Penerapan Prinsip Operasi *Chiper Block Chaining (CBC)* Pada Algoritma Kriptografi Klasik

Joel T.H.P. Hutasoit *(13504144)*
Jurusan Teknik Informatika ITB Bandung, Email: if14144@students.if.itb.ac.id

Abstract *– Makalah ini membahas tentang implementasi prinsip dasar operasi yang digunakan pada algoritma kriptografi modern yaitu Chiper Block Chaining pada algoritma kriptografi klasik yaitu Chiper Abjad Majemuk dengan Substitusi Periodik. Hal ini dilakukan untuk lebih memperkuat algoritma klasik tersebut dimana akan dapat menutupi ataupun memberikan lapisan terhadap pola penggunaan kunci yang sama dalam melakukan substitusi periodik.*

Kata Kunci *: Kriptografi Klasik, Chiper Block Chaining, CBC, Substitusi Periodik, Algoritma*

1. PENDAHULUAN

Operasi *Chiper Block Chaining* merupakan penerapan mekanisme umpan balik pada sebuah blok bit dimana hasil enkripsi blok sebelumnya diumpanbalikkan ke dalam proses enkripsi blok *current*. Operasi ini diterapkan pada algoritma kriptografi modern yang sudah beroperasi pada level bit (*0 atau 1*) maupun sekelompok / blok bit dan bukan karakter.

Algoritma kriptografi klasik merupakan algoritma yang masih beroperasi pada level karakter (*pada umumnya memakai set karakter ASCII sebanyak 256 jenis karakter ~ 1 byte / 8 bit*) dengan menggunakan metode substitusi (pergantian karakter) maupun transposisi (pergeseran karakter).

Salah satu algoritma kriptografi klasik yang akan dieksplorasi adalah algoritma *chiper* abjad majemuk dengan substitusi secara periodik yang menggunakan kunci tertentu, dimana secara singkat dapat digambarkan sebagai berikut :

```
P : KRIPTOGRAFIKLASIK
K : LAMPIONLAMPIONLAM
C : VRUEBCTCARXSWNDIW
```

Gambar 1 : Contoh Operasi Chiper Substitusi Periodik

Dimana *P* merupakan plainteks, *K* merupakan kunci dan *C* merupakan chiperteks. Algoritma ini pada dasarnya sudah dapat menutupi kelemahan algoritma kriptografi klasik yaitu dengan menutupi frekuensi kemunculan huruf pada *chiper-text* tetapi tidak menutup kemungkin *chiper-text* bisa dipecahkan kriptanalis dengan menemukan pola substitusi yang memungkinkan dikarenakan substitusi dilakukan dengan kunci yang sama secara periodik. Untuk itu penambahan sebuah lapisan yang akan menutupi kemunculan pola substitusi secara periodik tersebut perlu dilakukan pada fungsi enkripsi maupun dekripsi yaitu dengan menerapkan prinsip *Chiper Block Chaining* dari algoritma kriptografi modern untuk kemudian dioperasikan dengan kunci yang sudah ditentukan.

2. IMPLEMENTASI PRINSIP OPERASI CBC PADA ALGORITMA CHIPER SUBSTITUSI PERIODIK

Proses enkripsi pada algoritma *chiper* substitusi periodik seperti pada contoh sebelumnya (*gambar 1*) dapat dituliskan sebagai :

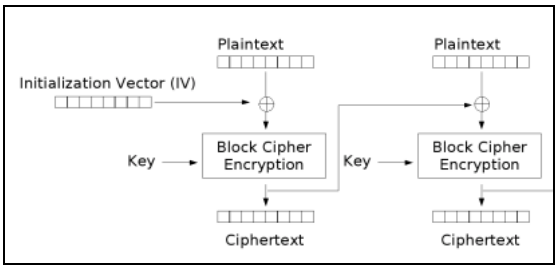
$$C_i(P) = (P + K_i) \bmod n \text{ …… persamaan (1)}$$
untuk $K = k_1k_2\ldots k_m$

Dimana $C_i(P)$ menyatakan chiperteks karakter ke-i dari plainteks P didapat dengan mengoperasikan setiap karakter P terhadap kunci K karakter ke-i. Operasi enkripsi dimulai dari m karakter pertama dengan menggunakan pola sesuai persamaan (1) dimana untuk m karakter berikutnya kembali menggunakan pola yang sama. Algoritma ini tentunya dapat menutupi frekuensi kemunculan huruf seperti pada algoritma klasik substitusi yang biasa, tetapi penggunaan pola yang sama secara periodik dengan "parameter kunci" yang sama (statis) tentunya membuat kriptanalis akan mudah memecahkan chiperteks melalui analisis matematis dengan bantuan komputer.

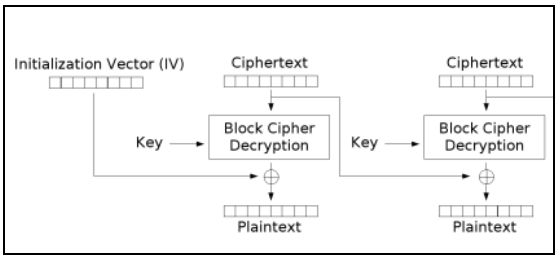
Oleh karena itu, untuk lebih memperkuat algoritma ini ada baiknya operasi substitusi dengan menggunakan pola yang sama tersebut dilakukan dengan menggunakan parameter yang berubah (dinamis) dari periode pertama ke periode berikutnya. Parameter yang dinamis tersebut dapat diperoleh dengan berbagai cara seperti dengan menggunakan tabel substitusi maupun bilangan acak. Kedua contoh tersebut memiliki kelemahan dalam hal fleksibiltas (misalkan pemakaian *space* tambahan untuk menyimpan tabel substitusi) maupun reabilitasnya (integritas teks setelah proses enkripsi dan dekripsi).

Salah satu pendekatan yang dipakai adalah dengan menerapkan prinsip-prinsip dasar operasi yang digunakan pada algoritma kriptografi modern yaitu *Chiper Block Chaining* (CBC), dimana bit-bit *plaintext* dibagi ke dalam blok-blok yang memiliki panjang tertentu untuk kemudian setiap blok

dioperasikan terlebih dahulu terhadap hasil operasi dari blok sebelumnya (dengan menggunakan operator XOR) setelah itu dimasukkan ke fungsi enkripsi dengan kunci yang sudah ditentukan. Sebaliknya operasi dekripsi dilakukan dengan memasukkan blok *chipertext* ke fungsi dekripsi untuk kemudian dioperasikan dengan blok *chipertext* sebelumnya, seperti yang digambarkan pada skema berikut :



Gambar 2 : Proses Enkripsi CBC



Gambar 3 : Proses Dekripsi CBC

Secara matematis, metode enkripsi (EK) maupun dekripsi (DK) dari CBC di atas dapat dinyatakan sebagai :

$$Ci = EK(Pi \oplus C_{i-1}) \ldots\ldots \text{ persamaan (2)}$$
$$Pi = DK(Ci) \oplus C_{i-1} \ldots\ldots \text{ persamaan (3)}$$

Dimana *Pi* maupun *Ci* merupakan bit ke-i dari suatu blok bit. Sesuai dengan skema dan persamaan diatas maka hal yang sama dapat diterapkan pada algoritma kriptografi klasik khususnya algoritma *chiper* dengan substitusi periodik. Adapun relevansi yang dilakukan pada eksplorasi kali ini dapat dilihat pada tabel berikut :

Tabel 1 : Relevansi Awal Prinsip CBC

Prinsip CBC	Klasik	Modern
Satuan Blok	Karakter (*byte*)	Bit
Modus Operasi	Karakter (*byte*)	Bit
Initial Vector	Blok Karakter *Dummy* Terinisialisasi	Blok Bit *Dummy* Secara Acak atau dari *user*
Panjang Blok	Sesuai Panjang Kunci (n karakter)	Ditentukan oleh *user* (n bit)
Operasi *Block Chaining*	Penjumlahan, Pengurangan dan Modulo	XOR

Sesuai dengan tabel relevansi dan skema operasi pada gambar (2) dan (3) di atas maka secara matematis persamaan enkripsi (EK) dan dekripsi (DK) untuk implementasi CBC pada chiper substitusi periodik dapat diturunkan sebagai berikut :

$$Ci = EK(Pi + C_{i-1}) \ldots\ldots \text{ persamaan (4)}$$
$$Pi = DK(Ci) - C_{i-1} \ldots\ldots \text{ persamaan (5)}$$

Dimana baik *Pi* maupun *Ci* merupakan karakter ke-i dari suatu blok karakter. Dengan mensubstitusikan persamaan (1) ke (4) dan ke (5), maka secara keseluruhan implementasi CBC pada chiper substitusi periodik dapat dituliskan sebagai :

$$Ci(P) = (Pi + C_{i-1}) + Ki \bmod n \ldots \text{ persamaan (5)}$$
$$Pi(C) = ((Ci - Ki) - C_{i-1}) \bmod n \ldots \text{ persamaan (6)}$$
untuk $K = k_1 k_2 \ldots k_m$

Dimana n merupakan jumlah jenis karakter yang dioperasikan (dalam eksplorasi sebanyak jumlah alphabet yaitu 26 huruf) dan m merupakan panjang kunci. Sejauh ini C_0 ataupun *initial vector* yang digunakan adalah blok karakter yang diisi karakter 'A' sepanjang kunci. Sesuai dengan relevansi maka plainteks maupun chiperteks dibagi ke dalam blok-blok sesuai dengan panjang kunci yang diberikan kemudian operasi enkripsi pada suatu blok dilakukan dengan melakukan operasi juga terhadap blok sebelumnya (*chaining*). Untuk contoh sebelumnya, operasinya adalah sebagai berikut (sesuai skema gambar 2) :

```
P  : KRIPTOG RAFIKLA SIK
BC : AAAAAAA VRUEBCT XRL
K  : LAMPION LAMPION LAM
C  : VRUEBCT XRLBTYG AZH
```
Gambar 4 : Operasi CBC pada Chiper Substitusi Periodik (Enkripsi)

Dimana *BC* merupan *BlockChain* yaitu hasil operasi dari blok sebelumnya untuk dioperasikan (dikaitkan) ke blok *current (*yang sedang diproses*)*.

Sebaliknya untuk operasi dekripsi, chiperteks terlebih dahulu dioperasikan terhadap kunci untuk kemudian dioperasikan dengan chiperteks blok sebelumnya (*chaining*) seperti yang ditunjukan oleh gambar berikut (sesuai skema gambar 3) :

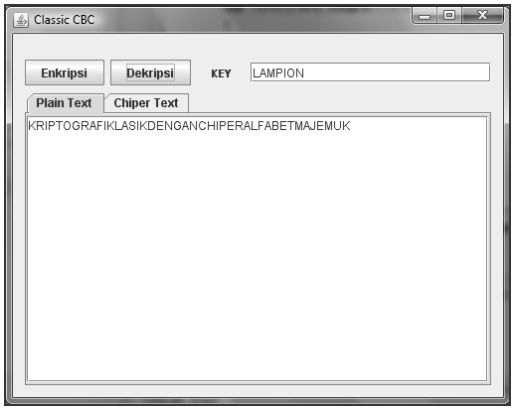
```
C  : VRUEBCT XRLBTYG AZH
K  : LAMPION LAMPION LAM
BC : AAAAAAA VRUEBCT XRL
P  : KRIPTOG RAFIKLA SIK
```
Gambar 5 : Operasi CBC pada Chiper Substitusi Periodik (Dekripsi)

Dapat dilihat bahwa *plaintext* dibagi ke dalam blok-blok yang panjangnya sesuai dengan panjang kunci yaitu 7 karakter. Setiap operasi selalu

melibatkan hasil operasi sebelumnya (umpanbalik) dimana pada operasi pertama dilakukan terhadap blok yang berisikan *dummy character*, sesuai dengan prinsip dasar operasi CBC. Dengan demikian substitusi periodik dengan pola kunci yang sama akan tertutupi dan pembentukan *chipertext* tidak bergantung hanya pada blok *plaintext* nya tetapi juga blok sebelumnya.

3. HASIL DAN PEMBAHASAN

Untuk melihat hasil kinerja implementasi CBC pada algoritma kriptografi klasik ini maka dilakukan pengujian dengan membuat aplikasi sederhana dengan menggunakan bahasa pemrograman JAVA dan kakas pemrograman *NetBeans* serta antar muka sebagai berikut :



Gambar 6 : Program Uji

Pengujian dilakukan dengan memasukkan data plainteks ke aplikasi untuk kemudian dienkripsi dan dibandingkan dengan hasil dari algoritma chiper substitusi periodik tanpa implementasi CBC. Adapun plainteks (P) dan kunci (K) yang akan digunakan adalah plainteks sebagai berikut :

```
P = KRIPTOGRAFI KLASIK DENGAN
    CHIPER ALFABET MAJEMUK
K = LAMPION
```

Dengan menggunakan algoritma chiper substitusi periodik didapatkan chiperteks (C) sebagai berikut :

```
C = VRUEBCTCARXSWNDIWSMBTLNOXXVRCAUU
    IPREMMYMAHV
```

Sedangkan dengan algoritma chiper substitusi periodik yang menggunakan prinsip CBC didapatkan chiperteks sebagai berikut :

```
C = VRUEBCTXRLBTBGAZHTFCZLMVPVFQNMSJ
    DUHRYEHPUOM
```

Dari kedua chiperteks di atas dapat dilihat pada blok teks yang pertama kedua algoritma menghasilkan chiperteks yang sama. Hal ini dikarenakan *Initial Vector* yang digunakan masih berupa karakter *dummy* yaitu “AAAAAAA” (dimana nilai A adalah 0) sehingga ketika dioperasikan *initial vector* tidak akan memberikan perbedaan dengan chiper substitusi periodik yang biasa. Sedangkan untuk blok chiperteks yang berikutnya sudah menunjukkan saling keterkaitan antara satu blok dengan blok yang lain. Hal ini menunjukkan pengaruh prinsip *block chaining* yang diterapkan sehingga menambah variatif pola dari *chipertext* yang dihasilkan. Hal ini tentu akan mempersulit kriptanalis dalam memecahkan chiperteks karena keseluruhan chiperteks terbagi dalam blok-blok yang keseluruhannya saling berkaitan. Kesalahan melakukan kriptanalisis pada satu blok tentu akan memperbesar kesalahan pada blok berikutnya. Hal yang paling beresiko bagi kriptografer adalah jika ***seandainya*** plainteks tidak sengaja dienkripsi dalam 2 modus substitusi periodik (dengan dan tanpa prinsip CBC) dimana kedua salinan chiperteks dar masing-masing modus sampai ke tangan kriptanalis. Kejadian seperti ini justru akan mempermudah kriptanalisis jika dia sebelumnya sudah mengetahui prinsip di balik algoritma tersebut termasuk penggunaan *initial vector* yang cukup sederhana karena pencarian kunci cukup dengan membandingkan blok karakter pertama yang sama dari kedua chiperteks (yaitu “VRUEBCT”) untuk dibandingkan dengan blok karakter kedua dengan panjang yang sama dengan yang pertama (yaitu “XRLBTBG”).

Untuk pengujian terakhir dilakukan dengan melakukan dekripsi chiperteks untuk melihat reabilitas dari algoritma ini. Dengan chiperteks di atas maka dapat diperoleh kembali plainteks semula, yaitu :

```
P = KRIPTOGRAFIKLASIKDENGANCHIPERALF
    ABETMAJEMUK
```

Hal ini membuktikan bahwa penerapan prinsip CBC pada algoritma chiper substitusi periodik tidak akan mengganggu reabilitas dari algoritma itu sendiri.

Di sisi lain penerapan CBC pada algoritma chiper substitusi periodik ini tidak akan berpengaruh pada hasilnya jika panjang plainteks lebih kecil atau sama dengan panjang kunci, hal ini dikarenakan penggunaan blok karakter *dummy* sebagai *initial vector* yang masih sederhana yaitu diinisialisasi dengan karakter ‘A’ sepanjang kunci. Untuk itu pembangkitan ***initial vector*** secara **acak** dengan menggunakan fungsi yang terdefenisi perlu diterapkan untuk menambah kekuatan dari algoritma ini.

Pada akhir eksplorasi penulis mencoba menerapkan pembangkitan ***initial vector (IV)*** ataupun C_0 dengan tidak menggunakan *dummy character* yang sederhana tetapi dengan menggunakan karakter kunci dalam keterurutan yang terbalik (*reverse*) atau dapat dituliskan sebagai persamaan berikut :

$IV_i = K_{(m-1)-i}$ …… *persamaan (7)*
dimana m adalah panjang K

Dengan menerapkan persamaan tersebut maka akan dapat menutupi salah satu kelemahan dari algoritma ini dimana *chipertext* yang dihasilkan dari plaintext P sebelumnya akan menjadi :

```
C = IFCTNCEKFTQFBRNNPIRCKYADEHFBAAAY
    PUSEMMWBUZZ
```

Jika dibandingkan dengan hasil *chipertext* sebelumnya maka *chipertext* terakhir ini memiliki kekuatan (*robustness*) yang semakin sulit untuk dipecahkan. Perbedaan secara menyeluruh jika dibandingkan dengan implementasi CBC yang pertama dikarenakan perubahan *IV* pada blok awal sehingga berantai mempengaruhi blok-blok berikutnya.

Adapun algoritma implementasi terakhir sesuai dengan relevansi dan persamaan-persamaan sebelumnya dapat dilihat pada pseudo code berikut :

```
FUNCTION Dekripsi (STRING  chipertext, STRING
key) → STRING
  STRING  blockChain, plaintext
  CHAR    cc
  INT     i, j, aggregate

  {Inisialisasi Block Chain}
  FOR i=0 → length(key)
    concat (blockChain, key((length(key)-1)-i))

  {Proses Dekripsi}
  j ← 0;
  FOR i=0 → length(chipertext)
    cc      = chipertexti
    IF j >= length(key) THEN
      {Reinisialisasi BlockChain & Iterasi j}
      blockChain         ←         substring
(chipertextlength(plaintext)-length(key)                  →
chipertextlength(chipertext))
      j ← 0
    aggregate  ←  (cc-65)  -  ((keyj-65)  +
(blockChainj-65))
    WHILE (aggregate < 0)
      aggregate ← aggregate + 26
    aggregate ← aggregate + 65
    concat (plaintext,aggregate)
    j ← j + 1

  → plaintext
```

```
FUNCTION Enkripsi (STRING  plaintext, STRING
key) → STRING
  STRING  blockChain, chipertext
  CHAR    cc
  INT     i, j, aggregate

  {Inisialisasi Block Chain}
  FOR i=0 → length(key)
    concat (blockChain, key((length(key)-1)-i))

  {Proses Enkripsi}
  j ← 0;
  FOR i=0 → length(plaintext)
    cc ← plaintexti
    IF j >= length(key) THEN
      {Reinisialisasi BlockChain & Iterasi j}
      blockChain         ←         substring
(chipertextlength(chipertext)-length(key)                 →
chipertextlength(chipertext))
      j ← 0;
      aggregate   ← 65 + (((cc-65) + ((keyj-
65) + (blockChainj-65))) mod 26);
      concat (chipertext,aggregate)
      j ← j + 1

  → chipertext
```

Jadi secara keseluruhan dapat dibuat tabel relevansi final implementasi prinsip CBC pada algoritma kriptografi klasik, yaitu chiper substitusi periodik sebagai berikut :

Tabel 2 : Relevansi Final Prinsip CBC

Prinsip CBC	Klasik	Modern
Satuan Blok	Karakter (*byte*)	Bit
Modus Operasi	Karakter (*byte*)	Bit
Initial Vector	Blok Karakter Kunci dengan Keterurutan Terbalik (*reverse*)	Blok Bit *Dummy* Secara Acak atau dari *user*
Panjang Blok	Sesuai Panjang Kunci (n karakter)	Ditentukan oleh *user* (n bit)
Operasi *Block Chaining*	Penjumlahan, Pengurangan dan Modulo	XOR

4. KESIMPULAN

Algoritma kriptografi klasik dengan metode substitusi periodik masih memiliki peluang untuk dipecahkan dikarenakan pola statis yang dihasilkan pada *chipertext* nya dikarenakan penggunaan parameter kunci yang sama pada setiap operasi bloknya (sepanjang kunci). Implementasi prinsip CBC dari algoritma kriptografi modern terbukti mampu memberi warna, kekuatan (*robustness*) dan variasi yang lebih baik dalam *chipertext* yang dihasilkan meskipun di suatu sisi akan memberikan efek yang sama jika panjang *plaintext* lebih kecil atau sama dengan panjang kunci. Dengan penerapan fungsi pembangkit *initial vector* tersendiri maka hasil enkripsi akan memberikan *chipertext* yang memang berbeda secara keseluruhan dari algoritma klasik chiper substitusi periodik tanpa operasi CBC.

DAFTAR REFERENSI

[1] Munir, Rinaldi. *Diktat Kuliah IF5054 Kriptografi*. Institut Teknologi Bandung. 2006