BAB III

PENYANDIAN *ONE TIME PAD* MENGGUNAKAN SANDI *VIGENERE*

3.1 SANDI *VIGENERE*

Sandi *Vigenere* termasuk dalam kriptografi klasik dengan metode sandi poli-alfabetik sederhana, mengenkripsi sebuah *plaintext* yang terdiri dari beberapa karakter huruf menggunakan kunci yang terdiri barisan karakter yang diambil secara acak dan berbeda.

Definisi 3.1.1: Sandi *Vigenere* (Stinson, 2006: 13)

Diberikan bilangan bulat positif m, didefinisikan

$$\mathcal{P} = \mathcal{C} = \mathcal{K} = (\mathbb{Z}_{26})^m$$

untuk sebuah kunci $\mathcal{K} = (k_1, k_2, \dots, k_m)$, kita definisikan

$$e_{\mathcal{K}}(X) = (x_1 + k_1, x_2 + k_2, \dots, x_m + k_m)$$

dan

$$d_{\mathcal{K}}(X) = (y_1 - k_1, y_2 - k_2, \dots, y_m - k_m)$$

semua operasi dilakukan dalam $\mathbb{Z}_{26}$.

Secara matematis jika bekerja dalam $\mathbb{Z}_n$, maka proses enkripsi dapat ditulis sebagai berikut,

$$c_i = (p_i + k_j) \; mod \; n$$

dimana, $c_i =$ karakter ke-i *ciphertext*

$p_i =$ karakter ke-i *plaintext*

$k_j =$ karakter ke-j kunci

$n =$ bilangan bulat modulo n

$i, j =$ indeks (bilangan asli)

untuk mendapatkan c_i, dilakukan operasi penjumlahan modulo n antara p_i dan k_j. Setelah seluruh karakter *plaintext* sudah diubah menjadi karakter *ciphertext*, hal selanjutnya adalah pengirim mengirimkan kunci dan *ciphertext* kepada penerima. Kemudian penerima melakukan proses dekripsi untuk dapat memecahkan sandi dalam bentuk *ciphertext* tersebut. Proses dekripsi adalah proses mengubah *ciphertext* menjadi *plaintext*.

Diketahui bahwa proses enkripsi dapat ditulis,

$$\begin{aligned} c_i - (p_i + k_j) &= qn \quad ; q \in \mathbb{Z} \\ c_i - p_i - k_j &= qn \\ c_i - k_j &= p_i + qn \\ p_i &= c_i - k_j - qn \\ p_i - (c_i - k_j) &= -qn \\ p_i - (c_i - k_j) &= rn \quad ; r \in \mathbb{Z} \\ p_i &= (c_i - k_j) \bmod n \end{aligned}$$

dan proses dekripsi sandi *Vigenere* ditulis dengan:

$$p_i = (c_i - k_j) \bmod n$$

Contoh 3.1.2 Misalkan bekerja pada $\mathbb{Z}_{10}$, artinya terdapat 10 karakter yaitu $(a, b, c, d, e, f, g, h, i, j)$ jika diubah ke $\mathbb{Z}_{10}$ menjadi (0,1,2,3,4,5,6,7,8,9).

Dengan *plaintext* "$bacagedhi$" diubah ke $\mathbb{Z}_{10}$ menjadi (1,0,2,0,6,4,3,7,8), dengan pasangan kunci acak "fdi", (5,3,8). Dengan proses enkripsi:

$$c_i = (p_i + k_j) \bmod n$$

$$c_1 = (1 + 5) \bmod 10 = 6$$

$$c_2 = (0 + 3) \bmod 10 = 3$$

$$c_3 = (2 + 8) \bmod 10 = 0$$

$$c_4 = (0 + 5) \bmod 10 = 5$$

$$c_5 = (6 + 3)\ mod\ 10 = 9$$

$$c_6 = (4 + 8)\ mod\ 10 = 2$$

$$c_7 = (3 + 5)\ mod\ 10 = 8$$

$$c_8 = (7 + 3)\ mod\ 10 = 0$$

$$c_9 = (8 + 8)\ mod\ 10 = 6$$

diperoleh *ciphertext* (6,3,0,5,9,2,8,0,6) jika diubah ke karakter huruf menjadi, "$gdafjciag$". Sedangkan untuk proses dekripsi:

$$p_i = \left(c_i - k_j\right) mod\ n$$

$$p_1 = (6 - 5)\ mod\ 10 = 1$$

$$p_2 = (3 - 3)\ mod\ 10 = 0$$

$$p_3 = (0 - 8)\ mod\ 10 = 2$$

$$p_4 = (5 - 5)\ mod\ 10 = 0$$

$$p_5 = (9 - 3)\ mod\ 10 = 6$$

$$p_6 = (2 - 8)\ mod\ 10 = 4$$

$$p_7 = (8 - 5)\ mod\ 10 = 3$$

$$p_8 = (0 - 3)\ mod\ 10 = 7$$

$$p_9 = (6 - 8)\ mod\ 10 = 8$$

diperoleh *plaintext* (1,0,2,0,6,4,3,7,8) jika diubah ke karakter huruf akan kembali menjadi *plaintext* "$bacagedhi$".

3.2 SANDI *ONE TIME PAD*

Sandi *one time pad* (OTP) merupakan sandi yang mencapai kerahasiaan sempurna, artinya kriptanalis tidak dapat dengan mudah untuk memecah sandi

tidak dengan cara statistik kemunculan huruf sekalipun. Hal tersebut, karena kunci yang digunakan untuk sandi ini diberikan secara acak, hanya digunakan sekali dan panjangnya mengikuti panjang *plaintext*.

Definisi 3.2.1: Sandi *one time pad* (Menezes *et al.*, 1996: 21)

$$c_i = p_i \oplus k_i \quad untuk\ i = 1, 2, 3, \ldots$$

dimana, $c_i =$ karakter ke-i *ciphertext*

$p_i =$ karakter ke-i *plaintext*

$k_i =$ karakter ke-i kunci

$i =$ indeks (bilangan asli)

$\oplus =$ operator *bitwise* XOR

Jika barisan kunci diambil secara acak dan hanya digunakan sekali. Sedangkan untuk proses dekripsi sandi OTP adalah

$$p_i = c_i \oplus k_i$$

Contoh 3.2.2 Misalkan *plaintext* "$Mat\ UPI$" diubah ke dalam kode ASCII dapat dilihat pada tabel 2.1 menjadi $(77, 97, 116, 32, 85, 80, 73)$. Dengan panjang barisan kunci sesuai dengan panjang *plaintext* dan diambil secara acak misalkan $(4, 23, 49, 90, 16, 83, 22)$. Dengan proses enkripsi:

$$c_i = p_i \oplus k_i$$

$$\begin{aligned} c_1 &= 77 \oplus 4 \\ p_1 &= 77 \quad 0100\ 1101 \\ k_1 &= 4 \quad\ 0000\ 0100 \\ \hline & \quad\ 0100\ 1001 = 73 \end{aligned} \oplus$$

sehingga diperoleh $c_1 = 73$,

$$\begin{aligned} c_2 &= 97 \oplus 23 \\ p_2 &= 97 \quad 0110\ 0001 \\ k_2 &= 23 \quad 0001\ 0111 \\ \hline & \quad\ 0111\ 0110 = 118 \end{aligned} \oplus$$

sehingga diperoleh $c_2 = 118$,

$$c_3 = 116 \oplus 49$$
$$p_3 = 116 \quad 0111\ 0100$$
$$k_3 = 49 \quad 0011\ 0001 \ \oplus$$
$$0100\ 0101 = 69$$

sehingga diperoleh $c_3 = 69$,

$$c_4 = 32 \oplus 90$$
$$p_4 = 32 \quad 0010\ 0000$$
$$k_4 = 90 \quad 0101\ 1010 \ \oplus$$
$$0111\ 1010 = 122$$

sehingga diperoleh $c_4 = 122$,

$$c_5 = 85 \oplus 16$$
$$p_5 = 85 \quad 0101\ 0101$$
$$k_5 = 16 \quad 0001\ 0000 \ \oplus$$
$$0100\ 0101 = 69$$

sehingga diperoleh $c_5 = 69$,

$$c_6 = 80 \oplus 83$$
$$p_6 = 80 \quad 0101\ 0000$$
$$k_6 = 83 \quad 0101\ 0011 \ \oplus$$
$$0000\ 0011 = 3$$

sehingga diperoleh $c_6 = 3$,

$$c_7 = 73 \oplus 22$$
$$p_7 = 73 \quad 0100\ 1001$$
$$k_7 = 22 \quad 0001\ 0110 \ \oplus$$
$$0101\ 1111 = 95$$

sehingga diperoleh $c_7 = 95$. Setelah semua c_i diperoleh, maka didapat *ciphertext* adalah $(73, 118, 69, 122, 69, 3, 95)$ diubah kembali karakter melalui kode ASCII menjadi "$IvEzE$ETX_".

Untuk proses dekripsi adalah sebagai berikut,

$$p_i = c_i \oplus k_i$$

$$p_1 = 73 \oplus 4$$
$$c_1 = 73 \quad 0100\ 1001$$
$$k_1 = 4 \quad 0000\ 0100 \quad \oplus$$
$$0100\ 1101 = 77$$

sehingga diperoleh $p_1 = 77$,

$$p_2 = 118 \oplus 23$$
$$c_2 = 118 \quad 0111\ 0110$$
$$k_2 = 23 \quad 0001\ 0111 \quad \oplus$$
$$0110\ 0001 = 97$$

sehingga diperoleh $p_2 = 97$,

$$p_3 = 69 \oplus 49$$
$$c_3 = 69 \quad 0100\ 0101$$
$$k_3 = 49 \quad 0011\ 0001 \quad \oplus$$
$$0111\ 0100 = 116$$

sehingga diperoleh $p_3 = 116$,

$$p_4 = 122 \oplus 90$$
$$c_4 = 122 \quad 0111\ 1010$$
$$k_4 = 90 \quad 0101\ 1010 \quad \oplus$$
$$0010\ 0000 = 32$$

sehingga diperoleh $p_4 = 32$,

$$p_5 = 69 \oplus 16$$
$$c_5 = 69 \quad 0100\ 0101$$
$$k_5 = 16 \quad 0001\ 0000 \quad \oplus$$
$$0101\ 0101 = 85$$

sehingga diperoleh $p_5 = 85$,

$$p_6 = 3 \oplus 83$$
$$c_6 = 3 \quad 0000\ 0011$$
$$k_6 = 83 \quad 0101\ 0011 \quad \oplus$$
$$0101\ 0000 = 80$$

sehingga diperoleh $p_6 = 80$,

$$p_7 = 95 \oplus 22$$
$$c_7 = 95 \quad 0101\ 1111$$
$$k_7 = 22 \quad 0001\ 0110$$
$$0100\ 1001 = 73 \quad \oplus$$

sehingga diperoleh $p_7 = 73$. Setelah semua p_i diperoleh dari proses dekripsi, yaitu $(77, 97, 116, 32, 85, 80, 73)$ kemudian diubah menjadi karakter pada tabel ASCII "$Mat\ UPI$" sama seperti *plaintext* awal.

3.3 PENYANDIAN SANDI *ONE TIME PAD* MENGGUNAKAN SANDI *VIGENERE*

Diberikan sebuah *plaintext* dengan panjang m. Proses enkripsi sandi OTP menggunakan sandi *Vigenere*, selanjutnya disebut sandi modifikasi, hampir sama dengan proses enkripsi sandi OTP, yaitu dengan melakukan operasi XOR antara p_i dan $p_i{}^{-1}$. Untuk memudahkan, notasi yang digunakan adalah:

$m =$ panjang *plaintext*

$\bar{P}' =$ *plaintext* awal berupa teks $(a_1, a_2, \dots, a_m)$

$$\bar{P} = \begin{cases} \bar{P}', \text{jika } m \text{ genap} \\ \bar{P}' + ' \ ', \text{jika } m \text{ ganjil; untuk suatu } ' \ ' \text{ karakter spasi} \end{cases}$$

$(\bar{P})^{-1} =$ invers/ balikan dari $\bar{P}$

$= (a_m, a_{m-1}, \dots, a_1)$

$\overline{P^o} = (a_1, a_2, \dots, a_{\frac{m}{2}})$

$\overline{P^1} = (a_{\frac{m}{2}}, a_{\frac{m}{2}+1}, \dots, a_m)$

$\bar{K} =$ kunci

$\bar{C} =$ *ciphertext* berupa teks $(c_1, c_2, \dots, c_m)$

$\overline{C^o} = (c_1, c_2, \dots, c_{\frac{m}{2}})$

$\overline{C^1} = (c_{\frac{m}{2}}, c_{\frac{m}{2}+1}, \dots, c_m)$

$p =$ *plaintext* dalam bentuk kode ASCII

$(p)^{-1} =$ invers/ balikan dari p

$p_i =$ karakter ke-i dari p

$(p^{-1})_i =$ karakter ke-i dari p^{-1}

$k =$ kunci dalam bentuk kode ASCII

$k_i =$ karakter ke-i dari k

$c =$ *ciphertext* dalam bentuk kode ASCII

$c_i =$ karakter ke-i dari c

$i =$ indeks (bilangan asli)

Berdasarkan Amroodi *et al.* (2013) secara matematis proses enkripsi sandi modifikasi ini ditulis dengan:

$$\bar{C} = \bar{P} \oplus (\bar{P})^{-1}$$

dengan notasi $\bar{C}, \bar{P}$, dan $(\bar{P})^{-1}$ seperti yang dimaksud di atas.

Contoh 3.3.1 Misalkan *plaintext* "$kesan$" karena panjang *plaintext* adalah ganjil, maka diberikan penambahan karakter spasi " " agar panjang *plaintext* menjadi genap. Selanjutnya *plaintext* diubah ke dalam kode ASCII, menjadi $(107, 101, 115, 97, 110)$.

$$\bar{P}' = kesan \quad ; m = 5$$
$$\bar{P} = kesan \quad ; m = 6$$
$$(\bar{P})^{-1} = nasek$$
$$p = (107, 101, 115, 97, 110, 32)$$
$$(p)^{-1} = (32, 110, 97, 115, 101, 107)$$

Proses enkripsi untuk $\bar{P} = kesan$ dan $(\bar{P})^{-1} = nasek$ adalah,

$$c_i = p_i \oplus (p^{-1})_i$$

$$\begin{aligned} c_1 &= 107 \oplus 32 \\ p_1 &= 107 \quad 0110\ 1011 \\ (p^{-1})_1 &= 32 \quad 0010\ 0000 \\ &\hline \\ &0100\ 1011 = 75 \end{aligned} \oplus$$

sehingga diperoleh $c_1 = 75$,

$$\begin{aligned} c_2 &= 101 \oplus 110 \\ p_2 &= 101 \quad 0110\ 0101 \\ (p^{-1})_2 &= 110 \quad 0110\ 1110 \\ &\hline \\ &0000\ 1011 = 11 \end{aligned} \oplus$$

sehingga diperoleh $c_2 = 11$,

$$\begin{aligned} c_3 &= 115 \oplus 97 \\ p_3 &= 115 \quad 0111\ 0011 \\ (p^{-1})_3 &= 97 \quad 0110\ 0001 \\ &\hline \\ &0001\ 0010 = 18 \end{aligned} \oplus$$

sehingga diperoleh $c_3 = 18$,

$$\begin{aligned} c_4 &= 97 \oplus 115 \\ &= 115 \oplus 97 \\ &= 18 \end{aligned} \qquad \begin{aligned} c_5 &= 110 \oplus 101 \\ &= 101 \oplus 110 \\ &= 11 \end{aligned}$$

$$\begin{aligned} c_6 &= 32 \oplus 107 \\ &= 110 \oplus 32 \\ &= 75 \end{aligned}$$

setelah semua c_i diperoleh, maka

$$c = (75, 11, 18, 18, 11, 75)$$

diubah dalam karakter ASCII menjadi,

$$\bar{C} = K\texttt{VTDC1DC1VT}K$$

Untuk memudahkan pembacaan lihat tabel 3.1.

Tabel 3.1 Tabel Enkripsi Sandi Modifikasi

$\bar{P}$	k	e	s	a	n	
$(\bar{P})^{-1}$		n	a	s	e	k
$\bar{C}$	K	VT	DC1	DC1	VT	K

Sebagaimana yang dikemukakan oleh Amroodi *et al.* (2013) bahwa, misalkan

$$\bar{P} = \overline{P^o} \parallel \overline{P^1} \;\; ; \quad \parallel = \text{operator rangkaian.}$$

dengan, $m\,(\overline{P^o}) = m\,(\overline{P^1})$.

Kunci yang digunakan untuk proses dekripsi,

$$\bar{K} = \overline{P^o}$$

Untuk arti notasi $\bar{P}, \overline{P^o}, \overline{P^1}$, m dan $\bar{K}$ seperti yang di atas.

Contoh 3.3.2 Misalkan $\bar{P} = kesan$, maka

$$\bar{P} = kesan$$
$$\bar{P} = kes \parallel an$$

Jadi, $\overline{P^o} = kes$ dan $\overline{P^1} = an$. Dan kunci yang digunakan untuk proses dekripsi adalah,

$$\bar{K} = \overline{P^o}$$
$$\bar{K} = kes.$$

Sedangkan proses dekripsi sandi modifikasi ini cukup unik, yaitu dengan men-XOR-kan setengah karakter pertama dari *ciphertext* yang telah didapat dari proses enkripsi. Karena *ciphertext* yang digunakan hanya setengah dari panjang

ciphertext yang telah diperoleh, maka *plaintext* yang dapat dipecahpun hanya setengah sampai karakter terakhir dari *plaintext* yang asli.

Dalam jurnal Amroodi *et al.* (2013) dijelaskan bahwa secara matematis proses dekripsi sandi modifikasi ini dapat ditulis:

$$(\overline{P^1})^{-1} = \overline{C^o} \oplus \overline{K}$$

dengan notasi $(\overline{P^1})^{-1}$, $\overline{C^o}$ dan $\overline{K}$ seperti yang dimaksud di atas.

Contoh 3.3.3 Misalkan $\bar{C} = K\,\texttt{VTDC1DC1VT}\,K$, maka $\overline{C^o} = K\,\texttt{VTDC1}$ dan $\overline{K} = kes$.

$$c^0 = (75, 11, 18)$$
$$k = (107, 101, 115)$$

Sehingga proses dekripsinya adalah

$$(p^1)^{-1}{}_i = (c^0)_i \oplus k_i$$

$$\begin{aligned} (p^1)^{-1}{}_1 &= 75 \oplus 107 \\ (c^0)_1 &= 75 \quad 0100\ 1011 \\ k_1 &= 107 \quad 0110\ 1011 \\ &\qquad\qquad \oplus \\ &\quad 0010\ 0000 = 32 \end{aligned}$$

sehingga diperoleh $(p^1)^{-1}{}_1 = 75$,

$$\begin{aligned} (p^1)^{-1}{}_2 &= 11 \oplus 101 \\ (c^0)_2 &= 11 \quad 0000\ 1011 \\ k_2 &= 101 \quad 0110\ 0101 \\ &\qquad\qquad \oplus \\ &\quad 0110\ 1110 = 110 \end{aligned}$$

sehingga diperoleh $(p^1)^{-1}{}_2 = 110$,

$$\begin{aligned} (p^1)^{-1}{}_3 &= 18 \oplus 115 \\ (c^0)_3 &= 18 \quad 0001\ 0010 \\ k_3 &= 115 \quad 0111\ 0011 \\ &\qquad\qquad \oplus \\ &\quad 0110\ 0001 = 97 \end{aligned}$$

sehingga diperoleh $(p^1)^{-1}{}_3 = 97$. Setelah semua $(p^1)^{-1}{}_i$ diperoleh, maka

$$(p^1)^{-1} = (32, 110, 97)$$

diubah dalam karakter menjadi,

$$(\overline{P^1})^{-1} = na$$

Untuk memudahkan pembacaan lihat tabel 3.2.

Tabel 3.2 Tabel Dekripsi Sandi Modifikasi

$\overline{C^o}$	K	VT	DC1
$\overline{K}$	k	e	s
$(\overline{P^1})^{-1}$		n	a

Setelah memperoleh setengah sampai karakter terakhir dari *plaintext* yang asli dari proses dekripsi, hal selanjutnya yang dilakukan adalah penggabungan kunci dengan invers dari hasil dari proses dekripsi tadi. Proses penggabungan dengan berdasarkan teorema berikut ini,

Teorema 3.3.4 (Amroodi *et al.*, 2013)

Diketahui, $\bar{P}, \overline{P^o}, \overline{P^1}, \overline{C^o}$ *dan* $\overline{K}$ *seperti yang di atas. Maka,*

$$\bar{P} = \overline{K} \parallel (\overline{C^o} \oplus \overline{K})^{-1}$$

Bukti:

Karena, $(\overline{P^1})^{-1} = \overline{C^o} \oplus \overline{K}$ sedemikian sehingga,

$$\overline{K} \parallel (\overline{C^o} \oplus \overline{K})^{-1} = \overline{K} \parallel ((\overline{P^1})^{-1})^{-1}$$

$$= \overline{K} \parallel \overline{P^1}$$

$$= \overline{P^o} \parallel \overline{P^1}$$

$$= \bar{P}$$

■

Menurut teorema 3.3.4, jika gabungan dari kunci dan operasi XOR dari $\overline{C^o}$, yaitu setengah karakter pertama dari *ciphertext* dan kunci yang diinverskan akan menghasilkan *plaintext*.

Contoh 3.3.5 Misalkan $\overline{C^o} = K\texttt{VTDC1}$ dan $\overline{K} = kes$.

$$\overline{P} = \overline{K} \parallel (\overline{C^o} \oplus \overline{K})^{-1}$$

$$= \overline{K} \parallel ((\overline{P^1})^{-1})^{-1}$$

$$= kes \parallel (na)^{-1}$$

$$= kes \parallel an$$

$$= kesan$$

Terlihat pada contoh 3.3.5, bahwa benar jika gabungan dari kunci (kes) dan operasi XOR dari $\overline{C^o}$ dan kunci yang diinverskan, $\overline{K}$, ($(na)^{-1} = an$) akan menghasilkan *plaintext* secara utuh ($kesan$).