a key. */

void makekey(long int size,char *name)
{
 int byte;
 int count=0;
 FILE * filein;

 filein=fopen(name,"wb");

while(count&lt;size)
   {
   byte=rand() % 256;
   fprintf(filein,"%c",byte);
   count++;
   }
fclose(filein);
}[ 6]
```

4. KESIMPULAN

Pada akhir makalah ini penulis dapat mengambil beberapa kesimpulan, antara lain :

1. Algoritma Vernam atau One-time pad merupakan algoritma pengenkripsian data dan informasi yang relative sederhana dan mudah digunakan namun cukup aman dalam menjamin kerahasiaan informasi atau data yang ingin dikirimkan oleh pengirim pesan kepada penerima pesan tanpa dapat diketahui oleh pihak lain.

2. Teknik pengenkripsian ini relative sederhana karena hanya menggunakan algoritma XOR dalam pembuatan sandinya.
3. Keamanan algoritma pengenkripsian ini sangat bergantung pada kerahasiaan kunci rahasia (secret key) dan pad yang digunakan baik dalam mengenkripsi maupun mendekripsi data atau informasi, karena walaupun untuk memecahkan sandi yang dibuat sangat sulit namun apabila kunci telah ditemukan akan sangat mudah untuk memecahkan sandi tersebut.
4. Untuk menjamin kerahasiaan data dana informasi serta menjamin keamanan kunci rahasia yang digunakan maka kunci yang di-generate harus benar-benar random atau acak dan hanya dapat dipergunakan sebanyak satu kali saja.

Pengenkripsian One-time pad ini juga memiliki beberapa kelemahan , di antaranya adalah :

1. Panjang kunci rahasia (secret key) yang digunakan dalam pengenkripsian harus paling tidak sama dengan panjang plainteks, hal ini mengakibatkan semakin turunnya tingkat keamanan pengenkripsian apabila plainteks semakin panjang.
2. Jika sebuah kunci telah dipergunakan maka kunci tersebut sudah tidak boleh dipergunakan kembali. Akibatnya jumlah kunci yang masih dapat dipergunakan akan semakin berkurang seiring dengan semakin seringnya penggunaan metode enkripsi ini. Penggunaan kunci yang sama lebih dari satu kali jelas akan berpengaruh buruk bagi kerahasiaan data atau informasi yang dienkripsikan.
3. pertukaran kunci yang terjadi antara pihak pengirim pesan dan penerima pesan juga merupakan titik rentan dalam metode pengenkripsian ini, karena terjadinya kebocoran kunci dapat terjadi pada saat terjadinya pertukaran informasi mengenai kunci rahasia tersebut.

5. REFERENSI

[1] Munir, Rinaldi ,“Diktat Kuliah Matematika IF 2151 Matematika Diskrit ” Edisi Keempat, 2004, hal V-22

[2] http://hadiwibowo.wordpress.com/kriptografi
tanggal 2 januari 2008

[3]http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci213673,00.html
tanggal 2 Januari 2008

[4] [5]
ezine.echo.or.id/ezine14/04_Algoritma_Enkripsi_One_Time_Pad.txt
tanggal 2 Januari 2008

[6]http://www.gmonline.demon.co.uk/cscene/CS4/CS4-03.html
Tanggal 2 Januari 2008