Studi Perbandingan Secom Cipher Dan VIC Cipher Terhadap Algoritma Kriptography Kunci Simetri Klasik

Indra Sakti Wijayanto
13504029

Program Studi Teknik Informatika
Institut Teknologi Bandung
Jalan Ganesha 10 Bandung 40132

E-mail:
If14029@students.if.itb.ac.id

Abstraksi

Secom cipher merupakan algoritma kriptography kunci simetri, dengan kunci sepanjang 20 karakter. Algoritma ini adalah varian dari VIC cipher. Pada intinya ada 4 tahap yang dilakukan dalam melakukan enkripsi , yaitu: melakukan penghitungan terhadap kunci sehingga dihasilkan kombinasi angka pseudo acak, kemudian membuat kamus untuk melakukan transposisi plainteks dengan angka yang bersesuaian (straddling checkerboard), dan dua kali melakukan transposisi kolom, dimana transposisi yang kedua merupakan transposisi yang susunannya cukup rumit dan acak berdasar perhitungan yang telah dilakukan sebelumnya. Karena tahap yang dilakukan cukup panjang maka algoritma ini dikatakan cukup kuat dan aman, karena sangat sulit melakukan analisis terhadap cipher text untuk menemukan kunci yang digunakan. Sedangkan untuk melakukan dekripsi dilakukan dengan urutan yang terbalik, dengan tetap melakukan perhitungan terhadap kunci yang ada, dan memasukkan cipher text ke dalam kolom transposisi yang kedua sehingga dihasilkan kombinasi angka (pambacaan berdasar kunci transposisi) yang akan diubah menjadi plain teks berdasar straddling checkerboard. Sedangkan cara yang digunakan VIC cipher juga sama, hanya ada perbedaan pada pemilihan dan pemrosesan kunci sampai terbentuk straddling checkerboard. Di akhir makalah ini, penulis akan membandingkan algoritma secom cipher dengan] algoritma kunci simetri yang sudah lebih lama dikenal yaitu double playfair cipher, yang merupakan perbaikan dari playfair cipher, dan vigenere cipher.

Kata kunci : *Secom Cipher, VIC cipher, straddling checkerboard, transposisi kolom, double playfair cipher*

1. Pendahuluan

Algoritma kriptografi tradisional, dimana dalam melakukan enkripsi dan dekripsi menggunakan kunci simetri, telah banyak dipecahkan. Teknik untuk melakukan kriptanalisis biasa dilakukan dengan analisis frekuensi, metode kasiski, ataupun tebak-tebak (*brute force*). Untuk itu diperlukan kunci yang cukup panjang ataupun sangat acak sehingga sulit untuk dilakukan kriptanalisis. Salah satu algoritma yang mencoba untuk menggunakan kunci simetri tapi tetap kuat adalah “Secom Cipher” yang dipublikasikan oleh D. Rijmenants pada tahun 2005 dimana plain text dienkripsi menjadi angka semua. Secom cipher adalah varian dari VIC cipher yang digunakan Uni Soviet untuk berkomunikasi dengan mata-matanya. Dalam makalah ini, kami membahas secom cipher lebih mendalam karena secom cipher dibuat dengan tujuan memperbaiki VIC cipher, juga karena VIC cipher telah lebih dulu populer dan sudah mendapat pembahasan yang cukup. Dengan cipher text yang berupa angka, diharap akan sulit untuk melakukan kriptanalisis terhadap Secom Cipher. Hal ini karena hasil enkripsi (yang berupa angka) tidak berkorespondensi langsung terhadap huruf pada plainteks. Dan seperti algoritma enkripsi lain yang menggunakan kunci simetri, faktor keamanan kunci adalah yang utama . Jika kunci sudah sudah diperoleh dengan benar, maka

cipher text akan dapat segera diubah menjadi plain text.

Algoritma ini cukup menarik dipelajari karena dalam melakukan enkripsi menggunakan beberapa faktor kerumitan perhitungan dalam matematika. Meskipun algoritma ini belum terkenal dan belum bisa dikatakan paling efektif, kita dapat mempelajarinya algoritma ini dengan mengandalkan faktor keamanannya sebagai referensi bagi orang yang ingin menciptakan algoritma yang lebih handal, terutama dengan tetap mengandalkan penggunaan kunci simetri.

2. Algoritma Enkripsi Secom cipher

Dalam melakukan enkripsi dengan algoritma secom cipher dilakukan 4 tahapan utama, yaitu : penghitungan terhadap kunci yang ada sehingga dihasilkan kombinasi angka, pembentukan straddling checkerboard (table untuk melakukan substitusi terhadap plain text dengan angka yang bersesuaian), dan dua kali transposisi kolom.
Secara umum langkah yang dilakukan sebagai berikut:

1. Ambil 20 karakter pertama dari kunci, bagi dua kunci tersebut, masing-masing 10 karakter.
2. Hitung nilai index huruf pada masing-masing bagian kunci.
3. Bangkitkan 50 angka pseudorandom berdasar nilai index pada masing-masing bagian yang telah ditambahkan.
4. Ambil 10 angka terakhir dari 50 angka pseudorandom dan lakukan penghitungan index pada 10 angka tersebut.
5. Gunakan hasil penghitungan sebagai kunci untuk membentuk straddling checkerboard.
6. Ubah plain text berdasar straddling ceckerboard yang terbentuk.
7. Hitung jumlah kolom yang akan dibentuk pada transposisi kolom pertama dan kedua berdasar pada 50 angka pseudo random yang terbentuk.
8. Bentuk kombinasi angka sejumlah kolom yang akan dibentuk berdasar pada 50 angka pseudo random, dengan urutan berdasar nilai index pada bagian kedua dari kunci yang ada yang ditambah dengan kunci pada straddling checkerboard.
9. Lakukan transposisi kolom pertama dengan kunci adalah kombinasi angka yang terbentuk, dengan jumlah sesuai jumlah kolom yang akan dibentuk, dan isi kolom berdasar pada plain text yang telah disubstitusi dengan angka yang bersesuaian pada straddling checker board.
10. Lakukan transposisi kolom yang kedua berdasar pada pembacaan kolom pada kolom transposisi yang pertama, dengan kunci kombinasi angka sejumlah kolom yang akan terbentuk pada transposisi kedua ini.
11. Baca kolom pada transposisi kedua berdasar kunci yang ada dan kelompokkan dalam 5 angka, maka terbentuklah cipher text yang diinginkan.

2.1 Straddling checkerboard

Straddling checkerboard adalah salah satu cara khusus yang sering dipakai oleh mata-mata uni soviet untuk melakukan enkripsi pesan dengan substitusi plain text terhadap angka yang bersesuaian.

Contoh straddling checkerboard yang digunakan dalam melakukan enkripsi, misalnya sebagai berikut:

```
   9 8 2 7 0 1 6 4 3 5
   -------------------
   A T   O N E   S I R
2  B C D F G H J K L M
6  P Q U V W X Y Z . /
```

Dalam secom cipher, ada perbedaan dalam penempatan huruf, tetapi konsepnya sama. Cara pembentukannya akan diterangkan lebih lanjut dalam contoh melakukan enkripsi (Straddling checker board diatas adalah yang digunakan oleh VIC cipher).

Dalam straddling checkerboard ini setiap huruf dapat diganti dengan satu atau dua digit, inilah salah satu keunggulan straddling checkerboard, yang disebut juga prefix property (seperti halnya yang ada dalam kode huffman).

2.2 Transposisi kolom

Dalam algoritma secom cipher ini kita menggunakan dua kali transposisi kolom. Yang pertama adalah transposisi kolom sederhana (biasa) dan transposisi kolom yang kedua adalah transposisi dengan pengaturan tempat yang agak rumit (sering disebut disrupted transposition),

jadi tidak bisa dibuat aturan secara umum tentang penempatan karakter didalam kolom transposisi.

Secara umum, transposisi kolom adalah suatu cara untuk mengubah susunan pembacaan suatu pesan dengan menuliskannya dalam baris per baris, kemudian membacanya per kolom sesuai dengan index dari kunci yang ada.

2.3 Contoh Enkripsi

Dalam hal ini penulis akan memberikan contoh dalam melakukan enkripsi, sebagai berikut:

Plain text : institut teknologi bandung jalan ganesha 10 bandung

Kunci : teknik informatika itb bandung

Langkah 1:

20 karakter pertama, dengan menghilangkan spasi adalah : teknikinformatikaitb

Bagi menjadi 2 bagian, menjadi : teknikinfo dan rmatikaitb. Jika ada huruf besar, maka kita ubah semua menjadi lowercase (hal ini berlaku untuk kunci maupun plain text yang diberikan).

Langkah 2:

Nilai index dihitung, dengan memberi nilai 1 pada huruf yang terendah (paling awal pada alfabet) dan seterusnya, sehingga kita memberi nilai 0 untuk huruf yang tertinggi. Jika ada huruf yang sama, maka kita memberi nilai lebih kecil untuk huruf yang lebih awal muncul. Pada contoh ini, perhitungan menghasilkan sebagai berikut:

```
teknikinfo   rmatikaitb
0157364829   8719462503
```

Langkah 3:

Tambahkan nilai penghitunga index kedua bagian, dengan mengabaikan *carry* (hanya memperhatika satuan nya saja), sebagai berikut :

```
0157364829
8719462503   +
8866726322
```

Bangkitkan 50 angka pseudo random berdasar hasil penjumlahan ini (digunakan sebagai kunci), yaitu dengan menambahkan setiap 2 angka, dimulai pada awal kunci dan maju terus ke angka berikutnya sampai terbentuk 50 angka, yaitu dengan 5 baris dan 10 kolom (penghitungan juga hanya memperhatikan satuannya saja). Hasil pembangkitan angka sebagai berikut:

```
8866726322
----------
6423989548
0652774928
6179413104
7863544141
5498985556
```

Langkah 4 :

Pada 50 angka pseudorandom diatas, diambil 10 angka terakhir, yaitu pada baris terakhir. Kemudian dari 10 angka tersebut,kita hitung kembali index huruf seperti sebelumnya, hanya saat ini kita menggunakannya untuk angka bukan huruf. Hasil penghitungan index sebagai berikut:

```
5498985556
2197083456   à index yang didapat.
```

Langkah 5 :

Pembentukan straddling checkerboard, dengan kunci index yang telah terbentuk diatas. Caranya adalah dengan meletakkan kunci pada baris pertama, kemudian pada baris kedua kita letakkan huruf-huruf yang mempunyai frekuensi tertinggi pada teks berbahas inggris, dalam hal ini secom cipher memakai huruf “e s t o n i a ” , menyesuaiakan baris pertama(kunci yang ada) dan dengan memberikan spasi (ruang kosong) pada huruf ketiga, keenam, dan kesembilan. Kemudian kita letakkan angka ketiga, keenam, dan kesembilan dari kunci yang diberikan pada bagian paling kiri dari straddling checkerboard (anggap sebagai kolom ke0, dengan satu angka pada tiap baris, dimulai pada baris ketiga). Kemudian pada baris ketiga kita letakkan huruf – huruf alphabet yang belum digunakan pada baris kedua. Dan awal kolom peletakan huruf pada

setiap baris berdasar pada angka pada kolom ke 0 baris yang bersesuaian. Dan kita tambahkan karakter * pada baris keempat ,yang menandakan spasi untuk menggenapkan menjadi 10 karakter. Pada baris kelima kita letakkan angka, dimulai dari 1, dan angka terakhir adalah 0.

Karakter yang digunakan untuk menggenapkan straddling checkerboard, adalah sebagai berikut:

```
Baris Ketiga:B C D F G H J K L M
Baris Empat:P Q R U V W X Y Z *
Baris Kelima  : 1 2 3 4 5 6 7 8 9 0
```

Hasil penghitungan straddling checkerboard, adalah:

Kunci **à** 21**9**70**8**34**5**6

```
  | es to ni a
9| dfghjklmbc
8| uvwxyz*pqr
5| 7890123456
```

Dalam straddling checkerboard diatas, ruang kosong pada baris pertama berkorespondensi dengan angka 9, 8 , 5 pada kunci yang digunakan, sehingga pada baris ketiga sampai baris kelima kita gunakan angka tersebut untuk memulai peletakan karakter (ditandai dengan garis bawah, menunjukkan awal peletakan).

Langkah 6 :

Kita ubah semua huruf , angka,dan spasi pada plain text menjadi angka-angka yang bersesuaian berdasar straddling checkerboard yang telah terbentuk.

Cara yang kita lakukan untuk mengubah huruf pada baris kedua (‘e’,’s’,’t’,’o’,’n’,’i’,’a’) adalah dengan mengambil angka yang bersesuaian pada kunci yang ada (sesuai kolom yang bersesuaian). Sedangkan pada baris lain, dilakukan dengan membaca pada kolom ke0 (sebagai angka puluhan) diikuti angka pada kunci sesuai dengan kolom masing-masing. Misal: karakter ‘c’ akan diganti dengan angka 96, demikian seterusnya.

Hasil pengubahan plain text akan menghasilkan kombinasi angka sebagai berikut:

```
institut*teknologi*
```

```
43174782783729830930994 83
bandung*jalan*
95639282399839069363 83
ganesha*10*
99632197683505783
Bandung
95639282399
```

Dalam contoh ini, kita tetap mempertimbangkan spasi dan mengubahnya dengan angka yang bersesuaian. Dimana pada straddling checkerboard ditandai dengan karakter *. Namun agar lebih aman, bisa juga kita mengabaikan spasi pada plain text kita.

Langkah 7 :

Untuk menghitung jumlah kolom yang akan digunakan pada transposisi kolom pertama dan kedua, kita akan mulai mencari angka yang unik (berbeda) pada 50 angka pseudo random yang telah terbentuk, mulai angka yang terakhir. Dari kanan ke kiri kita jumlah kan angka yang berbeda sampai menghasilkan angka lebih besar dari 9, demikian juga untuk jumlah kolom untuk transposisi kedua, dilakukan dengan menjumlahkan angka dimulai setelah angka yang terakhir digunakan pada penghitungan yang pertama. Dalam contoh ini, penghitungan kita akan menghasilkan kombinasi angka sebagai berikut:

Kolom pada trasposisi pertama: 6+5 =11
Kolom pada trasposisi kedua : 5+8 =13

Kita pilih angka 5 dan 8 untuk membentuk transposisi kedua (bukannya 5 + 5) karena kedua angka yang dijumlahkan tidak boleh sama.

Langkah 8:

Kita akan mulai membentuk kombinasi angka untuk menghasilkan kunci pada transposisi kolom.

Mula – mula kita akan menulis kembali 50 angka pseudo random yang telah kita bentuk, dengan kunci yang dihasilkan dengan menambahkan kunci pada straddling checkerboard dengan penghitungan index pada bagian kedua kunci awal(kunci awal: teknik informatika itb).

Penghitungan ini, dengan mengabaikan carry, menghasilkan sebagai berikut:

21**9**70**8**34**5**6 **à** kunci pada straddling checkerboard

8719462503 **à** index pada kunci awal bagian kedua.

21**9**70**8**34**5**6
8719462503
———————— +
0806445959

angka ini akan kita gunakan sebagai kunci, dan ditulis sebagai berikut:

0806445959
——————

6423989548
0652774928
6179413104
7863544141
5498985556

Dari langkah ke 7 kita mengetahui bahwa kita akan membentuk total 24 kolom untuk transposisi pertama dan kedua. Sehingga kita juga memerlukan kunci sebanyak 24 karakter. 24 karakter yang akan digunakan sebagai kunci akan kita bentuk dari kombinsai 50 angka dengan kunci diatas (0806445959). Kita baca per kolom dengan kunci kita gunakan sebagai index urutan pembacaan.
Maka kita dapatkan 24 angka sebagai berikut:

97459871489 4345420453293

Langkah 9:

Dari kunci yang telah terbentuk diatas, kita bisa memulai membentuk transposisi kolom yang pertama, yang sebenarnya sangat simple. Yaitu dengan meletakkan kunci (bagian pertama pada 24 angka diatas) sebagai baris pertama dan melanjutkan dengan mengisikan plaintext yang telah disubstitusikan dengan angka yang bersesuaian (sesuai dengan checker board). Dalam hal ini kita akan melakukan proses selanjutnya dalam grup yang terdiri dari 5 angka, sehingga karena kita mengetahui bahwa kode plain text diatas sudah kelipatan 5(berjumlah 75 karakter), maka kita tidak perlu menambahkan karakter 0 pada akhir pembacaan, agar jumlah yang diproses dapat genap dibagi 5. jika hal ini dilakukan maka pada proses dekripsi nanti akan menghasilkan adanya karakter tambahan di akhir pesan.

Hasil pembentukan tranposisi kolom yang pertama, dalam contoh ini akan menghasilkan sebagai berikut:

97459871489
43174782783
72983093099
48395639282
39983906936
38399632197
68350578395
639282399

Langkah 10 :

Untuk membentuk transposisi kolom yang kedua, kali ini sedikit lebih complex. Hal ini sengaja dibuat agar tidak mudah dipecahkan oleh para criptanalist.
Sebelumnya kita akan mulai membaca per kolom dari hasil transposisi kolom yang pertama. Dimana kita membaca berdasar kunci yang ada (urut index). Setelah kita baca per kolom, maka dihasilkan kombinasi angka sebagai berikut:

2396289 1939339 7029139
7898952 3289883 8930373
7069652 898399 4743366
4353908 392675

Setelah pembacaan selesai, kita akan membentuk transposisi kolom kedua berdasar kunci yang sebelumnya telah kita dapatkan, yaitu 10 angka pada bagian kedua dari 21 angka yang telah kita peroleh pada langkah 8.

Pada proses pembentukan, kita akan mulai dengan membentuk daerah segitiga pada trasposisi kita, dengan cara sebagai berikut:
1. Perhatikan index yang terkecil (yang belum digunakan) pada kunci.
2. Segitiga yang kita bentuk adalah segitiga siku-siku dengan siku-siku pada kanan atas. Sedangkan index terkecil kunci , maka kolom yang bersesuaian dengan baris paling awal ruang segitiga yang akan kita bentuk (mulai dari baris pertama) menjadi salah satu titik sudutnya.
4. titik sudut ketiga adalah kolom paling kanan dari tranposisi kedua kita jika memang dapat mencapai kolom terakhir.

Segitiga yang kita bentuk tadi menandakan ruang kosong pertama. Jika diperlukan, kita bisa membentuk segitiga yang kedua, ketiga, dan seterusnya. Hal ini berdasar pada kode angka pada plain text, jika masih tersisa maka sangat mungkin dilakukan penambahan pembentukan ruang segitiga.. Hal ini bisa kita hitung dengan membagi jumlah kode plain text dengan kunci yang terbentuk untuk transposisi kolom ini. Pembagian ini akan menghasilkan jumlah baris yang perlu kita bentuk (batas segitiga berdasar pada jumlah baris yang penuh), dan juga jumlah kolom yang akan terbentuk pada baris terakhir.

Maksud segitiga sebagai ruang kosong awal adalah kita mula-mula mengisikan kombinasi kode, yang telah terbentuk pada transposisi pertama, pada bagian diluar ruang segitiga yang telah kita bentuk.

Dalam contoh yang kita gunakan saat ini ,kode yang ada pada transposisi pertama berjumlah 75 angka, berarti kita bisa memastikan akan terbentuk 5 baris yang berisi 13 kolom dan 1 baris sisa yang berisi 10 kolom saja. Hasil penghitungan ini akan terbentuk ruang segitiga kosong sebagai berikut:

```
4345420453293
23962--------
891939-------
3397029------
13978989-----
523289883----
8930373706
```

pada langkah ini ruang segitiga ditandai dengan karakter '-'. Karena sebelumnya kita mengetahui kita hanya akan dapat membentuk 6 baris yang kolomnya terisi penuh, maka pembentukan ruang segitiga juga sampai baris ke 6 saja.

Setelah terbentuk ruang segitiga, kita lanjutkan pembentukan transposisi kedua ini dengan menggenapkan ruang segitiga (yang masih kosong), yaitu mengisinya dengan nilai kelompok angka yang terbentuk pada tranposisi pertama yang belum digunakan pada tranposisi kedua diatas.

Proses pengisian ini cukup mudah, yaitu kita langsung memasukkan angka per baris, mengisi ruang segitiga yang telah terbentuk.

Hasil pengisian ruang segitiga kita akan menghasilkan transposisi kolom yang kedua, sebagai berikut:

```
4345420453293
2396296528983
8919399947433
3397029664353
1397898990839
5232898832675
8930373706
```

Langkah 11 :

Langkah terakhir yang akan kita lakukan adalah membentuk cipher text dengan membaca hasil tranposisi kolom kedua yang telah kita lakukan.

Pembacaan kita lakukan perkolom, dengan urutan sesuai dengan kunci yang ada pada baris paling atas. (dengan aturan seperti biasa, terurut naik dengan index terkecil 1 dan paling besar 0). Setelah kita baca per kolom, maka kita langsung kelompokkan hasil pembacaan kita dalam kelompok yang terdiri dari 5 angka, dimana hal ini menjadi salah satu ciri khas dari secom cipher.

Hasil pembacaan yang telah dikelompokkan, adalah sebagai berikut :

```
99299 79438
63933 29874
02633 39528
31589 19933
23088 35969
87697 72024
69308 35376
99883
```

Cipher text diatas dibaca perbaris, dapat divariasikan juga dengan membaca cipher text tanpa pengelompokan.

Demikianlah cara melakukan enkripsi dengan contoh yang diberikan.

3. Algoritma Dekripsi Secom cipher

3.1 Cara Melakukan Dekripsi

Secara intuitive dekripsi dapat dilakukan dengan melakukan operasi kebalikan dari proses enkripsi yang dilakukan . Dalam hal ini jika kita mengetahui kunci dengan benar, maka seharusnya proses dekripsi akan dapat dilakukan dengan cepat dan benar.

Secara singkat, sebagaimana dalam melakukan enkripsi, dalam melakukan dekripsi tahap yang kita lakukan adalah :

1. Melakukan penghitungan index terhadap kunci yang ada, termasuk generate 50 angka pseudo random sampai menghasilkan straddling checkerboard.
2. Kita lakukan penghitungan kunci untuk transposisi kolom pertama dan kedua. Cara yang digunakan sama persis dengan cara melakukan dekripsi.
3. Kita mulai pada transposisi kolom kedua, kita bisa membentuk ruang kosong (ruang segitiga) berdasar index kunci yang ada. Masukkan digit pada cipher text pada transposisi kolom kedua sesuai dengan index.
4. Baca digit pada transposisi kolom ini per baris yang ada di luar ruang segitiga/ ruang kosong (kita ketahui dari index kunci yang ada). Setelah semua angka yang ada diluar ruang kosong selesai dibaca, kita lanjutkan dengan pembacaan digit didalam ruang kosong. Pembacaan juga dilakukan per baris.
5. Hasil pembacaan akan menjadi masukan untuk transposisi kolom pertama. Karena sebelumnya kita sudah mengetahui kunci yang akan digunakan pada transposisi kolom pertama, maka langsung saja kita masukkan hasil pembacaan transposisi kolom kedua tadi ke transposisi kolom pertama per kolom sesuai dengan index kunci yang ada.
6. Langkah terakhir yang kita lakukan adalah dengan membaca nilai pada transposisi kolom pertama yang telah kita bentuk per baris. Maka kita telah mendapatkan representasi plaintext dalam bentuk digit yang berkorespondensi. Untuk mendapatkan pesan plaintext yang sesungguhnya, kita melakukan substitusi terhadap digit yang kita dapatkan tadi sesuai dengan straddling checkerboard yang telah kita bentuk.

3.2 Contoh Dekripsi

Kita akan menerapkan cara diatas untuk melakukan dekripsi terhadap contoh yang kita gunakan pada proses enkripsi diatas.

Cipher text yang diberikan adalah :

```
99299 79438
63933 29874
02633 39528
31589 19933
23088 35969
87697 72024
69308 35376
99883
```

Langkah 1 :
Kita menghitung index kunci dengan tahapan yang sama dengan cara yang kita lakukan dalam melakukan enkripsi, maka akan didapat straddling checkerboard sebagai berikut :

```
   21970834 56
  ------------
  | es to ni a
9 | dfghjklmbc
8 | uvwxyz*pqr
5 | 7890123456
```

Kunci : `teknik informatika itb bandung`

Langkah 2 :

Lakukan cara yang kita gunakan pada proses enkripsi, sampai kita mendapatkan kunci yang akan digunakan pada transposisi kolom. Penghitungan ini menghasilkan nilai sebagai berikut :

`97459871489` **à** kunci transposisi pertama
`4345420453293` **à** kunci tranposisi kedua

Langkah 3 :
Kita masukkan nilai digit pada cipher text kedalam tranposisi kolom kedua per kolom sesuai dengan index kunci yang ada. Hasil dari proses ini adalah, sebagai berikut :

`4345420453293`

2396296528983
8919399947433
3397029664353
1397898990839
5232898832675
8930373706

72983093099
48395639282
39983906936
38399632197
68350578395
639282399

Langkah 4 :

Setelah membentuk tranposisi kedua, kita akan membacanya per baris, dimulai dengan baris pertama dan seterusnya dan kita baca dulu semua baris yang ada di luar ruang kosong. Ruang kosong ini kita ketahui dari index kunci (cara menentukan ruang kosong ini dapat dilihat lebih jelas pada proses enkripsi). Setelah semua baris di luar ruang kosong selesai kita baca, kita lanjutkan membaca semua baris di dalam ruang kosong, kita lakukan per baris juga urut dari baris pertama dan seterusnya.

Hasil pembacaan baris pada transposisi kolom kedua ini akan digunakan sebagai masukan pada transposisi kolom pertama, dengan hasil sebagai berikut :

23962891939339702913978989523289883893737069652898399474336643539086392675

Langkah 5 :

Kita mulai membentuk transposisi kolom pertama dengan mempertimbangkan kunci yang ada. Yaitu kita masukkan angka hasil pembacaan transposisi kedua per kolom sesuai dengan index pada kunci yang ada.

Hasil yang terbentuk pada transposisi kolom pertama adalah :

97459871489
43174782783

Langkah 6 :

Pada langkah terakhir ini kita baca angka pada transposisi kolom pertama diatas perbaris, mulai dari yang pertama.

Hasil pembacaan adalah representasi digit dari plain text, dengan membandingkannya dengan straddling checkerboard yang telah ada (kit abaca seperti pembacaan pada proses dekripsi, yaitu kolom ke0 sebagai puluhan, dan angka pada kunci sebagai satuan), maka kita akan mendapatkan representasi seperti ini:

43174782783729830930994831
institut*teknologi*

9563928239983906936383
bandung*jalan*

99632197683505783
ganesha*10*

95639282399
Bandung

Supaya lebih jelas kita pisahkan plain teks dengan spasi yang ditandai dengan *, Maka dihasilkan:

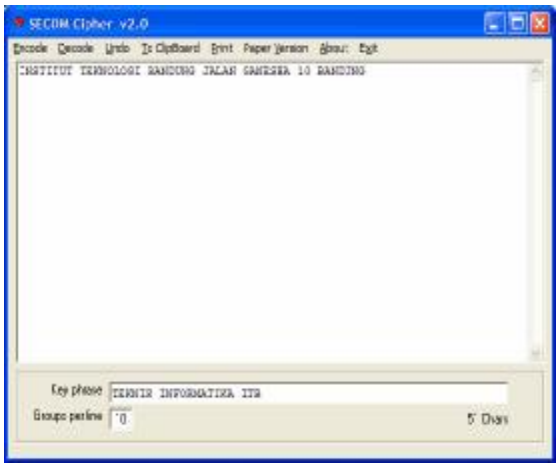
Plain text :
institut teknologi bandung jalan ganesha 10 bandung

Demikianlah proses dekripsi yang bisa kita lakukan dengan algoritma secom cipher

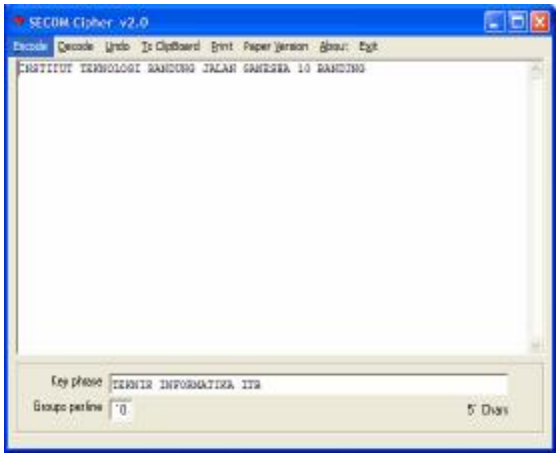
4. Overview Program Secom cipher

Dalam web site yang saya kunjungi sudah tersedia program secom cipher yang siap digunakan untuk melakukan enkripsi dan dekripsi yang menggunakan tools borland delphi.

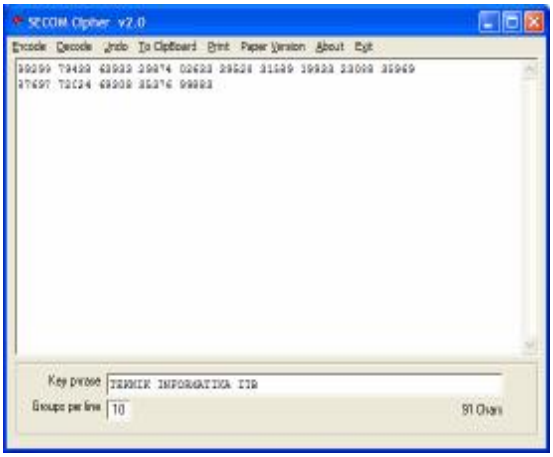
Pada awal program, kita memasukkan plain teks dan juga kunci dengan panjang minimal 20 karakter.
Plain text kita masukkan dalam suatu rich text box, sedangkan kunci ada pada textbox.



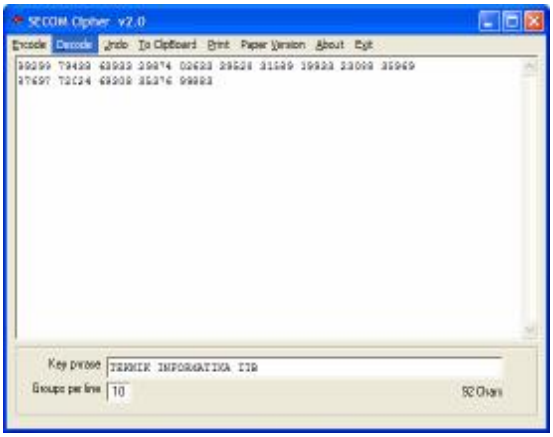
Untuk melakukan enkripsi kita menekan tab encode dan program otomatis akan mengenkripsi plaintext kita.



Hasilnya enkripsi akan langsung ditulis di rich text box yang telah tersedia.



Untuk melakukan dekripsi, caranya sama yaitu dengan menekan tab decode maka hasil plain text akan ditampilkan di rich text box itu juga.



Untuk mengetahui pembuat program , disediakan menu About, akan muncul semacam alert seperti gambar dibawah ini :

Mengenai dokumentasi dan gambaran umum program disediakan di file yang diikutkan dalam folder instalasi program secom cipher ini

5. VIC Cipher

Pada dasarnya VIC cipher adalah pendahulu dari secom cipher, dan telah digunakan oleh mata-mata uni soviet dalam kegiatan spionase. Algoritma enkripsi dan dekripsi bisa dikatakan sangat mirip, bahkan hanya berbeda dalam pemilihan kunci dan beberapa proses diawal yang sebenarnya tidak mengubah konsep.

5.1 enkripsi

Secara garis besar tahapan enkripsi meliputi penghitungan kunci (*key phrase*), pembentukan straddling checkerboard, dan dua kali transposisi kolom (cara yang digunakan dalam transposisi kolom sama persis dengan cara yang digunakan secom cipher). Dalam bagian ini, hanya akan dibahas langkah –langkah yang mempunyai perbedaan dengan secom cipher.
Langkah yang digunakan sebagai berikut :

1. Untuk menggunakan VIC cipher, kita harus mengingat 6 digit tanggal kita melakukan enkripsi (misal 30 September 2006 menjadi 930200). Kemudian sebagai kunci kita harus mengingat 20 karakter kunci (sama seperti secom cipher). Dan yang terakhir kita juga harus mengingat 5 digit angka acak, terserah kita, sebagai indikator.
2. langkah kedua sama dengan secom cipher yaitu menghitung index dari 20 karakter kunci yang sebelumnya telah dikelompokkan dalam 2 bagian 20 karakter.
3. pada pembangkitan 50 angka acak, kunci yang digunakan cukup berbeda dengan cara menghasilkan kunci pada secom cipher. Dalam VIC cipher, cara yang digunakan adalah sebagai berikut:
    a. jumlahkan 5 digit pertama dari 6 digit tanggal dengan 5 digit angka indicator (abaikan carry).
    b. Dari hasil penjumlahan tadi kembangkan 5 digit angka tersebut menjadi 10 digit seperti pembangkitan 50 angka acak pada secom cipher, yaitu menambahkan setiap 2 angka yang bersebelahan.
    c. Tambahkan 10 angka hasil pengembangan dengan index bagian pertama dari kunci.
    d. Kemudian kita hitung index dari hasil penjumlahan tersebut. Penghitungan index ini berbedadengan sebelumnya dimana angka yang sama akan mendapat index yang sama pula (pada secom cipher, penghitungan index akan memberikan index yang lebih kecil untuk angka yang sama yang muncul lebih awal). Hasil penghitungan index ini akan menjadi kunci dalam pembangkitan 50 angka pseudorandom (caranya sama dengan secom cipher).
4. langkah ke 4 sama dengan tahapan pada secom cipher yaitu penghitungan index dari 10 angka terakhir dari 50 angka pseudorandom yang terbentuk.
5. pembentukan straddling checkerboard, agak berbeda dengan secom cipher karena pada VIC cipher ini tidak ada baris yang berisi angka, dan pada baris kedua kita tidak menuliskan ‘estonia’ , seperti pada secom cipher, tapi kita menuliskan “AT ONE SIR” dengan mengosongkan kolom ke 3 dan ke 7, sehingga pada kolom ke 0 hanya didapat 2 angka saja (hal ini karena VIC cipher tidak mempunyai baris yang berisi angka saja). Perbedaan lain adalah digunakannya karakter ‘.’ dan ‘/’ menggantikan karakter ‘*’ pada secom cipher (ada dua karakter, karena pada VIC cipher baris pertamanya sudah berisi 8 karakter, sehingga sampai baris keterakhir masih tersisa 2 tempat

kosong). Contoh straddling checkerboard VIC cipher:

```
   9 8 2 7 0 1 6 4 3 5
   -------------------
   A T   O N E   S I R
2  B C D F G H J K L M
6  P Q U V W X Y Z . /
```

6. langkah selanjutnya (6 sampai dengan 11) sama persis dengan apa yang dilakukan pada secom cipher. Perbedaannya hanyalah penambahan karakter ‘9’ untuk menggenapkan cipher text yang belum habis dibagi 5 (pada secom cipher ditambah karakter ‘0’).

5.2 dekripsi

Secara umum algoritma dekripsi yang digunakan sama persis dengan algoritma dekripsi yang digunakan pada secom cipher. Perbedaan yang utama hanyalah pada proses pemrosesan kunci dan pembentukan straddling checker board (mengenai perbedaan pembentukannya dapat melihat pada proses enkripsi VIC cipher). Seteleh terbentuk straddling checker board, maka langkah selanjutnya sama persis. Hal ini karena pada prinsipnya proses dekripsi hanyalah proses kebalikan dari enkripsi, dimana hal ini dapat diperoleh dengan menelusuri proses enkripsi yang dilakukan.

6. Analisis dan perbandingan

6.1 Analisis

Secom Cipher dan VIC cipher merupakan salah satu jenis algoritma kriptography klasik, yaitu algoritma yang proses pembuatan cipher textnya dapat dilakukan dengan kertas dan pensil saja (tanpa bantuan komputer). Namun, bukan berarti karena kita dapat melakukan enkripsi pesan dengan pensil dan kertas saja, berarti kita akan mudah dalam melakukan kriptanalisisnya. Ada beberapa faktor kompleksitas yang menjadi pendukung bagi kehandalan secom cipher sehingga sulit untuk dipecahkan (terutama jika hanya menggunakan pensil dan kertas saja).

Karena dalam proses enkripsi dan dekripsi]menggunakan cara substitusi dan transposisi yang divariasikan (adanya straddling checkerboard dan disrupted transposition)., maka secom cipher dan vic cipher termasuk kategori super enkripsi, yaitu proses enkripsi yang menggabungkan metode cipher substitusi dan cipher transposisi.

Dengan melihat cara enkripsi-dekripsi, hasil cipher text, dan hasil dari proses dekripsi maka penulis dapat menemukan/menganalisis beberapa kelebihan dan kekurangan dari segi algoritmanya sendiri.

6.1.2 Kelebihan

Dengan melihat cipher text yang dibandingkan dengan plain textnya kita akan langsung melihat adanya kelebihan dari secom cipher, yaitu tidak adanya hubungan langsung antara cipher text yang dibentuk dengan plain text yang diberikan. Hal ini menjadi suatu kekuatan untuk bertahan dari usaha kriptanalisis untuk mencoba menganalisis frekuensi dari cipher text, juga usaha untuk menebak plain text dari potongan cipher teks (known plain text attack). Penggunaan kunci sepanjang 20 karakter bisa jadi menjadi suatu kelebihan, dalam hal banyaknya kombinasi kunci yang harus dicoba jika ingin memcahkan cipher text dengan mencoba semua kemungkinan kunci (*brute force*) yaitu sebanyak 10^{20}.

Penggunaan straddling checkerboard untuk melakukan substitusi terhadap plain text juga menjadi suatu kekuatan tersendiri. Hal ini bermanfaat untuk mengacaukan pengamatan kriptanalis karena setiap huruf abjad dapat berkorespondensi dengan satu ataupun dua digit. Misal huruf ‘a’ berkorespondensi dengan angka 3, huruf ‘b’ dengan angka ‘5’,huruf ‘c’ dengan 35, maka ketika muncul pasangan angka 35 dalam cipher text, tidak bisa dipastikan langsung apakah itu satu atau dua huruf di plain textnya.

Proses enkripsi dan dekripsi dapat dilakukan dengan manual, sehingga bisa menjadi contoh yang cukup baik untuk belajar. Algoritma yang menggabungkan beberapa cara yang biasa digunakan dalam kriptography klasik, seperti substitusi dan transposisi. Hal ini terutama sebagai pertimbangan bagi para ahli dibidang kriptography untuk menyusun algoritma kriptography modern yang memanfaatkan kunci simetri.

Usaha untuk mencoba – coba kunci tidak akan berhasil melihat kemungkinan plain text jika ada karakter kunci yang salah dimasukkan. Sedangkan usaha merubah blok cipher text hanya akan merubah beberapa huruf dari plain textnya tanpa merubah keseluruhan, tetapi jika kita menambah ataupun mengurangi jumlah blok cipher text maka hal itu akan mempengaruhi keseluruhan plain text yang dihasilkan sehingga pesan yang terbaca menjadi tidak berarti, bahkan sulit ditebak maksudnya.

6.1.3 Kekurangan

Kekurangan algoritma ini cukup banyak juga. Salah satu yang cukup jelas adalah pembengkakan ukuran cipher text dari plain text yang diberikan. Hal ini dapat terjadi karena setiap huruf pada plain text dapat disubstitusikan satu atau dua digit di plain text, sebab lainnya adalah penambahan karakter (misalnya ‘0’) untuk cipher text yang belum merupakan kelipatan lima.

Kelemahan lain adalah yang diakibatkan oleh penambahan karakter ‘0’ tadi adalah adanya karakter tambahan pada plain text hasil dekripsi yaitu dengan karakter yang pada straddling checkerboard dikodekan dengan ‘0’. Dalam beberapa kondisi hal ini tidak terlalu bermasalah, karena karakter yang ditambahkan sama. Tetapi penambahan ini akan menjadi persoalan tersendiri ketika karakter tambahan tadi hanya sebanyak satu atau dua karakter dan tambahan karakter ini dapat memberikan semantik yang berbeda pada plain text.

Algoritma secom cipher, termsuk juga VIC cipher, juga kurang efektif karena hanya mampu mengkodekan huruf dalam abjad biasa, angka, dan spasi. Tentu hal ini menjadi tidak relevan lagi digunakan saat ini karena kebanyakan kita tidak hanya ingin mengenkripsi teks, tapi juga gambar, suara, dan pesan lain yang hanya bisa dimanipulasi dengan baik jika kita beroperasi dalam mode byte. Jika kita ingin memperbesar straddling checkerboard yang terbentuk, maka tetap saja tidak akan bisa melebihi 110 karakter , yaitu 10 karakter untuk representasi satu angka, dan 100 karakter untuk representasi dua angka. Maximal berjumlah 110 karakter bukannya 100 karakter karena representasi ‘01’ berbeda dengan ‘1’. Hal ini karena kunci yang digunakan adalah dalam bentuk angka yang domainnya terbatas. Tentu saja hal ini masih menyulitkan jika kita ingin mengkodekan semua kode dalam byte yang mungkin (256 karakter).

Kekurangan lain yang dimiliki, secara umum diwarisi dari jenis algoritma kriptography klasik, yaitu mengenai keamanan kunci. Penggunaan kunci simetri mengharuskan proses enkripsi dan dekripsi mnegunkan satu kunci yang sama. Hal ini sangat sulit dilakukan pada saat ini karena salah satu tujuan kriptography adalah untuk pengamanan komunikasi data. Tentu saja jika untuk melakukan pengamanan harus menggunakan kunci yang juga harus diamankan penggunannya, maka hal ini bisa menjadi sangat tidak efektif.

6.2 Perbandingan

6.2.1 Perbandingan Secom dengan VIC cipher

Pada pembahasan VIC cipher kita telah mengetahui bahwa secom cipher sebenarnya adalah varian dari VIC cipher. Jadi, kita tidak bisa membandingkan perbedaan algoritma yang digunakan secara gamblang.

Bahkan secara intuitive kita bisa mengatakan bahwa kedua algoritma ini

sama. Perbedaan yang tampak adalah proses pemilihan kunci yang digunakan, terutama dalam proses pembentukan straddling checkerboard. Pada VIC cipher kita harus mengingat tanggal melakukan enkripsi dan 5 digit angka sebagai indicator untuk proses penghitungan kunci. Hal ini cukup menimbulkan kesulitan karena semakin banyak yang harus kita hafalkan selain kunci yang panjangnya 20 karakter. Tentu saja hal ini menambah kekuatan keamanan dari VIC cipher, karena proses dekripsi tidak hanya membutuhkan 20 karakter kunci, tapi juga tanggal enkripsi dan 5 angka indicator. Hal ini berbeda dengan secom cipher yang hanya membutuhkan 20 karakter kunci untuk melakukan dekripsi, sehingga kemungkinan melakukan kripatanalisis dengan benar lebih besar. Namun begitu, faktor kelemahan ingatan dari seorang agent (pelaku enkripsi) menjadi tantangan tersendiri, karena angka yang dihafalkan tidak punya pola yang mudah dihapal tanpa harus menuliskannya.

Meskipun VIC cipher bisa jadi lebih aman, tapi secom cipher lahir untuk menambah fleksibilitas VIC cipher, hal ini terbukti dengan adanya tambahan jumlah baris pada straddling checkerboard, sehingga jenis karakter plain teks yang dapat dikodekan dapat bertamabah (yaitu dengan menangani angka juga).

6.2.2 Perbandingan dengan algoritma kunci simetri klasik

Untuk menguji kehandalan secom cipher ataupun VIC cipher, kita juga akan membandingkannya dengan algoritma klasik lain, yaitu double playfair cipher dan vigenere cipher. Kita tidak membandingkan dengan algoritma modern karena algoritma modern kebanyakan mengandalkan operasi byte sehingga mempunyai perbedaan yang mendasar.

6.2.2.1 Double Playfair cipher

Double Playfair cipher adalah salah satu algoritma perbaikan dari playfair cipher. Sebagaimana playfair cipher, double playfair cipher juga menggunakan segiempat 5 x 5, tetapi berjumlah dua dengan menggunakan dua kunci yang berbeda. Pengisian segiempat ini sama dengan playfair cipher, hanya untuk segiempat kedua pengisian dilakukan dengan urutan spiral, yaitu dimulai dengan kolom pertama kebawah kemudian kolom kedua dimulai dari bawah dan seterusnya. Misalnya kunci yang digunakan adalah Hamburg dan NewYork maka segiempat akan menjadi sebagai berikut :

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

Kemudian perbedaan selanjutnya adalah penyusunan pasangan abjad plain text yang dienkripsi. Dalam double playfair cipher, plain text dikelompokkan dalam 7 karakter, dan kelompok kedua dibawah kelompok kedua, dan seterusnya. Untuk dua kelompok terakhir, jika jumlahnya tidak sampai 7 untuk tiap kelompok maka digenapkan dengan jumlah yang sama untuk setiap kelompok. Jika jumlahnya ganjil maka kita tambahkan karakter X untuk menggenapkannya. Misalnya plain text "MY HOVERCRAFT IS FULL OF EELS" maka akan dipecah menjadi :

```
MYHOVER FULLO
CRAFTIS FEELS
```

Kemudian ada perbedaan dalam melakukan enkripsi, yaitu enkripsi dilakukan dua kali. Pada setiap tahap dibaca pasangan huruf secara vertical, pada contoh diatas pasangan pertama yaitu MC, kemudian huruf pertama kita letakkan di segiempat pertama dan huruf kedua kita pada segiempat kedua. Hasilnya adalah titik sudut yang terbentuk dari kotak yang terbentuk dengan M sebagai titik kiri atas dan C sebagai titik kanan bawah, dibaca dari titik kanan atas kemudian kiri bawah. Dalam hal ini hasil pembacaan pertama adalah GQ. Untuk tahap kedua yang dienkripsi adalah hasil pembacaan pertama (GQ) dengan cara yang sama dihasilkan MP. Maka pasangan MC di plain text akan dikodekan menjadi MP.

Demikian seterusnya. Untuk keadaan yang lain maka cara yang digunakan sama dengan playfair cipher.

Untuk membandingkan algoritma double playfair cipher dengan secom dan VIC cipher kita akan menguji jumlah percobaan yang dibutuhkan untuk menemukan kombinasi kotak yang benar. Jumlah percobaan yang dibutuhkan pada double playfair akan lebih sedikit, karena kata kunci akan terdiri karakter yang unik (kalaupun tidak unik akan diabaikan duplikasinya). Hal ini berlaku jika kunci yang digunakan kurang dari 21 karakter (akan ada 20 ! kemungkinan kotak yang terbentuk).

Untuk mendapatkan kombinasi kotak yang lebih banyak dari jumlah kunci yang harus dicoba pada secom cipher dan VIC cipher (10^{20}) dibutuhkan 21 karakter untuk masing –masing kunci atau salah satu kunci minimal 22 karakter. Tentu saja jika kita menggunakan lebih dari 20 karakter untuk setiap kunci maka akan sangat menyulitkan dalam mengingatnya.

Kekurangan lain double playfair cipher dibandingkan dengan secom cipher adalah tidak tersedianya nilai angka dalam kotak yang digunakan untuk melakukan substitusi (hal ini sama dengan VIC cipher) sehingga dalam hal angka secom cipher memiliki keunggulan karena telah mengatasi penggunaan angka dalam pesan. Namun hal ini bisa diatasi dengan memperluas bujur sangkar yang digunakan, tidak hanya 5 x 5 lagi tapi bahkan bisa mencapai 16 x 16. hal in sangat dimungkinkan, apalagi dengan kemampuan komputasi saat ini. Dan jika ini dilakukan, maka double playfair cipher mampu mengenkripsi semua jenis dokumen dan menjadi lebih unggul dari secom dan VIC cipher, yang mana tidak bisa memperbesar straddling checker board melebihi 110 karakter.

Kesamaan secom cipher, VIC cipher, dan double playfair cipher adalah tidak adanya hubungan langsung antara plain text dan cipher text. Perbaikan dari playfair cipher ini dilakukan dengan mengubah cara membentuk pasangan abjad pada plain text yang akan dikodekan, sehingga tidak menjadi pasangan huruf yang berarti lagi. Namun begitu, double playfair cipher ini masih kalah dibandingkan secom dan VIC cipher karena pasangan huruf yang sama akan menghasilkan cipher text yang sama. Dalam secom dan VIC cipher, hubungan antar huruf maupun kata di plain text dengan cipher text benar-benar tidak terlihat lagi. Selain itu, kesamaan lain adalah kelemahan kedua algoritma dalam melakukan proses dekripsi terhadap plain text yang jumlah karakternya tidak sesuai harapan. Hasil dekripsi akan menghasilkan tambahan karakter ataupun keambiguan karakter (dalam double playfair cipher terjadi pada huruf I dan J). Meskipun hal ini dapat diatasi dengan menganalisis semantik kata yang ada, hal itu bisa berguna jika kita benar-benar melakukannya untuk kata yang bermakna, jika tidak maka kita akan mengalami kesulitan dalam menentukan kata apa yang sebenarnya terbentuk pada kata yang paling akhir.

6.2.2.2 Vigenere cipher

Jenis algoritma klasik lain yang juga mengandalkan kunci simetri adalah vigenere cipher. Algoritma ini tergolong sangat tua (dipublikasikan tahun 1856) dan sudah sangat sering dibahas dalam berbagai literatur tentang kriptography, jadi disini kita tidak membahasnya dengan sangat lengkap. Pada intinya vigenere cipher menggunakan bujur sangkar vigenere , dan menggunakan kunci yang diberikan untuk mengenkripsi plaintext dengan bantuan bujur sangkar vigenere.

Salah satu hal yang menjadikan vigenere menarik adalah kemampuan vigenere untuk membentuk cipher abjad-majemuk. Menurut data yang penulis dapat, pada saat vigener dipublikasikan, belum ada algoritma kunci simetri yang dapat melakukannya dengan baik.

Namun, vigenere cipher mempunyai beberapa titik lemah dalam keamannya.

Pada proses enkripsi, kunci yang diberikan akan diulang penggunaanya sebelum semua plain text dikodekan. Hal ini dilakukan sebagai akibat dari substitusi langsung tiap huruf plain text dengan huruf yang bersesuaian di bujur sangkar vigenere berdasar huruf pada kunci yang diberikan. Tentu saja pengulangan penggunan kunci bisa mengakibatkan pengulangan kriptogram dari plain text yang sama jika mendapat kunci yang sama. Hal inilah yang menjadi ide bagi Kasiski untuk menemukan panjang kunci, dan setelah panjang kunci diketahui maka usaha selanjutnya dapat dilakukan *bruteforce* ataupun memanfaatkan heuristic bahasa. Adanya hubungan dari blok cipher text dengan plain text (berupa panjang kata) menjadikan vigenere jauh lebih mudah dipecahkan dibanding secom cipher ataupun VIC cipher. Tentu saja secom dan VIC cipher juga diketahui panjang kuncinya, tapi pada vigenere kita dapat mengetahui beberapa kriptogram dengan panjang tertentu yang mempunyai plain text yang sama. Pada text yang sangat panjang, hal semakin mudah saja, karena akan ada lebih banyak kriptogram yang berulang.

Kelebihan yang ada pada vignere, sebagaimana pada double playfair cipher, adalah adanya kemungkinan untuk mengembangkan bujur sangkar vigenere menjadi berukuran 16 x 16, sehingga dapat digunakan untuk mengkodekan semua jenis karakter dalam byte yang mungkin.

7. Kesimpulan

Saat ini hampir semua algoritma klasik dengan kunci simetri telah dipecahkan, dan mempunyai berbagai kelemahan sehingga kurang relevan diterapkan pada saat ini. Hal ini terutama karena kemampuan komputer dapat direkayasa sehingga semakin cepat dalam melakukan percobaan kriptanalisis dengan metode *bruteforce*, sehingga algoritma kunci simetri modern (misalnya DEA) pun akan sulit bertahan karena kebergantungan pada kunci bisa dipecahkan dengan kekuatan komputasi dan peningkatan kemampuan hardware.

Meskipun algoritma kunci simetri tidak mungkin kita terapkan langsung untuk keamanan data saat ini, kita dapat menggunakan landasan berpikir dan konsep yang digunakan dalam menyusun algoritma klasik tadi untuk melakukan perbaikan pada algoritma kunci simetri modern yang lebih baik.

Algoritma klasik sebenarnya dapat dimodifikasi agar dapat beroperasi dalam mode byte, yaitu dengan memperbesar jangkauan kamus yang digunakan, dimana sebelumnya hanya menangani huruf maka diubah menjadi 256 byte. Kekuatan algoritma juga dapat ditambah dengan melakukan fungsi enkripsi berkali-kali dimana setiap fungsi akan menghasilkan nilai kunci yang berbeda yang akan digunakan pada proses selanjutnya. Cara lain adalah dengan menggabungkan beberapa cara yang pernah digunakan, misalnya substitusi, transposisi, operasi matematika, dan lain-lain.

Jadi, mempelajaari algoritma klasik sangatlah bermanfaat jika kita ingin memperdalam pengetahuan di bidang kriptography. Paling tidak kita bisa mengetahui algoritma apa saja yang pernah digunakan dan bagaimana melakukan serangan terhadap algoritma dengan fungsi-fungsi tertentu.

8. Daftar Pustaka

http://www.wikipedia.org/vic_cipher
http://www.informatika.org/~rinaldi
Munir, Rinaldi. 2006. Diktat Kuliah IF5054 kriptografi. Bandung.
http://www.pbs.org/wgbh/nova
http://users.telenet.be/d.rijmenants/index.htm
http://www.hypermaths.org/quadibloc/crypto/pp1324.html