STUDI PENGEMBANGAN *THUE MORSE* CIPHER BLOCK CHAINING

Dian Perdhana Putra

Teknik Informatika ITB
Jl. Ganesha 10 Bandung
e-mail: if17096@students.if.itb.ac.id

ABSTRAK

Algoritma kriptografi modern tidak berbasis karakter, melainkan menggunakan basis bit. Pada algoritma kriptografi modern, digunakan prinsip *block cipher* dimana operasi enkripsi dan dekripsi dilakukan terhadap blok-blok biner berukuran n-bit. *Block cipher* beroperasi dalam blok-blok, di mana setiap blok memilki panjang yang sama dengan blok lainnya, misalnya 64 bit atau 128 bit. Muncul permasalahan dalam *block cipher* ini, yaitu untuk setiap pesan plainteks, setiap blok yang dienkripsi dengan kunci yang sama akan menghasilkan output yang sama. Oleh karena itu diciptakanlah *mode of operation* yang menjamin *confidentiality* dari pesan. Salah satu mode operasi yang ada dalam operasi *block cipher* adalah *cipher block chaining*.

Pada *cipher block chaining*, operasi pada suatu blok akan mempengaruhi blok lainnya. Hal ini membuat mode operasi *cipher block chaining* cukup kuat. Makalah ini akan membahas bagaimana menambah keamanan dari mode operasi *cipher block chaining* dengan menggunakan barisan *Thue Morse* (*Thue Morse Sequence*) dengan menggunakan karakter ASCII Extended 256.

Kata kunci: *block cipher*, *cipher block chaining*, *Thue Morse Sequence*.

1. DASAR TEORI

Kriptografi, berasal dari bahasa Yunani, *krýpto*, yang bermakna tersembunyi dan *gráfo*, yang bermakna menulis, adalah ilmu dan seni dalam menyembunyikan pesan atau informasi. Kriptografi merupakan bagian penting dalam ilmu matematika dan sains komputer.

Kriptografi memiliki dua buah proses utama yaitu proses enkripsi dan dekripsi. Enkripsi adalah mengubah plainteks (teks awal) menjadi cipherteks (teks yang menyembunyikan pesan). Dekripsi adalah proses simetri dengan enkripsi, di mana pada proses dekripsi, cipherteks diubah menjadi bentuk awal. Algoritma yang mencakup proses enkripsi dan dekripsi disebut sebagai *cipher*.

Proses enkripsi dapat dinyatakan sebagai berikut :

$$E(P) = C \qquad (1)$$

Dalam proses enkripsi, fungsi enkripsi E memetakan plainteks P ke cipherteks C. Sedangkan, proses dekripsi dapat dinyatakan sebagai berikut :

$$D(C) = P \qquad (2)$$

Dalam proses dekripsi, fungsi dekripsi D memetakan cipherteks C ke plainteks awal P. Sementara, fungsi *cipher* yang merupakan komposisi antara proses enkripsi dan dekripsi dapat ditulis sebagai berikut

$$D(E(P)) = P \qquad (3)$$

Dalam fungsi *cipher*, terlihat bahwa proses dekripsi dari hasil enkripsi sebuah plainteks P akan menghasilkan plainteks semula.

1.1 *Cipher Block Chaining*

Dalam kriptografi modern, dikenal mode operasi yang disebut *block cipher*. *Block cipher* beroperasi dalam blok-blok, di mana setiap blok memilki panjang yang sama dengan blok lainnya, misalnya 64 bit atau 128 bit. Muncul permasalahan dalam *block cipher* ini, yaitu untuk setiap pesan plainteks, setiap blok yang dienkripsi dengan kunci yang sama akan menghasilkan output yang sama. Oleh karena itu diciptakanlah *mode of operation* yang menjamin *confidentiality* dari pesan. Salah satu jenis dari *mode of operation* ini adalah *cipher block chaining* atau CBC.

Cipher block chaining diciptakan oleh IBM pada tahun 1976. Pada CBC *mode*, operasi XOR diterapkan antara blok plainteks dengan blok cipherteks sebelumnya sebelum dilakukan enkripsi dengan fungsi enkripsi tertentu. Oleh karena itu, setiap blok cipherteks tergantung kepada blok-blok cipherteks sebelumnya.

Untuk blok plainteks pertama, di mana sebelumnya tidak ada blok cipherteks, digunakan sebuah blok *dummy*, yang disebut dengan *initialization vector* (*IV*). IV dapat dibangkitkan oleh pengguna atau dibuat secara acak dengan program. Karena IV dianggap sebagai blok cipherteks, IV tidak perlu dirahasiakan, namun sebaiknya tidak dapat diprediksi.

Untuk plainteks dengan jumlah blok sebanyak m , CBC *mode* memiliki operasi sebagai berikut untuk fungsi enkripsi :

$$C_i = \begin{cases} IV\,, i = 0 \\ E(P_i \oplus C_{i-1})\,, i = 1,2,\dots m \end{cases} \qquad (4)$$

dimana :
C_i : Blok cipherteks ke- i $(i = 0,1,2,\dots,m)$
P_i : Blok plainteks ke- i $(i = 0,1,2,\dots,m)$
IV : *Initialization Vector*
E : Fungsi enkripsi yang digunakan untuk mengenkripsi blok.

Sedangkan, untuk plainteks dengan jumlah blok sebanyak m , fungsi dekripsinya adalah :

$$C_0 = IV$$

$$P_i = D(C_i) \oplus C_{i-1})\,, i = 1,2,\dots m \qquad (5)$$

dimana :
C_i : Blok cipherteks ke- i $(i = 0,1,2,\dots,m)$
P_i : Blok plainteks ke- i $(i = 0,1,2,\dots,m)$
IV : *Initialization Vector*
D : Fungsi dekripsi yang digunakan untuk mendekripsi blok.

1.2 *Thue Morse Sequence*

Dalam papernya tahun 1912, seorang matematikawan Norwegia, Axel Thue bertanya mungkinkan membuat sebuah barisan bilangan biner tak hingga yang tidak memiliki cube (tiga buah bit yang sama berdekatan, 111 atau 000) dan overlapping square (instans dari 0X0X0 atau 1X1X1). Thue membuat barisan biner yang memenuhi pertanyaan tersebut:

$\boldsymbol{t} =$ 011010011001011010011001101001 ...

Prouhet-Thue-Morse sequence didefinisikan sebagai berikut [2]:

Definisi 1. *We denote by* $\boldsymbol{t} = (t_n)_{n\geq 0}$ *the Prouhet-Thue-Morse sequence over* $\{0,1\}$*, defined recursively by* $t_0 = 0$ *and* $t_{2n} = t_n$, $t_{2n+1} = \overline{t_n}$ *for all* $n \geq 0$*, where, for* $x \in \{0,1\}$ *, we define* $\bar{x} = 1 - x$.

Thue-Morse sequence dapat dilihat sebagai sekuens bit yang didefinisikan secara rekursif menggunakan operasi *bitwise negation*. Misalkan sebagai contoh, elemen pertama adalah 0. Maka untuk membentuk *Thue Morse sequence* , langkah-langkahnya adalah :

1. Mulai dengan `0`.
2. *Bitwise negation* dari `0` adalah `1`.
3. Gabungkan kedua elemen yang telah didapat tersebut, cari bitwise negationnya, yaitu `10`.
4. Maka empat elemen pertama dari *sequence* ini adalah `0110`.
5. Lakukan operasi *bitwise negation* juga ke empat elemen pertama.
6. Maka delapan elemen pertama adalah `01101001`.
7. Dan seterusnya.

Thue Morse sequence yang akan didapatkan akan menjadi seperti ini :

$\boldsymbol{t} =$ 01101001100101101001011001101001100101100110100101101001

1.3 ASCII

ASCII, kependekan dari *American Standard Code for Information Interchange*, merupakan karakter encoding yang didasarkan pada karakter dalam bahasa Inggris. ASCII digunakan untuk merepresentasikan teks dalam bentuk digital dan dipakai dalam media-media elektronik yang menggunakan teks.

ASCII memiliki 128 karakter, dimana 33 karakter di dalamnya merupakan karakter non cetak, yang merupakan *control character* (karakter yang diciptakan untuk mengatur *device*, misalnya *printer*). Sisanya, merupakan karakter cetak, dan *space* (spasi) dianggap sebagai *invisible graphic*.

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 (beil)	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

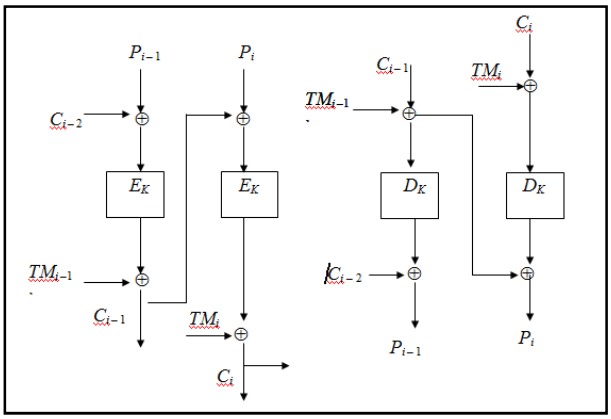
Gambar 1 : Tabel ASCII 128 Bit
(Sumber : www.LookupTables.com)

2. PENGGUNAAN THUE MORSE SEQUENCE DALAM CBC

Untuk menyembunyikan kunci agar tidak memiliki pola yang sama untuk setiap blok, penulis menggunakan blok-blok biner dari *Thue Morse Sequence*. Misalkan blok-blok biner tersebut berukuran 8 bit, maka :

$TM =$
01101001 10010110 10010110 01101001 …

Kemudian, lakukan operasi XOR antara blok TM dengan blok cipherteks sehingga skemanya menjadi seperti ini :



Gambar 2 : Skema Enkripsi dan Dekripsi CBC menggunakan *Thue Morse Sequence*

Untuk plainteks dengan jumlah blok sebanyak m , metode enkripsi dan dekripsinya menjadi seperti ini :

Enkripsi :

$$C_i = \begin{cases} IV\,, i = 0 \\ \left(E(P_i \oplus C_{i-1})\right) \oplus TM_i\,, i = 1,2,\dots m \end{cases} \quad (6)$$

dimana :
C_i : Blok cipherteks ke- i $(i = 0,1,2,\dots,m)$
P_i : Blok plainteks ke- i $(i = 0,1,2,\dots,m)$
IV : *Initialization Vector*
E : Fungsi enkripsi yang digunakan untuk mengenkripsi blok.
TM_i : Blok *Thue Morse Sequence* ke- i $(i = 0,1,2,\dots,m)$
Dekripsi :

$$C_0 = IV$$

$$P_i = D(C_i \oplus TM_i) \oplus C_{i-1}\,, i = 1,2,\dots m \quad (7)$$

dimana :
C_i : Blok cipherteks ke- i $(i = 0,1,2,\dots,m)$
P_i : Blok plainteks ke- i $(i = 0,1,2,\dots,m)$
IV : *Initialization Vector*
D : Fungsi dekripsi yang digunakan untuk mendekripsi blok.
TM_i : Blok *Thue Morse Sequence* ke- i $(i = 0,1,2,\dots,m)$

Contoh :

Misalkan kita memiliki plain teks :

`RINDU`

Bagi menjadi blok-blok biner 8 bit (berdasar karakter ASCII 128 bit) :

`01010010 01001001 01001110 01000100 01010101`

Misalkan, kita memiliki fungsi enkripsi dan dekripsi yaitu melakukan operasi XOR dengan kunci. Kunci yang kita pilih adalah `'K'`(blok binernya : `01001011`).

Misalkan IV yang dipilih adalah `11110000`.

Maka, proses enkripsinya adalah sebagai berikut :

C1 diperoleh sebagai berikut :
P1 ⊕ C0 = 01010010 ⊕ 11110000 = 10100010

Kemudian, hasil yang diperoleh dienkripsi dengan fungsi E sebagai berikut :
10100010 ⊕ *K* = 10100010 ⊕ 01001011 = 11101001

Setelah itu, lakukan fungsi XOR dengan blok *Thue Morse Sequence*.

11101001 ⊕ 01101001 = 10000000

Maka C1 adalah : 10000000

C2 diperoleh sebagai berikut :
P2 ⊕ C1 = 01001001 ⊕ 10000000 = 11001001

Kemudian, hasil yang diperoleh dienkripsi dengan fungsi E sebagai berikut :
11001001 ⊕ *K* = 11001001 ⊕ 01001011 = 10000010

Setelah itu, lakukan fungsi XOR dengan blok *Thue Morse Sequence*.

10000010 ⊕ 10010110 = 00010100

Maka C2 adalah : 00010100

C3 diperoleh sebagai berikut :
P3 ⊕ C2 = 01001110 ⊕ 00010100 = 01011010

Kemudian, hasil yang diperoleh dienkripsi dengan fungsi E sebagai berikut :
10100010 ⊕ *K* = 01011010 ⊕ 01001011 = 00010001

Setelah itu, lakukan fungsi XOR dengan blok *Thue Morse Sequence*.

11101001 ⊕ 10010110 = 01111111

Maka C3 adalah : 01111111

C4 diperoleh sebagai berikut :
P4 ⊕ C3 = 01000100 ⊕ 01111111 = 00111011

Kemudian, hasil yang diperoleh dienkripsi dengan fungsi E sebagai berikut :
00111011 ⊕ *K* = 00111011 ⊕ 01001011 = 01110000

Setelah itu, lakukan fungsi XOR dengan blok *Thue Morse Sequence*.

01110000 ⊕ 01101001 = 00011001

Maka C4 adalah : 00011001

C5 diperoleh sebagai berikut :
P5 ⊕ C4 = 01010101 ⊕ 00011001 = 01001100

Kemudian, hasil yang diperoleh dienkripsi dengan fungsi E sebagai berikut :
01001100 ⊕ *K* = 01001100 ⊕ 01001011 = 00000111

Setelah itu, lakukan fungsi XOR dengan blok *Thue Morse Sequence*.

00000111 ⊕ 10010110 = 10010001

Maka C5 adalah : 10010001

Hasil enkripsinya adalah :

10000000 00010100 01111111 00011001 10010001
Atau dalam bentuk teks :

Ç æ

3. IMPLEMENTASI

Penulis mengimplementasikan CBC dengan *Thue Morse Sequence* ke dalam bahasa Java. Berikut adalah potongan *source code*-nya.

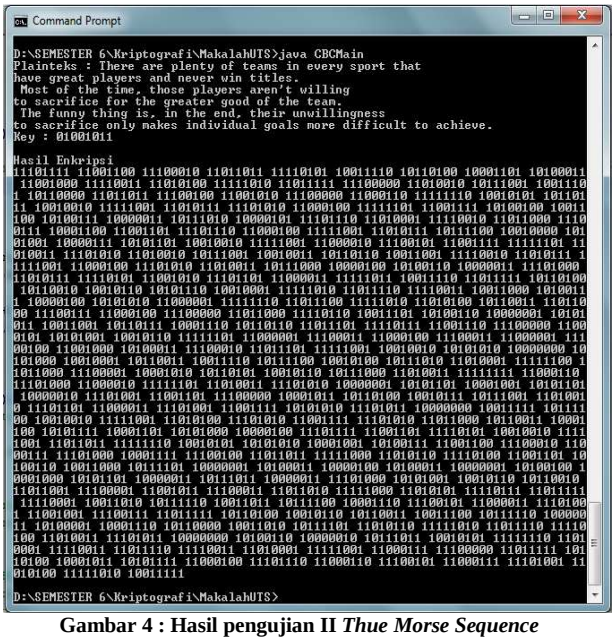
```
//Fungsi enkripsi dengan
//menggunakan Thue Morse Sequence

public static String encryptCBC
        (String key, String plainteks){

        String cipherteks="";
        String thuemorse="";
        int i=0;
        int j=0;

//variabel penyimpan jumlah blok
        int jumlahPlain = 0;

//membuat jumlah blok plainteks
//berdasarkan plainteks yang dimasukkan

    if((plainteks.length()%8)==0){
        jumlahPlain = plainteks.length()/8;
    }
    else{
        jumlahPlain =
               (plainteks.length()/8) +1;

        while(plainteks.length()%8!=0){
                plainteks+=" ";
                }
    }

//membuat blok seukuran 64 bit
//sebanyak jumlah blok
        String[] blokPlain =
                  new String[jumlahPlain];

//membuat thue morse sequence
        thuemorse =
              ThueMorseSequence(jumlahPlain);
        String[] blokTM =
                  new String[jumlahPlain];


i=0;
j=0;
//memasukkan plainteks
//ke dalam blok-blok 8 bit
        while(i<plainteks.length()){
           if((i%8)==0){
              blokPlain[j] =
                  plainteks.substring(i,i+8);
               j++;
           }
           i++;
        }


i=0;
j=0;
//memasukkan thue morse yang telah dibentuk
//ke dalam blok-blok 8 bit
        while(i<thuemorse.length()){
            if((i%8)==0){
                blokTM[j] =
                    thuemorse.substring(i,i+8);
                j++;
            }
            i++;
        }

//buat blok-blok cipherteks
        String[] blokCipher =
                  new String[jumlahPlain];

        String temp="";
        //proses enkripsi
        for(i=0;i<jumlahPlain;i++){
            if(i==0){
                temp ="";
                temp =
                  XOR(blokPlain[i],"11110000");
                blokCipher[i] =
                          encrypt(key,temp);
                blokCipher[i] =
                  XOR(blokTM[i],blokCipher[i]);
            }
            else{
                temp ="";
                temp =
               XOR(blokPlain[i],blokCipher[i-1]);
                blokCipher[i] =
                          encrypt(key,temp);
                blokCipher[i] =
                  XOR(blokTM[i],blokCipher[i]);
            }
        }

        for(i=0;i<jumlahPlain;i++)
                cipherteks += blokCipher[i];

return cipherteks;
}

//membuat thue morse sequence
//sepanjang bilangan tertentu dikali dengan
//delapan
public static String ThueMorseSequence(int i){
        String thuemorse="0";
        String negasi="";
        String temp="";
        int jumlah = i*8;

        while (thuemorse.length()<jumlah){
                negasi="";
                negasi =
                   bitwiseNegation(thuemorse);
                thuemorse+=negasi;
        }

return thuemorse;
}
```

```
//melakukan operasi bitwise negation
public static String bitwiseNegation
                    (String biner){
        String ret="";

        for(int i=0;i<biner.length();i++){

             if(biner.charAt(i)=='0'){
                ret+="1";
             }
             else if(biner.charAt(i)=='1'){
                ret+="0";
             }
        }

return ret;
}

//fungsi untuk operasi XOR
public static String XOR
          (String satu, String dua){
        String result="";

        //samakan panjang
        if(satu.length()>dua.length()){

           while(satu.length()>dua.length()){
              dua += " ";
            }
        }
        else{

           while(dua.length()>satu.length()){
              satu += " ";
            }
        }

        for(int i=0;i<satu.length();i++){

            if(satu.charAt(i)==dua.charAt(i)){
               result += '0';
            }
            else{
               result += '1';
            }
        }

return result;
}
```

4. PENGUJIAN

Penulis membuat program untuk menguji penggunaan *Thue Morse Sequence* dalam cipher block chaining. Hasil pengujian ditampilkan dalam biner, sebab ada beberapa karakter yang tidak dapat ditampilkan di layar dalam bentuk karakter ASCII.

Gambar 3 : Hasil pengujian I *Thue Morse Sequence*



Gambar 4 : Hasil pengujian II *Thue Morse Sequence*

Penggunaan *Thue Morse Sequence* dalam *cipher block chaining* dapat menambah kekuatan pada *cipher block chaining*. Sebelum diikutkan ke dalam operasi berikutnya, blok cipher akan di XOR dengan suatu blok biner dari *Thue Morse* Sequence. Karena *Thue Morse Sequence* sendiri merupakan suatu barisan bilangan palindrom dimana elemen ke n akan berulang di elemen ke 2n, maka hal ini akan menyebabkan hasil cipherteks menjadi semakin sulit untuk ditebak.

5. KESIMPULAN

Dari hasil analisis berbagai studi kasus yang telah dilakukan, penulis mengambil kesimpulan sebagai berikut :

1. Keamanan mode operasi *cipher block chaining* dapat ditingkatkan dengan berbagai cara, salah satunya dengan menambahkan metode untuk mengenkripsi hasil *cipher* dari suatu blok plainteks.
2. Penggunaan barisan bilangan dapat digunakan untuk membantu menyembunyikan hasil *cipher* dari suatu blok plainteks, salah satunya adalah *Thue Morse Sequence*.
3. Penggunaan barisan bilangan yang digunakan sebaiknya yang memiliki pola tertentu, namun sebaiknya tidak terlalu jelas sehingga mudah ditebak, atau terlalu rumit sehingga tidak mudah untuk dilakukan proses dekripsi. *Thue Morse Sequence* dipilih karena memenuhi kriteria pola yang tidak terlalu jelas dan tidak terlalu rumit.

6. REFERENSI

[1] Munir, Rinaldi, *Diktat Kriptografi*, Teknik Informatika ITB, 2010

[2] Ding, C, et al, *Sequences and their Applications: Proceedings of SETA '98*, Springer, 1999

[3] Mao, Wenbo, *Modern Cryptography: Theory and Practice*, Prentice Hall PTR, 2003

[4] Bishop, David, *Introduction to Cryptography with Java™ Applets*, Jones and Barness Publisher, 2003

[5] Mao, Wenbo, *Modern Cryptography: Theory and Practice*, Prentice Hall PTR, 2003

[6] Vaudenay, Serge, *A classical introduction to cryptography: applications for communications*, Springer, 2006

[7] http://mathworld.wolfram.com/Thue-MorseSequence.html diakses pada tanggal 24 Maret 2010

[8] www.cs.uwaterloo.ca/~shallit/Papers/ubiq.ps diakses pada tanggal 24 Maret 2010

[9] http://www.pvv.org/~asgaut/crypto/thesis/node15.html diakses pada tanggal 23 Maret 2010

[10] http://www.rsa.com/rsalabs/node.asp?id=2171 diakses pada tanggal 23 Maret 2010