Makalah IF3058

Analisis Kriptografi dalam penentuan Cipherteks kode ASCII melalui metode Aljabar Boolean

Rheno Manggala Budiasa - 13506119

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
e-mail: if16119@students.if.itb.ac.id

ABSTRAK

Penerapan Kriptografi dalam bidang komputer berkembang secara pesat. Mulai dari jaringan hingga perangkat lunak saat ini menggunakan kriptografi sebagai salah satu bentuk pengamanan. Komputer sebagai salah satu produk digital mengaplikasikan kriptografi dalam pembentukan chiperteks melalui kode-kode biner yang diterjemahkan khusus untuk selanjutnya dilihat oleh pengguna sebagai notasi atau karakter tertentu. Sistem Digital menawarkan beberapa metode untuk digunakan sebagai penerapan kriptografi, antara lain operasi dasar bilangan biner (AND, OR, NAND, NOR, XOR, dll) dan operasi-operasi aljabar boolean. Melalui metode-metode tersebut kita dapat menentukan beberapa jenis masukan dan keluaran dari suatu operasi digital yang tentunya erat kaitannya dengan aplikasi dari kriptografi. Sebagai contoh sederhana adalah jika kita mengetik setiap karakter pada *keyboard* maka akan ditampilkan oleh komputer di layar dalam bentuk kode ASCII. Komponen Digital mampu membaca perangkat digital berupa kode-kode biner untuk selanjutnya dilakukkan operasi-operasi tertentu. Beberapa perangkat lain selain komputer juga menerjemahkan kode biner menjadi karakter tertentu misalnya lampu LED tujuh segmen.

Kata kunci: Kriptografi, Jaringan, Perangkat Lunak, pengamanan, komputer, chiperteks, kode biner, sistem digital, aljabar Boolean, *keyboard*, ASCII, Komponen Digital, LED

I. PENDAHULUAN

Sistem Digital berperan penting dalam pembentukan kode ASCII. Kode ASCII yang diterima dari *keyboard* diterjemahkan oleh komputer sebagai deretan angka (terdiri dari 0 dan 1) disebut bit. Serangkaian bit tadi akan membentuk karakter ASCII disebut *byte*. Setiap kode ASCII akan mempunyai jumlah bit yang berbeda-beda. Melalui operasi tertentu seperti aljabar *boolean* kita dapat membuat karakter lain yang bisa digunakan sebagai sebuah chiperteks. Selain pembentukan kode ASCII komputer juga memanfaatkan kriptografi untuk hal lain misalnya enkripsi terhadap paket-paket pengiriman pada jaringan komputer.

Sistem digital banyak menggunakan operasi-operasi tertentu terhadap suatu input pada perangkat tertentu untuk menghasilkan output yang diinginkan. Operasi dasar pada sistem digital adalah AND, OR, NAND, NOR dan XOR. Semua operasi dasar tadi juga dapat digabungkan dengan operasi lain misalnya aljabar Boolean. Komputer sebagai salah satu perangkat digital sangat banyak menggunakan operasi-operasi tadi. Berikut akan dibahas salah satu operasi sistem digital yaitu aljabar Boolean.

II. METODE

Pada tahun 1849 George Boole mempublikasikan suatu skema aljabar yang melibatkan terkaan logika. Hampir 100 tahun setelah itu tepatnya tahun 1930 seorang ilmuan bernama Claude Shannon memperlihatkan bahwa aljabar boolean dapat membuat rangkaian dengan *switch*. Karena itu aljabar ini ini sering digunakan pada rangkaian logika.

II.1 Algoritma Kriptografi Klasik

Algoritma Kriptografi klasik merupakan algoritma yang berbasasis karakter. Algoritma ini merupakan basis dari Kriptografi modern. Algoritma Kriptografi klasik dibagi menjadi 2 bagian, yaitu :

1. *Cipher* Subsritusi (*Substitution Ciphers*)
2. *Cipher* Transposisi (*Transposition Ciphers*)

Cipher Substitusi merupakan algoritma kriptografi yang mengganti setiap karakter plainteks dengan karakter lain.

Cipher Substitusi dibagi menjadi 3 jenis, yaitu :

1. *Cipher* Abjad tunggal (*monoalphabetic cipher*)

   Satu huruf di plainteks diganti dengan satu huruf yang bersesuaian. Jumlah kemungkinan susunan huruf-huruf cipherteks yang dapat dibuat adalah sebanyak

   26! = 403.291.461.126.605.635.584.000.000
   Contoh: *Caesar Cipher*

   Tabel substitusi dapat dibentuk secara acak:

   Plainteks : A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
   Cipherteks: **D I Q M T B Z S Y K V O F E R J A U W P X H L C N G**

   Atau dengan kalimat yang mudah diingat:

   Contoh: we hope you enjoy this book
   Buang duplikasi huruf: wehopyunjtisbk
   Sambung dengan huruf lain yang belum ada:
   wehopyunjtisbkacdfglmqrvxz
   Tabel substitusi:

   Plainteks : A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
   Cipherteks: **W E H O P Y U N J T I S B K A C D F G L M Q R V X Z**

2. *Cipher* Substitusi Homofonik (*Homophonic Substitution Cipher*)

   Setiap huruf plainteks dipetakan ke dalam salah satu huruf cipherteks yang mungkin.

   Tujuan: menyembunyikan hubungan statistik antara plainteks dengan cipherteks

   Fungsi *ciphering* memetakan satu-ke-banyak (*one-to-many*).

   Misal: huruf E → **AB**, **TQ**, **YT,UX** (homofon)
   huruf B → **EK**, **MF**, **KY** (homofon)

3. *Cipher* abjad-majemuk (*Polyalpabetic substitution cipher*)

   Cipher abjad-tunggal: satu kunci untuk semua huruf palinteks

   Cipher substitusi-ganda: setiap huruf menggunakan kunci berbeda.

   Cipher abjad-majemuk dibuat dari sejumlah *cipher* abjad-tunggal, masing-masing dengan kunci yang berbeda.

   Kebanyakan *cipher* abjad-majemuk adalah *cipher* substitusi periodik yang didasarkan pada periode *m*.

   Contoh: (spasi dibuang)
   P:
   KRIPTOGRAFIKLASIKDENGANCIPHERALFABETMAJEMUK

   K:
   LAMPIONLAMPIONLAMPIONLAMPIONLAMPIONLAMPIONL

   C : **VR**...

   Perhitungan:

   (K + L) mod 26 = (10 + 11) mod 26 = 21 = V
   (R + A) mod 26 = (17 + 0) mod 26 = 17 = A
   dst

   Contoh 2: (dengan spasi)
   P:
   SHE SELLS SEA SHELLS BY THE SEASHORE
   K:
   KEY KEYKE YKE YKEYKE YK EYK EYKEYKEY
   C:
   CLC CIJVW QOE QRIJVW ZI XFO WCKWFYVC

4. Cipher substitusi poligram (*Polygram substitution cipher*)

   Blok huruf plainteks disubstitusi dengan blok cipherteks.

   Misalnya AS diganti dengan RT, BY diganti dengan SL

   Jika unit huruf plainteks/cipherteks panjangnya 2 huruf, maka ia disebut digram (*biigram*), jika 3 huruf disebut ternari-gram, dst

Tujuannya: distribusi kemunculan poligram menjadi *flat* (datar), dan hal ini menyulitkan analisis frekuensi

Cipher Tansposisi yaitu algoritma yang kriptografi yang mengubah posisi huruf di dalam plainteks. Dengan kata lain algoritma ini melakukan *transpose* terhadap rangkaian huruf di dalam plain teks. Nama lain untuk metode ini adalah permutasi, karena *transpose* setiap karakter di dalam teks sama dengan mempermutasikan karakter-karakter tersebut.

Contoh: Misalkan plainteks adalah
DEPARTEMEN TEKNIK INFORMATIKA ITB

Enkripsi:
DEPART
EMENTE
KNIKIN
FORMAT
IKAITB

Cipherteks: (baca secara vertikal)

DEKFIEMNOKPEIRAANKMIRTIATTENTB
DEKF IEMN OKPE IRAA NKMI RTIA TTEN TB

Dekripsi: Bagi panjang cipherteks dengan kunci.
(Pada contoh ini, 30 / 6 = 5)

DEKFI
EMNOK
PEIRA
ANKMI
RTIAT
TENTB

Plainteks: (baca secara vertikal)
DEPARTEMEN TEKNIK INFORMATIKA ITB

II.2 Aljabar Boolean

Seperti pada aljabar lainnya (misalnya aljabar linier dalam matematika), aljabar Boolean juga merupakan sekumpulan aturan yang diturunkan dari asumsi-asumsi dasar. Kumpulan asumsi ini dinamakan aksioma.

II.2.1 Aksioma Aljabar Boolean

1.a 0.0 = 0
1.b 1+1 = 1
2.a 1.1 = 1
2.b 0 + 0 = 0
3.a 0.1 = 1.0 = 0
3.b 1+0 = 0+1 = 1
4.a If x = 0, then $\bar{x} = 1$
4.b If x = 1, then $\bar{x} = 0$

Teorema satu variabel

5.a x.0 = 0
5.b x+1 = 1
6.a x.1 = x
6.b x+0 = x
7.a x.x = x
7.b x+x = x
8.a $x.\bar{x} = 0$
8.b $x+\bar{x} = 1$

9. $\bar{\bar{x}} = 1$

Teorema dua dan tiga variabel

10.a	x.y = y.x	(Komutatif)
10.b	x+y = y+x	
11.a	x.(y.z) = (x.y).z	(Asosiatif)
11.b	x+(y+z) = (x+y)+z	
12.a	x.(y+z) = x.y + x.z	(Distributif)
12.b	x+y.z = (x+y).(x+z)	
13.a	x+x.y = x	(Absorbsi)
13.b	x.(x+y) = x	
14.a	$x.y+x.\bar{y} = x$	(Kombinasi)
14.b	$(x+y).(x+\bar{y}) = x$	
15.a	$\overline{x.y} = \bar{x} + \bar{y}$	(De Morgan)
15.b	$\overline{x+y} = \bar{x}.\bar{y}$	
16.a	$x+\bar{x}.y = x+y$	
16.b	$x.(\bar{x}+y) = x.y$	
17.a	$x.y +y.z+\bar{x}.z = x.y + \bar{x}.z$	(Konsensus)
17.b	$(x+y).(y+z).(\bar{x}+z) = (x+y).(\bar{x}+z)$	

Operasi-operasi lain pada Aljabar Boolean :

X	Y	X AND Y	X OR Y	X NAND Y	X NOR Y	X XOR Y
0	0	0	0	1	1	0
0	1	0	1	1	0	1
1	0	0	1	1	0	1
1	1	1	1	0	0	0

II.2.2 Kode ASCII

ASCII (*American Standard Code for Information Interchange*) merupakan suatu standard internasional dalam kode huruf dan symbol seperti Hex dan Unicode tetapi ASCII lebih bersifat universal, contohnya 124 adalah untuk karakter "|". Ia selalu digunakan oleh komputer dan alat komunikasi lain untuk menunjukkan teks. Kode ASCII sebenarnya memiliki komposisi bilangan biner sebanyak 8 bit. Dimulai dari 0000 0000 hingga 1111 1111. Total kombinasi yang dihasilkan sebanyak 256, dimulai dari kode 0 hingga 255 dalam sistem bilangan Desimal

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.asciitable.com

Kode ASCII (Lengkap)

128	Ç	144	É	160	á	176	░	193	┴	209	╤	225	ß	241	±
129	ü	145	æ	161	í	177	▒	194	┬	210	╥	226	Γ	242	≥
130	é	146	Æ	162	ó	178	▓	195	├	211	╙	227	π	243	≤
131	â	147	ô	163	ú	179	|	196	─	212	╘	228	Σ	244	⌠
132	ä	148	ö	164	ñ	180	┤	197	┼	213	╒	229	σ	245	⌡
133	à	149	ò	165	Ñ	181	╡	198	╞	214	╓	230	µ	246	÷
134	å	150	û	166	ª	182	╢	199	╟	215	╫	231	τ	247	≈
135	ç	151	ù	167	º	183	╖	200	╚	216	╪	232	Φ	248	°
136	ê	152	_	168	¿	184	╕	201	╔	217	┘	233	Θ	249	·
137	ë	153	Ö	169	_	185	╣	202	╩	218	┌	234	Ω	250	·
138	è	154	Ü	170	¬	186	║	203	╦	219	▉	235	δ	251	√
139	ï	156	£	171	½	187	╗	204	╠	220	▄	236	∞	252	_
140	î	157	¥	172	¼	188	╝	205	═	221	▌	237	ϕ	253	²
141	ì	158	_	173	¡	189	╜	206	╬	222	▐	238	ε	254	■
142	Ä	159	ƒ	174	«	190	╛	207	╧	223	▀	239	∩	255	
143	Å	192	└	175	»	191	┐	208	╨	224	α	240	≡		

Source: www.asciitable.com

Kode ASCII (*Extended*)

III. Proses

Kode digital (biner) diterjemahkan oleh komputer secara per satuan byte (1 byte = 8 bit). Secara umum semua komputer saat ini mampu menerjemahkan kode-kode biner tersebut ke dalam suatu karakter tertentu (Lihat Kode ASCII). Kemampuan komputer ini dapat kita manfaatkan untuk mengubah setiap karakter yang dapat digunakan sebagai chiperteks. Ada banyak metode dalam kriptografi sebenarnya untuk menentukan chiperteks. Kita telah mengenal teorinya pada bab Algoritma Kriptografi Klasik. Namun pembahasan kali ini akan lebih ditekankan peran dari Aljabar Boolean dalam penentuan kode biner ASCII.

Hampir semua bahasa pemrograman sudah mampu menerjemahkan kode ASCII secara langsung. Misalnya Java sudah mampu menerjemahkan 26 karakter alfabet secara langsung menjadi desimal dalam format hexadesimal (8 bit). Sebagai contoh berikut : nilai pada kode ASCII adalah 97 desimal dan 01100001 dalam biner.

```
public class kodeASCII{
  public static void main(String[] args)
  {
    char ch = 'a';
    for(int i=0;i<26;i++)
    {
System.out.println(""+ch+" = "+(int)ch);
        ch++;
    }
  }
  }
```

Kode Program Java Menerjemahkan Karakter menjadi desimal

```
a = 97
b = 98
c = 99
d = 100
e = 101
f = 102
g = 103
h = 104
i = 105
j = 106
k = 107
l = 108
m = 109
n = 110
o = 111
p = 112
q = 113
r = 114
s = 115
t = 116
u = 117
v = 118
w = 119
x = 120
y = 121
z = 122
```

Hasil Eksekusi Program Java Menerjemahkan Karakter menjadi kode biner

Dari kode program di atas kita melihat setiap karakter di tambahkan (increment) satu bit sehingga menghasilkan satu karakter. Proses detailnya yaitu sebagai berikut :

Karakter ‘a’ dalam biner sama dengan 01100001
Secara logik komputer akan melakukan proses penambahan 1 bit setiap proses *increment*.

```
  01100001
         1
  --------+
  01100010  =  98 desimal = karakter ‘b’
```

Dengan cara seperti di atas kita dapat menentukan secara sederhana bagaimana menentukan chiperteks. Chiperteks yang dimaksud adalah pergeseran karakter karena penambahan jumlah bit.

Contoh :

Plainteks : rheno
Chiperteks : tjgpq

Dari hasil di atas terlihat bahwa chiperteks dibuat dengan menggeser 2 bit setiap karakter. Sehingga aljabar Booleannya adalah sebagai berikut :

'r' = 114 desimal = 01110010

01110010 01110011
1 1

——————+ ⇢ ——————+
01110011 01110100 = 116 desimal ='t'

'h' = 104 desimal = 01101000

01101000 01101001
1 1

——————+ ⇢ ——————+
01101001 01101010 = 106 desimal ='j'

Dan seterusnya…

Sehingga didapatkan chiperteks tjgpq

Operasi ini berlangsung terus menerus sampai karakter akhir plainteks.

Contoh lain adalah penggunaan salah satu operasi teorema aljabar Boolean. Misalnya kita ambil 16.a (Sub-Bab II.2 Aljabar Boolean). Kita umpamakan x dan y adalah suatu karakter. Pertama-tama kita OR-kan semua kode biner (x dengan $\bar{x}$) pada setiap karakter plainteks setelah itu setelah itu hasilnya dilakukan operasi AND (dengan y).

Contoh :

Plainteks : rheno

Misalkan kita ambil karakter pertama 'r' sebagai x. Selanjutnya karakter tersebut akan kita ambil karakter lain sebagai kunci (misalnya 'a' sebagai y). Maka proses pembentukan Chiperteksnya adalah :

r = 114 desimal = 01110010 $\bar{r}$ = 10001101
a = 97 desimal = 01100001

r OR $\bar{r}$ = 01110010
10001101
——————OR
11111111

(r OR $\bar{r}$) AND a = 11111111
10001101
——————AND
10001101

10001101 = 141 desimal = karakter 'ì'

Pada kode ASCII 10001101 merupakan karakter ì.

Untuk karakter selanjutnya kita bisa menggunakan karakter lain sebagai kunci misal 'b'.

h = 104 desimal = 01101000 $\bar{h}$ = 10010111
b = 98 desimal = 01100010

h OR $\bar{h}$ = 01101000
10010111
——————OR
11111111

(h OR $\bar{h}$) AND b = 11111111
01100010
——————AND
01100010

10001101 = 142 desimal = karakter 'Ä'

Pada kode ASCII 10001101 merupakan karakter Ä.

Dan seterusnya…

Sehingga kita bisa memperoleh chiperteks :
ìÄǺÉæ untuk kunci abcde

Kita juga bisa membuat chiperteks lain dari Aljabar Boolean yang lain sesuai dengan teorema yang sudah dijelaskan pada bab 2. Kita juga bisa membuat variasi terhadap seluruh aljabar Boolean dan tentunya dengan kunci yang bervariasi pula.

IV. KESIMPULAN

Beberapa kesimpulan yang dapat diambil adalah :

- Aljabar Boolean dapat memberikan konsep baru dalam menentukan chiperteks.
- Kunci yang dihasilkan oleh aljabar Boolean bisa dimodifikasi sehingga menghasilkan kode ASCII

yang *extended* atau dengan kata lain akan dihasilkan karakter lain yang tentunya akan sangat bervariasi.
- Dapat dihasilkan chiperteks yang bervariasi, karena mempunyai kunci yang bervariasi pula.
- Metode aljabar Boolean ini merupakan pengembangan dari algoritma kriptografi klasik.
- Dapat digunakan pada chip digital untuk enkripsi terhadap keamanan data

REFERENSI

[1] http://id.wikipedia.org/wiki/ASCII

[2] Staphen Brown and Zvonko Vranesic, *Fundamentals of Digital Logic with VHDL Design*, *Second Edition*, 2005, McGraw Hill Higher Education

[3] http://www.asciitable.com/

[4] Rinaldi Munir, Slide Kuliah Kriptografi, Teknik Informatika, Sekolah Teknik Elektro dan Informatika ITB 2006