BAB III

KOMBINASI *VIGÈNERE CIPHER* DAN *KEYED COLUMNAR TRANSPOSITION*

III.1 *VIGÈNERE CIPHER*

Cipher ini adalah termasuk cipher simetris, yaitu cipher klasik abjad majemuk. Karena setiap huruf dienkripsikan dengan fungsi yang berbeda. *Vigenère Cipher* merupakan bentuk pengembangan dari *Caesar Cipher*. Kelebihan sandi ini dibanding *Caesar Cipher* dan cipher monoalfabetik lainnya adalah cipher ini tidak begitu rentan terhadap metode pemecahan cipher yang disebut analisis frekuensi. Giovan Batista Belaso menjelaskan metode ini dalam buku La cifra del. Sig. Giovan Batista Belaso (1553); dan disempurnakan oleh diplomat Perancis Blaise de Vigenère, pada 1586. Pada abat ke-19, banyak orang yang mengira Vigenère adalah penemu cipher ini, sehingga, cipher ini dikenal luas sebagai *Vigenère Cipher*.

Cipher ini dikenal luas karena cara kerjanya mudah dimengerti dan dijalankan, dan bagi para pemula sulit dipecahkan. Pada saat kejayaannya, cipher ini dijuluki *le chiffre indéchiffrable* (bahasa Prancis: 'cipher yang tak terpecahkan'). Metode pemecahan cipher ini baru ditemukan pada abad ke-19. Pada tahun 1854, Charles Babbage menemukan cara untuk memecahkan *Vigenère Cipher*. Metode ini dinamakan Metode *Kasiski* karena Friedrich Kasiski-lah yang pertama mempublikasikannya.

Tabel 3.1 Tabel Bujursangkar *Vigènere*

Kunci \ 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
a	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
b	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	A
c	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
d	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
e	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
f	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
g	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
h	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
i	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
j	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
k	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
l	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
m	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
n	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
o	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
p	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
r	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
s	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
t	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
u	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
v	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
w	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
x	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
z	Z	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

Vigènere Cipher menggunakan Bujursangkar *Vigènere* untuk melakukan enkripsi dan dekripsi. Jika pada *Caesar Cipher* setiap huruf digeser dengan besar geseran yang sama, maka pada *Vigènere Cipher* setiap huruf digeser dengan besar yang berbeda sesuai dengan kuncinya.

III.1.1 Enkripsi *Vigènere Cipher*

Secara matematis, enkripsi *Vigènere Cipher* dengan jumlah karakter sebanyak 26 dapat ditulis dalam bentuk

$c_i \equiv (p_i + k_j) \bmod 26$ atau

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

untuk *Vigènere Cipher* dengan jumlah karakter n.

Ket : i = 1, 2, 3, …, (panjang kunci)

$j = ((i - 1) \bmod 5) + 1$

Contoh 3.1 (Enkripsi *Vigènere Cipher*)

Terdapat 10 karakter ($n = 10$) yang digunakan, yaitu A, B, C, D, E, F, G, H, I, dan _ yang bersesuaian dengan bilangan bulat 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 (modulo 10) seperti tabel (3.2).

Tabel 3.2 Tabel Dengan 10 Karakter

A	B	C	D	E	F	G	H	I	_
0	1	2	3	4	5	6	7	8	9

Misalkan plainteks yang akan dienkripsikan adalah ADA_ECI.

Plainteks : ADA_ECI yang bersesuaian dengan 0 3 0 9 4 2 8

Dengan kunci DIA yang bersesuaian dengan 3 8 0

Tabel 3.3 Tabel Enkripsi ADA_ECI Dengan Kunci Dia

A	D	A	_	E	C	I
0	3	0	9	4	2	8
D	I	A	D	I	A	D
3	8	0	3	8	0	3

Maka berdasarkan tabel (3.3) :

E(A) = (0+3) mod 10 = 3 = D

E(E) = (4+8) mod 10 = 2 = C

E(D) = (3+8) mod 10 = 1 = B

E(C) = (2+0) mod 10 = 2 = C

E(A) = (0+0) mod 10 = 0 = A

E(I) = (8+3) mod 10 = 1 = B

E(_) = (9+3) mod 10 = 2 = C

Cipherteks : DBACCCB

Kode enkripsi *Vigènere Cipher* (26 karakter) dalam *pseudecode*:

```
procedure EnkripsiVigenereCipher
DEKLARASI
   type ord    : array[A..Z] of char □ [0..25]
   type chr    : array[0..25] of char □ [A..Z]
   PlainTeks, CipherTeks, Kunci  : string
   PanjangKunci, pi  : integer
ALGORITMA
   read  (PlainTeks)
   read (Kunci)
   panjangKunci □ length(Kunci)

   for pi □ 1 to length(PlainTeks) do
      CipherTeks[pi] □ chr((ord(PlainTeks[pi])+
         ord(Kunci[((pi-1) mod PanjangKunci)+1])) mod 26)
   endfor
   write (CipherTeks)
```

III.1.2 Dekripsi *Vigènere Cipher*

Untuk melakukan dekripsi pada *Vigènere Cipher*, digunakan kebalikan dari fungsi enkripsinya.

Secara matematis, dekripsi *Vigènere Cipher* dengan jumlah karakter sebanyak 26 dapat ditulis dalam bentuk

$p_i \equiv (c_i - k_j) \bmod 26$ atau

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

untuk *Vigènere Cipher* dengan jumlah karakter n.

Ket : i = 1, 2, 3, …, panjang kunci

$j = ((i - 1) \bmod 5) + 1$

Contoh 3.2 (Dekripsi *Vigènere Cipher*)

Terdapat 10 karakter ($n = 10$) yang digunakan, yaitu A, B, C, D, E, F, G, H, I, dan _ yang bersesuaian dengan bilangan bulat 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 (modulo 10) seperti tabel (3.2).

Misalkan plainteks yang akan dienkripsikan adalah DBACCCB.

Plainteks : DBACCCB yang bersesuaian dengan 3 1 0 2 2 2 1

Dengan kunci DIA yang bersesuaian dengan 3 8 0

Tabel 3.4 Tabel Dekripsi DBACCCB Dengan Kunci DIA

D 3	B 1	A 0	C 2	C 2	C 2	B 1
D 3	I 8	A 0	D 3	I 8	A 0	D 3

Berdasarkan tabel (3.4) :

D(D) = (3-3) mod 10 = 0 = A *D*(C) = (2-8) mod 10 = 4 = E

D(B) = (1-8) mod 10 = 3 = D *D*(C) = (2-0) mod 10 = 2 = C

D(A) = (0-0) mod 10 = 3 = A *D*(B) = (1-3) mod 10 = 8 = I

D(C) = (2-3) mod 10 = 9 = _

Sehingga cipherteks DBACCCB kembali menjadi plainteks ADA_ECI.

Kode dekripsi *Vigènere Cipher* (26 karakter) dalam *pseudecode*:

```
procedure DekripsiVigenereCipher
DEKLARASI
   type ord    : array[A..Z] of char  [0..25]
   type chr    : array[0..25] of char  [A..Z]
   PlainTeks, CipherTeks, Kunci  : string
   PanjangKunci, pi, ord   : integer
ALGORITMA
   read  (CipherTeks)
   read (Kunci)
   panjangKunci  length(Kunci)

   for pi  1 to length(ChiperTeks) do
      ChiperTeks[pi]  chr((ord(ChiperTeks[pi])-
         ord(Kunci[((pi-1) mod PanjangKunci)+1])) mod 26)
   endfor
   write(ChiperTeks)
```

III.1.3 METODE KASISKI

Metode pemecahan *Vigènere Cipher* baru ditemukan pada abad ke-19 yaitu pada tahun 1854 oleh Charles Babbage. Namun orang pertama yang mempublikasikannya adalah Friedrich Kasiski sehingga metode ini dinamakan

metode Kasiski (*kasiski Method*). Metode ini adalah merupakan jenis *analytical attack* untuk mencari panjang kunci sehingga akan mengurangi kemungkinan kunci yang ada.

Langkah-langkah metode Kasiski:

1. Temukan semua kriptogram yang berulang di dalam cipherteks (pesan yang panjang biasanya mengandung kriptogram yang berulang).
2. Hitung jarak antara setiap kriptogram yang berulang.
3. Hitung semua faktor (pembagi) dari jarak-jarak tersebut.
4. Tentukan irisan dari himpunan faktor-faktor pembagi tersebut. Nilai-nilai dalam irisan tersebut kemungkinan besar adalah panjang kunci.

Contoh 3.3 (Metode Kasiski)

Misalkan cipherteks (dengan 26 huruf kapital) yang didapat adalah :

DQNKKCVPBNVIVMUFIGTNILRGGCVFXNFQEBNAI

Dengan menggunakan metode Kasiski, dari cipherteks tersebut ditemukan kriptogram yang berulang, yaitu CV dan BN. Jarak antara dua buah perulangan CV adalah 20. Semua faktor pembagi dari 20 adalah {20, 10, 5, 4, 2, 1}. Jarak antara dua buah perulangan BN adalah 25. Semua faktor pembagi dari 25 adalah {25, 5, 1}. Irisan dari kedua buah himpunan (faktor-faktor pembagi 20 dan 25) tersebut adalah 5 dan 1. Maka panjang kunci kemungkinan besar adalah 5 atau 1.

Setelah panjang kunci ditemukan, selanjutnya gunakan *exhaustive attack* untuk mencari kunci.

III.2 *KEYED COLUMNAR TRANSPOSITION* (*KCTR*)

KCTR dibuat pada tahun 1950-an, sebagai pengganti dan penyempurnaan dari algoritma *SCTR* yang pada waktu tersebut telah dapat dengan mudah dipecahkan oleh kriptanalis karena metodenya yang terlalu sederhana. Dengan cara mencoba satu demi satu kunci simetris yang mungkin (kunci simetris terbatas, hanya bilangan diantara satu sampai panjang chiperteks), kriptanalis hanya memerlukan selembar kertas dan pensil untuk menulis semua hasil percobaannya itu. Ditambah lagi bila kriptanalisnya lebih dari satu orang, percobaan memecahkan chiperteks dapat dibagi-bagi berdasarkan bilangan kunci simetris. Oleh karena itu, dibuatlah algoritma *SCTR* baru yang kunci simetrinya menggunakan permutasi dari *k* bilangan asli pertama (*keyword*). Fungsi kunci ini jelas untuk lebih memperkuat proses enkripsi dan mempersulit proses kriptanalisis, yaitu dengan cara mengubah urutan transposisi kolom berdasarkan urutan angka kunci.

III.2.1 Enkripsi *Keyed Columnar Transposition*

Metode enkripsi *KCTR* adalah hampir sama dengan *SCTR*, yaitu membagi plainteks menjadi blok-blok dengan panjang tertentu yang kemudian blok-blok tersebut disusun dalam bentuk baris dan kolom. Namun hasil enkripsinya adalah dengan membaca secara vertikal (tiap kolom) sesuai urutan kunci.

Langkah-langkah enkripsi *Keyed Columnar Transposition* dengan metode *irregular case* adalah:

1. Hitunglah panjang plainteks (n).
2. Tentukan panjang kunci k dengan $0 < k \leq n$.
3. Tentukan kunci dengan bentuk permutasi k bilangan asli pertama.
4. Bagilah plainteks menjadi blok-blok dengan panjang k (panjang blok terakhir = $((n - 1) \bmod k)+1$).
5. Susunlah tiap blok menjadi baris-baris sehingga membentuk baris dan kolom dengan k kolom dan b baris dimana $b = (n + k - 1)$ div k.
6. Buatlah blok baru dengan kolom pertama sebagai blok 1, kolom kedua sebagai blok 2, sampai blok k.
7. Hasil cipherteks adalah dengan membaca blok-blok baru tersebut sesuai urutan kunci.

Contoh 3.4 (Enkripsi *KCTR*)

Misalkan plainteks yang akan dienkripsi adalah ADA_ECI dengan kunci 2 3 1 ($k = 3$).

Maka plainteks akan dibagi menjadi blok-blok (3 kolom) dan disusun

1	2	3
A	D	A
_	E	C
I		

Kemudian hasil enkripsinya adalah dengan membaca kolom-kolom secara vertikal sesuai urutan kunci (231) dari 1=A_I, 2=DE dan 3=AC sehingga plainteks ADA_ECI akan dienkripsi menjadi cipherteks : DEACA_I

Kode enkripsi *Keyed Columnar Transposition* dalam *pseudecode*:

```
procedure EnkripsiColumnar
DEKLARASI
   PlainTeks, CipherTeks, Kunci     : string
   PanjangKunci, KolomKe, Baris, u  : integer
   Kolom: array[1..9] of string
ALGORITMA
   read(PlainTeks)
   read(Kunci)
   PanjangKunci  length(Kunci)
   for KolomKe  1 to PanjangKunci do
      for Baris  0 to ((Length(PlainTeks)-KolomKe) div
         PanjangKunci) do
         Kolom[KolomKe]  Kolom[KolomKe]+
            PlainTeks[kolomKe+(PanjangKunci*Baris)]
      endfor
   endfor
  for u  1 to PanjangKunci do
    CipherTeks  CipherTeks+ Kolom[Kunci[u]]
write(CipherTeks)
```

III.2.2 Dekripsi *Keyed Columnar Transposition*

Langkah-langkah dekripsi *Keyed Columnar Transposition* dengan metode *irregular case* adalah:

1. Hitunglah panjang plainteks (*n*).

2. Misalkan kunci adalah $a_1, a_2, a_3 \ldots, a_b$ dengan panjang k . ($a_1 \neq a_2 \neq a_3 \neq \cdots \neq a_b$, $a_i \leq k \in \mathbb{N}$), maka kolom ke-$a_1$ adalah u_1 karakter pertama cipherteks, dimana $u_1 = ((n - a_1) \text{ div } k) + 1$).
3. Kolom ke-a_2 adalah karakter ke-$(u_1 + 1)$ sampai karakter ke-$(u_1 + u_2)$ cipherteks, dimana $u_2 = ((n - a_2) \text{ div } k) + 1$).
4. Kolom ke-a_3 adalah karakter ke-$(u_1 + u_2 + 1)$ sampai karakter ke-$(u_1 + u_2 + u_3)$ cipherteks, dimana $u_3 = ((n - a_3) \text{ div } k) + 1$).
5. Begitu seterusnya sampai kolom ke-a_b adalah u_b karakter terakhir dari cipherteks, dimana $u_b = ((n - a_b) \text{ div } k) + 1$).
6. Urutkan kolom-kolom tersebut dari kolom-1 ke kolom-*k* (buat kembali dalam bentuk baris dan kolom).
7. Hasil enkripsinya adalah dengan membaca dari baris pertama sampai baris terakhir.

Contoh 3.5 (Dekripsi *KCTR*)

Misalkan cipherteks yang akan didekripsi adalah DEACA_I (*n* = 7) dengan kunci 2 3 1 (*k* = 3).

1. Kolom ke-2 adalah 2 $(u_1 = ((7 - 2) \text{ div } 3) + 1)$ karakter pertama cipherteks yaitu DE.
2. Kolom ke-3 adalah karakter ke-3 $(u_1 + 1 = 2 + 1)$ sampai karakter ke-4 $(u_1 + u_2 = 2 + 2)$ cipherteks yaitu AC, dimana $u_2 = (((7 - 3) \text{ div } 3) + 1) = 2$.

3. Kolom ke-1 adalah karakter ke-5 $(u_1 + u_2 + 1 = 2 + 2 + 1)$ sampai karakter ke-7 $(u_1 + u_2 + u_3 = 2 + 2 + 3)$ cipherteks, atau u_3 karakter terakhir dari cipherteks yaitu A_I, dimana $u_3 = ((7 - 1) \text{ div } 3) + 1)$.
4. Urutkan kolom-kolom tersebut dari kolom ke-1 sampai kolom ke-3 (buat dalam bentuk baris dan kolom).

1	2	3
A	D	A
_	E	C
I		

5. Hasil enkripsinya adalah dengan membaca dari baris pertama sampai baris ke 3, yaitu ADA_ECI.

Kode dekripsi *Keyed Columnar Transposition* dalam *pseudecode*:

```
procedure DekripsiColumnar
DEKLARASI
   CipherTeks, PlainTeks, Kunci     : string
   PanjangKunci, BarisKe, Baris, ci, u, cki, i: integer
   Kolom: array[1..9] of string
ALGORITMA
   read(CipherTeks)
   read(Kunci)
   PanjangKunci  length(Kunci)
   for u  1 to PanjangKunci do
      for cki  ci to ci+ ((length(CipherTeks)-Kunci[u])) div
         PanjangKunci) do
            Kolom[Kunci[u])]  Kolom[Kunci[u])]+
               CipherTeks[cki]
            ci  ci+ (((length(CipherTeks)-Kunci[u]) div
               PanjangKunci)+ 1)
      endfor
   endfor
   PlainTeks  ''
   for Baris  1 to (((length(CipherTeks)-KolomKe) div
PanjangKunci)+ 2) do
   for KolomKe  1 to PanjangKunci do
      PlainTeks  PlainTeks+ Kolom[KolomKe][baris]
   endfor
  for i  1 to length(CipherTeks) do
    PlainTeks  PlainTeks+ PlainTeks[i]
   write(PlainTeks)
```

III.3 KOMBINASI *VIGÈNERE CIPHER* DAN *KEYED COLUMNAR TRANSPOSITION*

Mengkombinasikan dua cipher tentu saja akan meningkatkan keamanan karena semakin sulitnya suatu kriptografi dapat dipecahkan. Sehingga dengan mengkombinasikan dua cipher yaitu *Vigènere Cipher* dan *Keyed Columnar Transposition* akan dibutuhkan dua kali pengkriptanalisisan. Sehingga dengan mengkombinasikan dua cipher tersebut akan ada dua cara, yaitu (1) diawali dengan *Vigènere Cipher* kemudian *Keyed Columnar Transposition* dan (2) diawali dengan *Keyed Columnar Transposition* kemudian *Vigènere Cipher*.

Secara matematis pengkombinasian dua cipher dapat ditulis

Enkripsi : $(E_1 \circ E_2)(P) = E_2(E_1(P)) = E_2(C_1) = C_2$

III.3.1 *Vigènere Cipher – Keyed Columnar Transposition*

Mengkombinasikan dua cipher yaitu dengan melakukan *Vigènere Cipher* diikuti dengan *Keyed Columnar Transposition* secara matematis dapat ditulis

Enkripsi : $(E_C \circ E_V)(P) = E_C(E_V(P)) = E_C(C_1) = C_2$

Dekripsi: $(D_V \circ D_C)(C_2) = D_V(D_C(P)) = D_V(C_1) = P$

Ket : E_V, D_V = enkripsi, dekripsi *Vigènere Cipher*

E_C, D_C = enkripsi, dekripsi *Keyed Columnar Transposition*

Contoh 3.6 (*Vigènere – KCTR)*

Plainteks : ADA_ECI yang bersesuaian dengan 0 3 0 9 4 2 8 (tabel 3.2). Dengan kunci *Vigènere Cipher* DIA dan kunci *Columnar Transposition Cipher* = 2 3 1.

Maka cipherteks yang dihasilkan dengan *Vigènere Cipher* adalah DBACCCB (contoh 3.1). Kemudian plainteks baru, yaitu DBACCCB dienkripsi dengan *Keyed Columnar Transposition* dengan kunci 231 (*k*=3).

1 2 3

D B A
C C C
B

Kemudian hasil enkripsinya adalah dengan membaca kolom-kolom secara vertikal sesuai urutan (kunci=231) 1=A_I, 2=DE, 3=AC.

Cipherteks : BCACDCB

III.3.2 *Keyed Columnar Transposition - Vigènere Cipher*

Mengkombinasikan dua cipher yaitu dengan melakukan *Keyed Columnar Transposition* diikuti dengan *Vigènere Cipher* secara matematis dapat ditulis

Enkripsi : $(E_V \circ E_C)(P) = E_V\left(E_C(P)\right) = E_V(C_1) = C_2$

Dekripsi: $(D_C \circ D_V)(C_2) = D_C\left(D_V(P)\right) = D_C(C_1) = P$

Contoh 3.7 (*KCTR – Vigènere)*

Plainteks : ADA_ECI

Dengan kunci *Keyed Columnar Transposition* = 2 3 1 dan kunci *Vigènere Cipher* DIA .

Dengan menggunakan *Keyed Columnar Transposition* dengan kunci 2 3 1 (k = 3) akan menghasilkan cipherteks DEACA_I (contoh 3.2). Kemudian plainteks baru, yaitu DEACA_I dienkripsi dengan *Vigènere Cipher* (k = DIA).

Plainteks : DEACA_I yang bersesuaian dengan 3 4 0 2 0 9 8 (tabel 2.3).

Dengan kunci DIA yang bersesuaian dengan 3 8 0, maka berdasar tabel (3.5)

Tabel 3.5 Tabel Enkripsi ACA_IDE Dengan Kunci DIA

A	C	A	_	I	D	E
0	2	0	9	8	3	4
D	I	A	D	I	A	D
3	8	0	3	8	0	3

E(D) = (3+3) mod 10 = 6 = G

E(A) = (0+8) mod 10 = 8 = I

E(E) = (4+8) mod 10 = 2 = C

E(_) = (9+0) mod 10 = 9 = _

E(A) = (0+0) mod 10 = 0 = A

E(I) = (8+3) mod 10 = 1 = B

E(C) = (2+3) mod 10 = 5 = F

Chiperteks : GCAFI_B