STUDI DAN IMPELENTASI
CIPHER SUBSTITUSI RANTAI SEGITIGA

Mohamad Firda Fauzan – NIM : 13504127

Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail : if14127@students.if.itb.ac.id

Abstrak

Makalah ini membahas tentang cipher substitusi segitiga.Cipher ini merupakan hasil rekayasa dari cipher substitusi klasik, yaitu dengan metode rantai segitiga. Cipher ini memiliki dua kunci , yang pertama nilai integer yang berupa nilai pergeseran dari karakter yang berfungsi sama seperti pada caesar cipher yaitu jumlah pergeseran dari karakter semula , yang kedua merupakan kunci sebanyak n *byte* yang terdiri dari barisan integer yang akan dikalikan dengan dengan kunci pertama. Panjang n *byte* menyesuaikan dengan ukuran file yang akan disandikan. Fungsi dari perkalian nilai pergeseran dengan yang memiliki panjang n *byte* tersebut untuk menentukan pergeseran dari masing-masing karakter yang akan dienkripsi, sehingga besarnya pergeseran dari tiap-tiap karakter dalam suatu string yang akan dienkripsi akan berbeda.

Algoritma kriptografi klasik merupakan algoritma penyandian yang sudah ada sebelum zaman digital seperti sekarang ini. Algoritma klasik termasuk dalam sistem kriptografi simetri yaitu algoritma yang memiliki kunci yang sama dalam melakukan enkripsi dan dekripsi. Hal itu dikarenakan pada waktu itu kriptografi kunci publik belum ditemukan.

Algoritma klasik pada dasarnya hanya terdiri dari cipher substitusi dan cipher transposisi. Cipher substitusi adalah proses penyandian dengan mensubstitusi karakter-karakter yang ada pada plainteks. Sedangkan cipher transposisi adalah proses mempertukarkan huruf-huruf yang ada dalam suatu string.

Algoritma yang baik adalah algoritma yang tahan terhadap serangan. Kebaikan suatu algoritma dapat ditentukan dengan berapa banyak usaha yang dibutuhkan dalam memecahkan algoritma tersebut. Kriteria aman itu persamaan matematisnya rumit sehingga sulit dipecahkan dengan metode analitik, biaya untuk memecahkannya tinggi melebihi nilai informasinya, dan waktunya lama melebihi masa kadaluarsa informasi itu. Sehingga suatu algoritma dikatakan aman jika dapat memenuhi ketiga kriteria tersebut.

***Kata kunci**: Cipher*, substitusi ,rantai segitiga , enkripsi, dekripsi.

1. Pendahuluan

Dengan seiring berkembangnya kemajuan teknologi informasi dan jaringan komputer diseluruh dunia, maka peranan pengamanan komputer sangat penting. Banyak kejahatan cyber yang memanfaatkan celah keamanan yang ada dan melakukan manipulasi. Pada saat ini banyak sekali cara-cara yang ditempuh untuk memperkuat pengamanan, seperti password pada email, nomor PIN pada ATM, keamanan data/informasi rahasia.Usaha yang dilakukan untuk mengurangi kejahatan cyber yaitu dengan memperkuat sistem keamanan komputer, yaitu dengan cara membuat algoritma penyandian (cipher) yang lebih kuat terhadap serangan. Ukuran dari kekuatan chiper tersebut adalah banyaknya usaha yang diperlukan dalam melakukan pemecahan dari kunci cipher.

Pengiriman data dan penyimpanan data melalui media elektronik memerlukan suatu proses yang dapat menjamin keamanan dan keutuhan dari data yang dikirimkan tersebut. Data tersebut harus tetap rahasia selama pengiriman dan harus tetap utuh pada saat penerimaan di tujuan. Untuk memenuhi hal tersebut, dilakukan proses penyandian (enkripsi dan dekripsi) terhadap data yang akan dikirimkan. Enkripsi dilakukan pada saat pengiriman dengan cara mengubah data asli menjadi data rahasia sedangkan dekripsi dilakukan pada saat penerimaan dengan cara mengubah data rahasia menjadi data asli. Jadi data yang dikirimkan selama proses pengiriman adalah data rahasia, sehingga data asli tidak dapat diketahui oleh pihak yang tidak berkepentingan. Data asli hanya dapat diketahui oleh penerima dengan menggunakan kunci rahasia.

Algoritma penyandian sudah ada sejak zaman sebelum masehi dan memiliki sejarah yang panjang. Bangsa Mesir diketahui telah menggunakkan algoritma penyandian sejak 4000 tahun sebelum masehi, dengan adanya temuan berupa hyrogliph yang tidak standar. Bukti lain adalah di Sparta Yunani sekitar permulaan 400 tahun sebelum masehi, yaitu para tentara Sparta menggunakan penyandian dengan alat yang bernama *scytale*. Alat ini merupakan pita panjang pada daun payrus dengan sebatang silinder. Cara kerja alat ini yaitu pita panjang dililitkan pada silinder, kemudian diatas pipa itu ditulis pesan secara horizontal, kemudian pita itu dilepas dari silinder, sehingga huruf-huruf pada pita itu tersusun acak sehingga tidak dapat dibaca. Kemudian pita tersebut dikirim. Setelah sampai tempat penerima, cara membacanya yaitu dengan menggulung kembali pita tersebut ke silinder yang memiliki diameter yang sama sehingga pesan yang semula dapat terbaca sesuai dengan maksudnya. Jadi kunci dari penyandian ini adalah besarnya diameter silinder tempat menggulung.

Selain dengan cara mekanik seperti pada *scytale,* dahulu juga terdapat algoritma kriptografi yang kemudian dikenal sebagai algoritma kriptografi klasik. Algoritma kriptografi klasik pada dasarnya terdiri dari cipher substitusi dan cipher transposisi. Algoritma klasik melakukan penyandian karakter per karakter. Caesar Cipher adalah salah satu dari algoritma klasik. Caesar cipher mula-mula digunakan oleh kaisar Romawi, Julius Caesar untuk menyandikan pesan yang ia kirim kepada para gubernurnya. Cipher ini diimplementasikan dengan caesar wheel, caranya dengan mengganti setiap karakter dengan karakter lain dalam susunan abjad secara berurutan. Selain caesar cipher banyak lagi algoritma klasik seperti affine cipher yang merupakan penyempurnaan dari caesar cipher, hill cipher. Caesar cipher merupakan cipher yang sangat rentan terhadap serangan karena hanya mensubstitusi perkarakter dengan berurutan, sehingga jika diketahui satu karakter bersesuaian dengan huruf tertentu, maka dengan sangat mudah karakter lain pun terlihat. Untuk itu maka cipher substitusi pun berkembang, sehingga ditemukan substitusi homofonik, cipher substitusi abjad majemuk dan substitusi poligram. Banyak lagi jenis cipher yang lain seperti Vigenere cipher yaitu cipher abjad majemuk yang dikembangkan oleh Blaise de Vigenere pada abad ke-16. Cara kerja cipher ini yaitu dengan menggunakan bujur sangkar vigenere untuk melakukan enkripsi. Cipher ini memiliki kunci berupa string. Cipher ini memang lebih sulit dipecahkan daripada caesar cipher, tetapi pada abad ke-19 cipher ini dapat dipecahkan oleh kasiski, yaitu dengan menentukan panjang kunci dengan cara menghitung perulangan dari cipherteksnya.

Walaupun banyak kelemahan dari algoritma kriptograi klasik, namun kriptografi klasik dapat dijadikan sebagai sumber pemahaman konsep dasar kriptografi, dan dari kelemahan-kelemahan itulah didapat suatu algoritma baru yang lebih aman terhadap serangan-serangan yang ada.

2. Landasan Teori Kriptografi

Kriptografi pada awalnya merupakan ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya. Kemudian seiring dengan berkembangnya kriptografi yaitu kriptografi tidak lagi sebatas mengenkripsikan pesan, tetapi juga memberikan aspek keamanan yang lain seperti serangan dari kriptanalisis. Karena itu pengertian kriptografi pun berubah menjadi ilmu sekaligus seni untuk menjaga keamanan pesan.

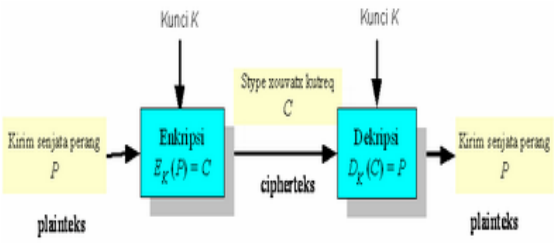
Sesuai dengan pengertian diatas kriptografi adalah cara untuk menyandikan pesan dari pengirim yang kemudian disebut plainteks menjadi pesan tersandikan (cipherteks) yang diterima oleh penerima pesan. Kemudian penerima pesan harus dapat mentransformasikan kembali pesan cipherteks tersebut menjadi plainteks.

Kekuatan dari kriptografi adalah tingkat kesulitan untuk mentransformasikan cipherteks menjadi plainteks. Kekuatan kriptografi dengan cara menjaga kerahasiaan algoritma yang digunakan disebut algoritma *restricted.* Tetapi algoritma restricted sudah tidak relevan lagi dipakai karena jika kerahasiaan algoritma sudah tidak terjaga lagi, maka perlu dibuat algoritma baru, sehingga sangat tidak efisien, karena diperlukan usaha yang besar untuk membuat algoritma kriptografi.

Untuk mengatasi masalah di atas saat ini kerahasiaan algoritma tidak lagi dilakukan. Sebagai gantinya yang dirahasiakan adalah kunci. Kunci adalah parameter yang digunakan untuk transformasi proses enkripsi (enciphering) dan proses dekripsi (deciphering). Kunci biasanya berupa string atau deretan bilangan. Dengan menggunakan kunci K, untuk mengenkripsikan plainteks P dengan fungsi E_k, di peroleh cipherteks C, begitu pula sebaliknya untuk proses dekripsi, sehingga fungsi enkripsi dan dekripsi menjadi :

$E_k(P) = C$
$D_k(C) = P$

Dan kedua fungsi itu memenuhi
$D_k(E_k(P)) = P$



Gambar 1 Skema Enkripsi dan Dekripsi dengan dengan parameter kunci

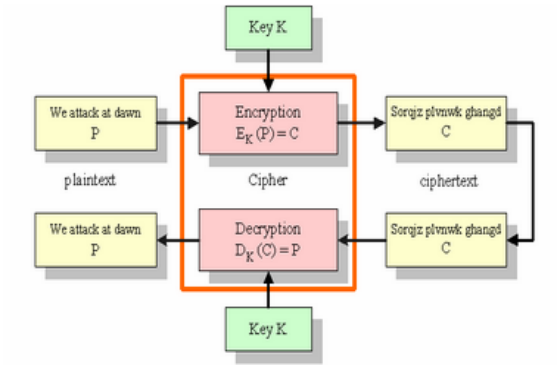
2.1 Aspek Keamanan yang disediakan Kriptografi

Kriptografi selain menyandikan pesan juga menyediakan beberapa aspek keamanan. Berikut aspek keamanan kriptografi :

1. Kerahasiaan (confidentiality), adalah layanan yang digunakan untuk menjaga isi pesan dari siapapun yang tidak berhak membacanya. Layanan ini direalisasikan dengan cara menyandikan pesan menjadi bentuk yang tidak dapat dimengerti. Misalnya pesan "Harap datang pukul 8" disandikan menjadi "TrxC#45motypetre!%".
2. Integritas data (data integrity), adalah layanan yang menjamin bahwa pesan masih asli / utuh atau belum pernah dimanipulasi selama pengiriman. Layanan ini direalisasikan dengan menggunakan tanda-tanda digital (digital signature). Pesan yang telah ditandatangani menyiratkan bahwa pesan yang dikirim adalah asli.
3. Otentifikasi (authenification), adalah layanan yang berhubungan dengan identifikasi, baik mengidentifikasi kebenaran pihak-pihak yang berkomunikasi (user authentification atau entity authentification) maupun mengidentifikasi kebenaran sumber pesan (data origin authentification). Layanan ini direalisasikan dengan menggunakan digital signature.
4. Nirpenyangkalan (non-repudiation), adalah layanan untuk mencegah entitas yang berkomunikasi melakukan penyangkalan, yaitu pengirim pesan menyangkal melakukan pengiriman atau penerima pesan menyangkal telah menerima pesan.

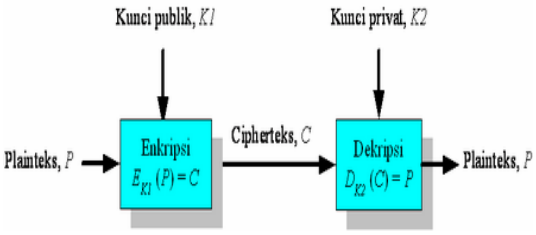
2.2 Jenis Kunci Kriptografi

Kunci kriptografi terbagi menjadi dua jenis, yaitu kriptografi kunci simetri (symetric key cryptography) dan kriptografi kunci asimetri (asymetric-key sryptography).
Kriptografi kunci simetri yaitu jika kunci enkripsi sama dengan kunci dekripsi. Algoritma nya disebut algoritma simetri atau algoritma konvensional.
Contoh algoritma kunci simetri adalah algoritma DES (Data Encryption Standard), Rijndael, Blowfish, IDEA, GOST, serpent, RC2, RC4, RC5 dan algoritma pada cipher klasik.



Gambar 2 Skema kriptografi simetri. Kunci enkripsi sama dengan kunci dekripsi, yaitu K.

Kriptografi kunci asimetri yaitu sistem kriptografi menggunakan kunci yang berbeda untuk enkripsi dan dekripsi. Misal kunci untuk enkripsi adalah K1 dan kunci untuk dekripsi adalah K2, yang dalam hal ini K1 ≠ K2. Kriptografi kunci asimetri disebut juga kriptografi kunci-publik. Algoritma kriptografinya disebut algoritma asimetri atau algoritma kunci publik.
Contoh algoritma asimetri adalah RSA (Rivest-Shamir-Adleman)



Gambar 3 Skema kriptografi asimetri. Kunci enkripsi tidak sama dengan kunci dekripsi. Kunci enkripsi bersifat publik (tidak rahasia), sedangkan kunci dekripsi bersifat private (rahasia)

2.3 Serangan Terhadap Kriptografi

Serangan (attack) adalah setiap usaha atau percobaan yang dilakukan oleh kriptanalis untuk menemukan kunci atau menemukan plainteks dari cipherteksnya.

Prinsip kerckhoff menyatakan ”semua algoritma kriptografi harus publik, hanya kunci yang rahasia”

Dari prinsip kerckhoff dapat ditarik kesimpulan bahwa setiap serangan terhadap kriptografi diasumsikan kriptanalis mengetahui algoritma kriptografi yang digunakan, sehingga hanya kunnci yang menjadi sistem keamanan dikriptografi.

Berdasarkan keterlibatan penyerangnya dalam dikomunikasi, maka serangan dapat dibagi atas dua macam, yaitu :

1. Serangan pasif (passive attack) yaitu penyerang tidak terlibat dalam komunikasi antara pengirim dan penerima, namun penyerang menyadap semua pertukaran pesan antara pengirim dan penerima. Tujuannya yaitu untuk mendapatkan sebanyak mungkin informasi yang digunakan untuk kriptanalisis.
2. Serangan aktif (active attack) yaitu penyerang mengintervensi komunikasi dan ikut mempengaruhi sistem untuk keuntungan penyerang.

Berdasarkan ketersediaan data yang ada, serangan terhadap kriptografi dapat dibedakan menjadi enam kategori :

1. Chipertext-only attack
   Kriptanalis memiliki beberapa chiperteks dari beberapa pesan, semuanya dienkripsi dengan algoritma yang sama. Tugas kriptanalis adalah menemukan plainteks atau kunci.
2. Known-plainteks attack
   Kriptanalis melihat kata-kata yang sering dipakai pada umumnya misalnya ”dengan hormat” pada surat, kemudian menerka kuncinya dengan cara membandingkan cipherteksnya terhadap kata yang sering muncul.
3. Chosen-plainteks attack
   Kriptanalis memilih plainteks yang mengarah kepada kunci, lalu plainteks tersebut dienkripsi, kemudian

cipherteksnya dicocokkan untuk mendapatkan kunci.

4. Adaptive-chosen-plainteks attack
   Kasus khusus untuk chosen-plainteks attack. Yaitu dengan melakukannya secara berantai.
5. Chosen-chiperteks attack
   Kebalikan dari chosen-plainteks attack, kriptanalis memilih cipherteks untuk didekripsikan menjadi plainteks lalu dicocokkan.
6. Chosen-text attack
   Gabungan teknik dari chosen-plainteks dengan chosen-cipherteks.

Berdasarkan teknik yang digunakan dalam menemukan kunci. Serangan dapat dibagi menjadi:

1. Exhaustive attack atau brute force attack yaitu kriptanalis mencoba semua kemungkinan kunci (trial and error).
2. Analytical attack, yaitu kriptanalis menghitung frekuensi kemunculan huruf-huruf pada cipherteks yang kemudian dicocokkan dengan frekuensi kemunculan huruf pada plainteks. Sehingga memberikan keterkaitan antara huruf-huruf tersebut.

2.4 Algoritma Kriptografi

Algoritma kriptografi (cipher) adalah aturan untuk enciphering dan deciphering atau fungsi matematika yang digunakan untuk enkripsi dan dekripsi.

Berdasarkan sejarah algoritma kriptografi dapat dibedakan menjadi algoritma kriptografi klasik dan algoritma kriptografi modern.

Algoritma kriptografi klasik yaitu algoritma kriptografi sebelum masuk era digital, kriptografi yang dilakukan berbasis karakter. Algoritma kriptografi klasik termasuk dalam sistem kriptografi simetri, karena kunci untuk melakukan enkripsi sama dengan kunci untuk melakukan enkripsi.

Algoritma kriptografi modern merupakan algoritma yang berkembang setelah berkembangnya era digital. Operasi yang digunakan umumnya dalam mode bit. Sehingga semua sistem yang terlibat di dalam nya seperti kunci, plainteks, dan cipherteks semuanya dinyatakan dalam rangkaian bit-bit biner,0 dan 1.

Algoritma kriptografi klasik tetap masih penting peranannya karena algoritma kriptografi klasik merupakan landasan dasar algoritma kriptografi modern, seperti operasi substitusi dan transposisi, hanya saja algoritma modern berada pada operasi bit per bit, bukan operasi karakter.

2.4.1 Algoritma Kriptografi Substitusi

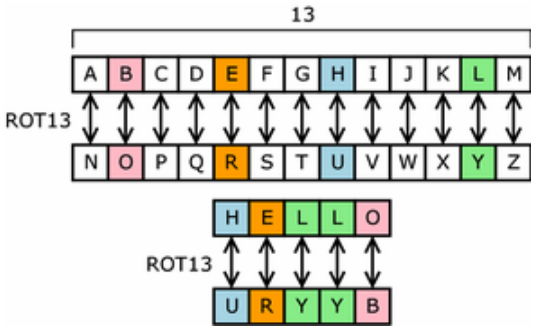
Algoritma kriptografi (cipher) substitusi merupakan cipher tertua. Prinsip utama cipher substitusi adalah menukarkan setiap huruf pada plainteks dengan sesuatu.

Cipher substitusi dapat dikelompokkan lagi menjadi :

1. Cipher substitusi abjad tunggal (monoalphabetic substitution cipher)
2. Cipher substitusi homofonik (Homophonic substitution cipher)
3. Cipher abjad majemuk (Polyalphabetic substitution cipher)
4. Cipher substitusi polygram (Polygram substitution cipher)

2.4.1.1 Cipher Substitusi Abjad Tunggal

Cipher substitusi abjad tunggal adalah cipher dengan mensubstitusi satu karakter pada plainteks dengan satu karakter yang bersesuaian. Jadi fungsi ciphering nya adalah fungsi satu ke satu.



Gambar 4 Pemetaan huruf-huruf plainteks dengan huruf-huruf cipherteks dalam ROT13

Teknik mensubstitusi abjad tunggal pun berbagai macam diantaranya ada yang secara langsung mengganti huruf yang satu dengan yang lain secara acak, cara lain yaitu dengan membuat kata kunci pada awal cipherteks kemudian sisa dari karakter yang tidak terdapat pada kata kunci tersebut. Untuk memperjelas cara kedua, dapt melihat contoh berikut :

[WIK06] Misal menggunakan kata kunci ”zebras” sehingga substitusi pada alfabetnya menjadi :

Plaintext alphabet:	`abcdefghijklmnopqrstuvwxyz`
Ciphertext alphabet:	`ZEBRASCDFGHIJKLMNOPQTUVWXY`

Sehingga pesan berikut :

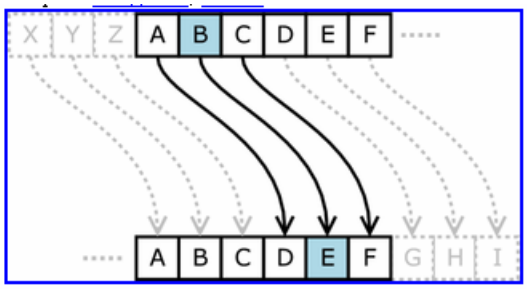
flee at once. we are discovered!

Dienkripsi menjadi

```
SIAA ZQ LKBA. VA ZOA RFPBLUAOAR!
```

Meskipun jumlah kemungkinan kunci pada substitusi abjad tunggal seperti diatas sangat besar yaitu $26! \approx 22^{88.4}$, namun cipher tersebut sangat mudah diserang Ciphertext-only attack yaitu dengan teknik analisis frekuensi pada cipherteksnya.

Caesar cipher adalah kasus khusus dari substitusi abjad tunggal dimana susunan huruf cipherteks diperoleh dengan menggeser huruf-huruf alfabet. Proses transformasi enciphering dan deciphering nya dengan mensubstitusi satu karakter dengan karakter lain pada susunan alfabet secara berurutan dan pergeserannya pun sama untuk semua karakter dalam alfabet. Misalnya suatu string digeser sebanyak 3 karakter. Maka karakter pertama menjadi 3 karakter dibawahnya, karakter kedua juga sama dan seterusnya, sampai semua string terenkripsi semuanya.



Gambar 5 Proses transformasi pada enchipering atau deciphering caesar cipher

Misal karakter A akan disubstitusi dengan D, karakter B akan disubstitusi dengan E, karakter C akan disubstitusi dengan F, dan karakter berikutnya.

Plaintext letter	A	B	C	D	.....	W	X	Y	Z
Ciphertext letter	D	E	F	G	.....	Z	A	B	C

Gambar 6 Tabel substitusi caesar cipher

Contoh [DAV 02] :
Misalnya string ”FIRE MISSILE” dienkripsi dengan caesar cipher sesuai dengan tabel caesar cipher pada gambar 4.

Maka hasil enkripsinya adalah

```
”ILUV PLVVLOH”
```

Dalam praktek, biasanya string hasil enkripsi akan dikelompokkan menjadi blok-blok string atau dengan menyatukan semua string sehingga menjadi string yang tanpa spasi.

Maka untuk contoh di atas jika dikelompokkan menjadi 3 karakter tiap blok menjadi

```
”ILU VPL VVL OH”
```

Atau jika tanpa spasi menjadi

```
”ILUVPLVVLOH”
```

Untuk mempermudah pengkodeaan, tiap-tiap karakter dipasangkan dengan nilai integer 0 sampai 25, seperti ditunjukkan pada gambar 5.

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
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Gambar 7 Tabel kode alfabet

Sehingga dengan tabel tersebut , secara matematis caesar cipher menyandikkann plainteks p_i menjadi c_i dengan aturan :

$$c_i = E(p_i) = p_i + (k \bmod n)$$

dan dekripsi cipherteks c_i menjadi p_i dengan aturan

$$p_i = D(c_i) = c_i - (k \bmod n)$$

dengan
p_i : plainteks ke-i
c_i : cipherteks ke-i
k : kunci

n : banyaknya range karakter.

Nilai n untuk enchipering dan deciphering hanya untuk huruf alfabet maka nilai n 26, sedangkan untuk enciphering dan deciphering dengan semua karakter ASCII maka nilai n adalah 256.

Kelemahan caesar cipher adalah :

1. Ciphertext only attack : Caesar cipher mudah diserang dengan hanya melihat cipherteksnya. Metode yang digunakan yaitu dengan analsisis frekuensi. Karena dengan melihat frekuensi cipherteks terbanyak disesuaikan dengan banyaknya frekuensi pada plainteks pada umumnya.
2. Exhaustive key search : Serangan ini mudah karena kunci dalam caesar cipher hanya 25 kemungkinan kunci untuk alfabet dan 255 kemungkinan kunci untuk ASCII.

2.4.1.2 Cipher Substitusi Abjad Majemuk

Cipher substitusi abjad majemuk merupakan cipher substitusi ganda (multiple substitution cipher) yang melibatkan penggunaan kunci berbeda. Cipher abjad majemuk dibuat dari sejumlah cipher abjad tunggal, masing-masing dengan kunci yang berbeda.

Pada umumnya cipher abjad majemuk adalah cipher substitusi periodik yang didasarkan pada periode m.

Misalkan plainteks P adalah

$P = p_1p_2....p_{m-1}p_m...$

Maka cipherteks nya adalah

$Ek(P) = f_1(p_1)f_2(p_2)...f_{m-1}(p_{m-1})f_m(p_m)...$

Dalam hal ini p_i merupakan karakter pada huruf-huruf didalam plainteks. Untuk kasus m = 1, substitusi ini sama dengan substitusi abjad tunggal.

Substitusi abjad majemuk yang paling populer adalah vigenere cipher yang ditemukan oleh Blaise de Vigenere yang dipublikasi pada tahun 1583 dan baru terpecahkan pada 1863. Pada Vigenere cipher berlaku untuk setiap P plainteks dan cipherteks C dimana

$P = p_1p_2p_3....p_{m-1}p_{m.....}p_{2m}$
$C = c_1c_2c_3....c_{m-1}c_{m.....}c_{2m}$

Dan K adalah kunci dengan panjang m dimana

$K = k_1k_2...k_m$

Untuk $1 \leq i \leq m$ dan panjang n adalah jumlah alfabet yang digunakan maka berlaku hubungan :

$c = (p + k_i) \bmod n$

kunci k akan berulang setelah periode m. Untuk lebih jelasnya lihat contoh berikut :

[WIK06] Misalnya enkripsi dengan kunci SPACE yang panjang nya = 5.

Plainteks	: DANGER WILL ROBIN SON.
Kunci	: SPACES PACE SPACE SPA
Chiperteks	: VPNIIJ LINP JDBKR KDN

Chiperteks diatas diperoleh sesuai dengan persamaan diatas, D = 3 dan S=19 maka chiper teksnya
c = (4 + 19) mod 26 = 22 , maka diperoleh V.

Keamanan Vigenere cipher melebihi keamanan pada substitusi tunggal, karena pada Vigenere cipher teknik analisis frekuensi tidak dapat diterapkan. Tetapi Vigenere Cipher mudah diserang jika sudah diketahu panjang kuncinya dan panjang kunci tersebut tidak terlalu panjang. Karena dengan exhaustive search dapat ditemukan.

Akhirnya Vigenere cipher terpecahkan setelah kasiski menemukan cara mencari panjang kunci. Panjang kunci ditemukan dengan cara menghitung semua jarak antara cipherteks yang berulang, kemudian jarak tersebut dicari faktor pembagi dari jarak tersebut. Sehingga ditemukan panjang kunci untuk cipherteks tersebut.

Contoh [WIK06]

DYDUXRMHTVDV**NQD**QNW**DYDUXRMH**ARTJGW**NQD**

Kriptogram yang berulang adalah **DYUDUXRM** dan **NQD**. Jarak antara dua buah perulangan **DYUDUXRM** adalah 18. Semua faktor pembagi 18 adalah {18, 9, 6, 3, 2}

Jarak antara dua buah perulangan **NQD** adalah 20. Semua faktor pembagi 20 adalah {20,10,5, 4, 2}.

Irisan dari kedua buah himpunan tersebut adalah 2, sehingga panjang kunci kemungkinan besar adalah 2.

2.4.1.3 One Time Pad Cipher

One Time Pad cipher adalah salah satu algoritma kriptografi yang tidak terpecahkan. One Time Pad (OTP) ditemukan pada tahu 1917 oleh Major Joseph Mauborgone. Cipher ini termasuk algoritma kriptografi kunci simetri.

One Time Pad merupakan algoritma kriptografi yang memiliki kunci berupa deretan-deretan karakter yang dibangkitkan secara acak. Kunci pada OTP hanya digunakan sekali saja untuk mengenkripsi pesan yang kemudian dipakai lagi untuk mendekripsi pesan itu. Setelah selesai maka kunci tersebut dihancurkan.

Aturan enkripsi OTP sama seperti pada cipher substitusi berabjad majemuk, yaitu untuk proses enkripsi :

$c_i = (p_i + k_i) \bmod 26$

sedangkan untuk dekripsi :

$p_i = (c_i - k_i) \bmod 26$

Contoh[MUN06] :

Plainteks : `ONETIMEPAD`
Kunci : `TBFRGFARFM`

Nyatakan A = 0 , B = 1, ..., z = 25 maka cipherteksnya adalah : `HOJKOREGHP`

Sistem OTP tidak dapat dipecahkan karena[MUN06] :

1. Barisan kunci acak yang ditambahkan ke pesan plainteks yang tidak acak menghasilkan cipherteks yang seluruhnya acak.
2. Beberapa barisan kunci yang digunakan untuk mendekripsikan cipherteks mungkin menghasilkan pesan-pesan plainteks yang mempunyai makna, sehingga kriptanalis tidak punya cara untuk menentukan plainteks mana yang benar.

Contoh[MUN06] :
Cipherteks `HOJKOREGHP`, misal kriptanalis mencoba kunci `LMCCAWAAZD` maka akan menghasilkan plainteks `SALMONEGGS`, kemudian kriptanalis juga mencoba kunci `ZDVUZOEYEO` maka akan menghasilkan plainteks `GREENFIELD`.

Kedua plainteks itu mempunyai makna, sehingga membingungkan kriptanalis

Meskipun OTP pada teori merupakan cipher yang tidak dapat dipecahkan, tetapi pada prakteknya cipher ini jarang dipakai karena masalah kepraktisan. Kunci OTP sangat panjang sehingga sulit dalam penyampaian, selain itu juga karena panjang kunci tersebut maka pengirim dan penerima tidak mungkin membangkitkan kunci secara simultan sehingga dalam transfer informasinya membutuhkan waktu yang lama.

3. Chiper Substitusi Rantai Segitiga

3.1 Latar belakang substitusi rantai segitiga

Algoritma klasik khususnya algoritma substitusi memiliki kekurangan dalam segi keamanan. Algoritma substitusi mudah untuk diserang. Sebagai contoh algoritma substitusi caesar cipher yang sangat mudah diserang dengan ciphertext only attack, yaitu dengan melihat ciherteks nya kemudian dengan analisis frekuensi sehingga dengan mudah dapat ditemukan kuncinya. Selain itu juga mudah diserang dengan exhaustive search, karena hanya memiliki sedikit kemungkinan kunci.

Selain caesar cipher algoritma substitusi abjad tunggal pun dapat diserang dengan teknik analisis frekuensi. Untuk cipher substitusi lainnya seperti cipher substitusi homofonik memang lebih baik dari pada cipher substitusi abjad tunggal, karena teknik analisis frekuensi tidak dapat lagi digunakan, hal itu disebabkan tidak ada lagi keterkaitan antara frekuensi pada plainteks dan cipherteks. Namun cipher substitusi homofonik ini juga dapat dipecahkan dengan serangan known-plainteks attack, yaitu dengan menerka pada kata-kata yang umum dipakai. Misal string ”dengan hormat” pada awal kalimat surat resmi.

Untuk algoritma cipher abjad majemuk yang merupakan sejumlah substitusi abjad tunggal yang dibuat dengan kunci yang berbeda memang lebih aman. Tetapi keberadaan kunci yang berulang-ulang menyebabkan cipherteks hasil enkripsinya juga terdapat pengulangan-pengulangan karakter. Contoh terkenal dari

substitusi abjad majemuk adalah vigenere cipher. Vigenere cipher memang mengatasi serangan dengan teknik analisis frekuensi dan known-plainteks karena setiap huruf yang sama dapat dienkripsikan menjadi huruf yang berbeda. Tetapi jika panjang kunci sudah diketahui dan tidak terlalu panjang maka dengan exhaustive search pencarian kunci dapat dilakukan.

One Time Pad merupakan algoritma klasik yang tidak dapat dipecahkan. Hal itu dikarenakan panjang kunci enkripsi memiliki panjang sama dengan jumlah karakter yang akan dienkripsikan. One Time Pad memiliki kelemahan panjang kunci yang terlalu panjang, tetapi selain kelemahan hal itu juga merupakan kelebihannya. Kelemahan dari One Time Pad ini menyebabkan One Time Pad sulit untuk didistribusikan.

Melihat dari sebab-sebab yang dijelaskan diatas maka sesungguhnya masih ada algoritma klasik yang tidak dapat dipecahkan seperti One Time Pad, tetapi tidak cocok lagi digunakan pada masa sekarang, sehingga yang menjadi permasalahan saat ini adalah bagaimana membuat kunci tersebut menjadi ringkas. Berdasarkan atas teori syarat bagaimana cara agar suatu cipher tidak dapat dipecahkan yaitu kunci harus acak dan dan panjang kunci harus sama dengan panjang plainteks maka muncul ide membuat algoritma kriptografi rantai segitiga.

3.2 Algoritma rantai segititga

Algoritma kriptografi rantai segitiga merupakan algoritma yang dibuat guna memperbaiki algoritma kriptografi klasik khususnya algoritma substitusi abjad tunggal yang sangat mudah diserang dengan teknik analisis frekuensi.

Algoritma kriptografi rantai segitiga merupakan cipher yang ide awalnya dari algoritma kriptografi One Time Pad, yaitu kunci yang dibangkitkan secara random dan panjang kunci sepanjang plainteks yang akan dienkripsi. Tetapi pada algoritma kriptogarfi rantai segitiga pembangkitan kunci-kunci tersebut secara otomatis dengan teknik berantai.

Algoritma rantai segitiga ini memiliki aturan substitusi berdasar pada caesar cipher yaitu dengan pergeseran huruf-huruf. Kekuatan cipher ini terletak pada kunci yaitu nilai integer yang menunjukkan pergeseran karakter-karakter sesuai dengan operasi pada caesar cipher. Kekuatan kedua terletak pada barisan bilangan-bilangan yang berfungsi sebagai pengali dengan kunci. Barisan bilangan tersebut dapat berupa bilangan tertentu seperti deret bilangan ganjil, deret bilangan genap, deret fibonaci, deret bilangan prima, serta deret bilangan yang dapat dibuat sendiri.

Berikut gambaran sederhana cipher rantai segitiga:

Misal plainteks yang akan dienkripsi adalah "ABCDEFG" dengan kunci 3 dan deret bilangan pengali adalah bilangan asli.

```
Plainteks      :     ABCDEFG
1 X 3                DEFGHIJ
2 X 3                 KLMNOP
3 X 3                  UVWXY
4 X 3                   HIJK
5 X 3                    XYZ
6 X 3                     QR
7 X 3                      M
```

Maka cipherteks yang dihasilkan **DKUHXQM**

Contoh diatas merupakan gambaran sederhana cipher rantai segitiga yang beroperasi pada alfabet 26 karakter. Angka disebelah kiri deret karakter merupakan nilai perseseran. Nilai pergeseran berfungsi untuk menggeser huruf-huruf yang mekanismenya sama seperti pada caesar cipher, yaitu untuk operasi pada alfabet 26 karakter , sehingga pada proses enkripsi tiap barisnya berlaku :

$$c_i = E(p_i) = (p_i + (k * R[baris])) \bmod 26$$

c_i adalah karakter ke-i cipherteks sedangkan adalah karakter ke-i plainteks. R adalah tabel yang berisi pola faktor pengali. Maka untuk bilangan asli R[1]=1, R[2] = 2, R[3]= 3, dan seterusnya. Jika faktor pengali bilangan ganjil maka R[1]=1, R[2]=3, R[3]= 5, dan seterusnya. Jika faktor pengali adalah bilangan dengan pola menyesuaikan (costumize), maka nilai tabel R pun berisi nilai-nilai tersebut.

Untuk kasus di atas maka untuk baris ke-1 yaitu A = 0, B = 1 , C = 2 , dan seterusnya sesuai dengan gambar 7 maka :

$$C_1 = (0 + (3*1)) \bmod 26$$
$$= 3 \text{ (D)}$$

$$C_2 = (1 + (3*1)) \bmod 26$$
$$= 4 \text{ (E)}$$

C_3 =(2 +(3*1)) mod 26
= 5 (F)

C_4 ... C_7 mekanismenya sejenis seperti 3 contoh diatas.

Mekanisme tersebut berlaku pula untuk baris kedua, ketiga dan seterusnya hanya saja faktor pengali mengikuti pola yang terdapat pada tabel R. Maka pada baris 2 dengan R[2], baris 3 dengan R[3] dan seterusnya. Selain itu untuk setiap baris bertambah maka karakter yang paling awal tidak dienkripsikan kembali.

Untuk dekripsi setiap barisnya juga berlaku hubungan kebalikan dari enkripsi yaitu :

$p_i=D(c_i)=(c_i-(k*R[baris]))mod\ 26$

Sekilas jika kita melihat contoh sederhana di atas maka cipher rantai segitiga ini berkesan sangat sederhana dan mudah diserang. Plainteks **ABCDEFG** menghasilkan cipherteks **DKUHXQM**. Maka kriptanalis dapat dengan mudah menentukan kunci, karena karakter pertama hanya dienkripsi sekali, kedua dua kali, ketiga tiga kali dan seterusnya. Sehingga untuk kasus diatas A(0) bersesuaian dengan D(3), dengan enkripsi sekali maka dapat ditentukan :

3 = (0 + X) mod 26 (1)

Untuk karakter kedua B(1) bersesuaian dengan K(10) dengan enkripsi dua kali maka :

10 = (Hasil enkripsi pertama + Y) mod 26
10 = ((1 + X)mod 26) + Y) mod 26 (2)

Dari persamaan 1 kriptanalis dapat mencari nilai X yaitu 3 dimana X adalah kunci * R [1].
Dengan memanfaatkan persamaan 1 maka persamaan 2 menjadi :
10 = ((1+3)mod 26)+Y) mod 26
10 = (4 + Y) mod 26
Maka dengan mudah didapat Y=6
Untuk kasus ke 3 dan seterusnya caranya juga sama sehingga dengan mudah diketahui keteraturannya.

Contoh cipher rantai segitiga diatas memang mudah diserang, karena contoh tersebut memang dibuat untuk menjelaskan bagaimana mekanisme dari cipher rantai segitiga bekerja.

Pada kenyataannya cipher substitusi segitiga tidak dibuat sesederhana itu, tetapi dengan mengenkripsi ganda yaitu mengenkripsi dua kali, jadi plainteks dienkripsi dengan cipher segitiga I, kemudian hasil enkripsi pertama dienkripsi kembali dengan cipher segitiga II yang arah segitiga II merupakan kebalikan arah segitiga I. Berikut contoh sederhana cipher segitiga ganda yang beroperasi pada alfabet 26 karakter :

Misal plainteks yang akan dienkripsi adalah "ABCD" dengan kunci 3 dan deret bilangan pengali adalah bilangan asli.

Plainteks : ABCD

Enkripsi pertama

1 X 3 **D**EFG
2 X 3 **K**LM
3 X 3 **U**V
4 X 3 **H**

Cipherteks yang dihasilkan enkripsi pertama **DKUH.** Cipherteks hasil enkripsi pertama kembali dienkripsi dengan arah segitiga berkebalikan.

Enkripsi kedua
Plainteks : DKUH

1 X 3 GNX**K**
2 X 3 MT**D**
3 X 3 V**C**
4 X 3 **H**

Maka cipherteks yang merupakan hasil akhir yaitu **HCDK.**

Pada contoh di atas mekanisme enkripsi pada enkripsi pertama sama dengan contoh sebelumnya yaitu verlaku hubungan

$c_i=E(p_i)=(p_i+(k*R[baris]))mod\ 26$

Sehingga plainteks **ABCD** dienkripsi dengan kunci 3 dan faktor pengali berupa deret bilangan asli diperoleh cipherteks **DKUH**.

Untuk enkripsi kedua juga berlaku mekanisme yang sama seperti enkripsi pertama hanya faktor pengali pun menjadi berulang dari awal lagi, dan karakter berikutnya yang tidak dienkripsi untuk setiap pertambahan baris adalah karakter terakhir.

Dengan mengenkripsi dua kali, maka cipher substitusi rantai segitiga ini tidak dapat ditembus dengan known-plainteks attack. Karena sulit mencari hubungan antara **HCDK** dengan **ABCD.**

Untuk itu maka standar untuk cipher segitiga ini adalah cipher segitiga ganda yaitu cipher rantai segitiga yang melakukan enkripsi ganda dengan pola seperti contoh diatas, yaitu dengan membuat pola enkripsi pertama dengan mengerucut ke arah kanan dan enkripsi kedua mengerucut ke arah kiri.

Secara matematis pola enkripsi rantai segitiga dapat digambarkan dengan matriks N x N dengan N merupakan panjang plainteks yang akan dienkripsi dan operasi pada alfabet ASCII.

Matriks dilambangkan dengan $M_{ij,}$ dengan $1 \leq i \leq N$ dan $1 \leq j \leq N$.
Nilai integer kunci dengan K.
Faktor pengali merupakan tabel integer R.
Plainteks dengan P dimana P merupakan tabel plainteks dengan panjang N yaitu P[N].

Berikut operasi matriks untuk proses pengenkripsian :

1. Matriks enkripsi segitiga pertama

   Untuk baris ke-1

   $M_{1j} = P[j] + (K*R[1])) \bmod 256$

   Untuk baris ke 2 dan selanjutnya untuk nilai $j \geq i$:

   $M_{ij} = M_{(i-1)j} + (K*R[i])) \bmod 256$

2. Matriks enkripsi segitiga kedua

   Nilai P diperoleh dari nilai M_{ij} pada i = j
   Untuk baris ke-1

   $M_{1j} = P[j] + (K*R[1])) \bmod 256$

   Untuk baris ke 2 dan selanjutnya untuk nilai $j \leq (N+1) - i$:

   $M_{ij} = M_{(i-1)j} + (K*R[i])) \bmod 256$

   Sehingga nilai cipherteks yang diperoleh adalah :

   M_{ij} pada nilai $j = (N+1)-i$

Sedangkan untuk proses dekripsi merupakan kebalikan dari proses enkripsi.

Berikut operasi matriks untuk proses dekripsi :

1. Matriks dekripsi segitiga pertama
   Operasinya merupakan kebalikan dari matriks enkripsi, jadi operasi ini kebalikan operasi matriks enkripsi segitiga kedua.

   Nilai P merupakan tabel dari cipherteks dengan panjang N yaitu C[N].

   Untuk baris terakhir kolom pertama

   $M_{ij} = C[j];$

   Untuk baris dan kolom lainnya untuk nilai $j \leq (N+1)-i$

   $M_{ij} = (M_{(i+1)j} - (K*R[i+1]) \bmod 256$

2. Matriks dekripsi segitiga kedua

   Nilai C diperoleh dari nilai M_{0j} untuk nilai $1 \leq j \leq N$.

   Untuk baris terakhir kolom terakhir untuk nilai $j > i$

   $M_{ij} = C[i]$

   Untuk baris dan kolom lainnya

   $M_{ij} = (M_{(i+1)j} - (K*R[i+1])) \bmod 256$

   Maka nilai plainteksnya adalah pada :
   M_{oj} untuk nilai $1 \leq j \leq N$.

3.3 Kekuatan Algoritma Rantai Segititga

Algoritma rantai segitiga yang merupakan algoritma kriptografi simetri melakukan proses enkripsi dan dekripsi dengan abjad tunggal. Algoritma kriptografi rantai segitiga telah memperhatikan unsur *confusion* dan *diffusion* yang artinya membingungkan dan menyebar. Selain itu cipher rantai segitiga juga telah mempertimbangkan unsur pemilihan kunci-kunci acak seperti pada One Time Pad, sehingga menghasilkan cipherteks yang acak pula. Serta panjang kunci yang memiliki panjang sama dengan plainteksnya. Selain hal tersebut di atas cipher rantai segitiga memiliki operasi yang sederhana yaitu dengan sistem operasi pada caesar cipher yaitu hanya dengan menggeser karakter-karakter yang akan dienkripsikan.

Untuk mempermudah pendeskripsian berikut contoh algoritma rantai segitiga.

Contoh :

Misal sebuah plainteks "ABCDEFG" akan dienkripsi dengan kunci 3 secara ganda dan deret bilangan pengali adalah bilangan asli beroperasi pada alfabet 26 karakter.

Plainteks	:	ABCDEFG
1 X 3		**D**EFGHIJ
2 X 3		**K**LMNOP
3 X 3		**U**VWXY
4 X 3		**H**IJK
5 X 3		**X**YZ
6 X 3		**Q**R
7 X 3		**M**

Plainteks	:	DKUHXQM
1 X 3		GNXKAT**P**
2 X 3		MTDQG**Z**
3 X 3		VCMZ**P**
4 X 3		HOY**L**
5 X 3		WD**N**
6 X 3		O**V**
7 X 3		**J**

Maka cipherteks yang dihasilkan **JVNLPZP**

Berikut uraian tentang kekuatan algoritma rantai segitiga :

1. Sesuai dengan prinsip membingungkan (confussion) dan menyebar (diffussion). Algoritma kriptografi rantai segitiga memiliki cipherteks yang tidak ada hubungannnya dengan plainteks, sehingga kriptanalis bingung dalam memecahkan cipher ini. Pada contoh di atas cipherteks yang dihasilkan dari plainteks "**ABCDEFG**" adalah "**JVNLPZP**". Tidak ada yang dapat ditarik hubungan antara kedua string tersebut, begitu pula untuk frekuensi kemunculan antara keduanya, sehinga hal ini membingungkan kriptanalis. Algoritma kriptografi rantai segitiga juga mengadopsi prinsip menyebar. Karena proses enkripsi dan dekripsi dilakukan berulang-ulang. Hasil dari proses pertama akan menjadi dasar untuk proses kedua, hasil proses kedua akan menjadi dasar proses ketiga dan berlaku untuk proses selanjutnya. Maka seandainya cipherteks mengalami perubahan sedikit, maka secara otomatis beberapa bagian yang bersesuaian akan mengalami kekacauan atau salah arti dalam proses dekripsi.
2. Kunci-kunci yang dibangkitkan secara acak sehingga membentuk cipherteks yang acak.
   Cipher ini memiliki kunci berupa bilangan integer. Kunci tersebut tidak dapat berjalan dengan sendirinya, kunci tersebut butuh faktor pengali. Faktor pengali itu dapat berupa deretan bilangan asli, deret bilangan ganjil, deret bilangan genap, deret bilangan prima dan deret deret lain, atau pun faktor pengali tersebut dapat menyesuaikan sesuai dengan kemauan pengenkripsi. Hasil dari perkalian kunci dengan faktor pengali menghasilkan kunci pergeseran yang nilainya acak, sehingga menghasilkan cipherteks yang acak pula. Pada contoh di atas kunci yang digunakan adalah nilai integer 3 dan faktor pengali nya adalah deret bilangan asli. Karena kunci tersebut dibangkitkan secara acak maka cipherteksnya pun acak, dan tidak ada hubungannya dengan plainteks.
3. Panjang kunci sama dengan plainteks yang akan dienkripsi.
   Karena dalam cipher rantai segitiga nilai kunci dikalikan faktor pengali, dalam hal ini faktor pengali tersebut merupakan tabel bilangan dengan jumlah sepanjang plainteks yang akan didekripsikan. Maka secara otomatis nilai ini sama saja dengan panjang kunci sama dengan panjang plainteks. Untuk faktor pengali dengan deret bilangan maka deret tersebut akan mengisi tabel bilangan sepanjang plainteks, sedangkan untuk faktor pengali yang menyesuaikan dengan masukkan dari kemauan pengenkripsi, maka tabel bilangan tersebut di isi dengan angka-angka acak tersebut secara sekuensial sesuai dengan urutan penulisan, sehingga jika jumlah bilangan random tersebut kurang dari panjang plainteks maka bilangan-bilangan tersebut ditulis berulang sesuai dengan periode jumlah bilangan acak tersebut. Contoh faktor pengali sesuai dengan kemauan pengenkripsi :
   Misal plainteks dengan panjang N = 10, dengan bilangan pengali acak 3, 8, 19, 7, 4, 97. Maka nilai tabelnya R[1]=2,

R[2]=8, R[3]=19, R[4]=7, R[5]=4, R[6]=97, R[7]=3, R[8]=8, R[9]=19, R[10]=7.

Dari uraian tentang kekuatan algoritma kriptografi rantai segitiga, sehingga menyebabkan hal-hal sebagai berikut :

1. Ciphertext-only attack dengan teknik Analisis frekuensi tidak lagi dapat digunakan.
   Cipherteks hasil enkripsi sudah tidak memiliki frekuensi yang bersesuaian dengan plainteks. Sehingga teknik analisis frekuensi tidak cocok lagi digunakan untuk menyerang algoritma ini.
2. Known-plainteks attack sulit untuk dilakukan.
   Dengan enkripsi ganda maka sulit dicari keterhubungan antara tiap-tiap karakter yang bersesuaian antara cipherteks dengan plainteks, sehingga sulit untuk melakukan serangan dengan known-plainteks attack.
3. Kunci yang panjang seperti pada One Time Pad.
   Dengan menggunakan faktor pengali yang pada prakteknya menggunakan tabel sepanjang panjang plainteks, maka dapat dihasilkan kunci yang panjangnya sepanjang plainteks seperti pada OTP sehingga dapat menghasilkan cipherteks yang acak.
4. Mempermudah dalam pendistribusian kunci.
   Karena untuk membuat kunci yang panjang seperti pada OTP sebenarnya dibutuhkan pendistribusian yang sulit, maka dengan rantai segitiga panjang kunci pun hanya dengan memilih deret bilangan atau bilangan acak menyesuaikan dan di implementasikan di tabel pengali seperti sudah dijelaskan pada bagian sebelumnya sehingga hal itu cukup untuk membuat perulangan pada karakter cipherteks terjadi seperti pada vigenere cipher.

3.4 Kelemahan Algoritma Rantai Segititga

Kriptografi rantai segitiga mempunyai kekurangan :

1. Kekuatan selain terdapat pada kunci juga terdapat pada faktor pengali. Faktor pengali yang mempunyai kekuatan terbaik adalah faktor pengali yang memiliki tingkat kerandoman yang sangat tinggi, sehingga untuk membuat faktor pengali yang random tersebut dibutuhkan suatu usaha yang lebih besar, dan kekurangpraktisan dalam prakteknya.
2. Rantai segitiga dalam mengenkripsi dilakukan secara berantai, sehingga enkripsi karakter yang satu mempengaruhi semua, sehingga cipherteks harus dijaga keasliannya. Jika ada perubahan sedikit pada cipherteks maka akan menghasilkan plainteks yang tidak bermakna.
3. Kebutuhan ruang dan waktu yang besar. Karena dalam cipher ini jika suatu plainteks mengandung `N` karakter, maka dibutuhkan ruang sebesar `N X N` untuk proses enchipering dan deciphering. Untuk kebutuhan waktu dapat dihitung berdasarkan jumlah substitusi karakter-karakter dalam proses enciphering dan deciphering. Jika plainteks N karakter maka untuk baris ke 1 dalam matriks N X N dibutuhkan sejumlah N substitusi. Untuk baris kedua dibutuhkan N-1 substitusi. Sehingga dapat ditemukan persamaan :

   Jumlah substitusi baris ke-n : `N-(n-1)`

   Sehingga jumlah semua substitusi semua : `(1+N) . (N/2)`

   Sehingga untuk cipher segitiga ganda menjadi `(1+N) . N.`

   Jika satu operasi substitusi membutuhkan waktu 1 milidetik maka dibutuhkan (N^2 + N) milidetik.

4. Pengujian

4.1 Perancangan Kasus Uji Pengujian Algoritma Rantai Segitiga

Berdasarkan tata ancang dan teknik pengujian yang telah dijelaskan, maka dirancang kasus-kasus uji sebagai berikut:

1. Kasus Uji 1
   Kasus Uji 1 bertujuan untuk mengetahui mekanisme dan menguji kebenaran enkripsi dan dekripsi sederhana dengan

enkripsi tunggal pada alphabet 26 karakter.

2. Kasus Uji 2
   Kasus Uji 2 bertujuan untuk mengetahui mekanisme dan menguji kebenaran proses enkripsi dan dekripsi dengan enkripsi dan dekripsi ganda pada alphabet 26 karakter .
3. Kasus Uji 3
   Kasus Uji 3 bertujuan untuk mengetahui mekanisme dan menguji kebenaran proses enkripsi dan dekripsi dengan enkripsi dan dekripsi ganda pada alphabet ASCII.
4. Kasus Uji 4
   Kasus Uji 4 bertujuan untuk membandingkan proses enkripsi dan dekripsi tunggal dan ganda pada alphabet 26 karakter.
5. Kasus Uji 5
   Kasus Uji 5 bertujuan untuk membandingkan proses enkripsi dan dekripsi ganda dengan faktor pengali pada pilihan pada alphabet 26 karakter dengan alphabet ASCII.
6. Kasus Uji 6
   Kasus Uji 6 bertujuan untuk membandingkan proses enkripsi dan dekripsi ganda dengan faktor pengali yang menyesuaikan pada alphabet 26 karakter dengan alphabet ASCII.
7. Kasus Uji 7
   Kasus Uji 7 bertujuan untuk menguji tingkat keamanan data algoritma kriptografi rantai segitiga tunggal dengan alphabet 26 karakter.
8. Kasus Uji 8
   Kasus Uji 8 bertujuan untuk menguji tingkat keamanan data algoritma kriptografi rantai segitiga ganda pada alphabet 26 karakter.
9. Kasus Uji 9
   Kasus Uji 9 bertujuan untuk menguji tingkat keamanan data algoritma kriptografi rantai segitiga ganda dengan faktor pengali sesuai dengan pilihan pada alphabet ASCII.
10. Kasus Uji 10
    Kasus Uji 10 bertujuan untuk menguji tingkat keamanan data algoritma kriptografi rantai segitiga ganda dengan faktor pengali yang menyesuaikan pada alphabet ASCII.
11. Kasus Uji 11
    Kasus uji 11 bertujuan untuk membandingkan keamanan algoritma rantai segitiga dengan operasi alfabet 26 karakter dengan alfabet ASCII.
12. Kasus Uji 12
    Kasus uji 12 bertujuan untuk membandingkan kebutuhan ruang dan waktu antara substitusi rantai segitiga dengan substitusi abjad tunggal.

4.2 Evaluasi Hasil Pengujian Algoritma Rantai Segitiga

Hasil pengujian ini menggunakan dua buah program yaitu program untuk alfabet 26 karakter yang selanjutnya disebut program segitiga I dan program untuk alfabet ASCII yang selanjutnya disebut program segitiga II. Semua kasus uji adalah berkas teks berekstensi .txt. Kasus uji yang menggunakan program segitiga I yaitu kasus uji 1,2,6,7,8 dan 11. Sedangkan kasus uji untuk program segitiga II yaitu kasus uji 3,6,9 dan 10. Untuk kasus uji dengan program segitiga I berkas hanya mengandung 26 huruf kecil karakter alfabet, sedangkan untuk kasus uji dengan program segitiga II, berkas boleh mengandung semua karakter ASCII.

1. Hasil uji kasus uji 1 dan kasus uji 2
   Pengujian terhadap plainteks "`abcdefg`" dengan kunci 3 dan faktor pengali bilangan asli. Diperoleh hasil "`dkuhxqm`" pada enkripsi tunggal dan hasil "`jvnlpzp`". Maka hasil tersebut sesuai dengan uraian proses kerja cipher substitusi segitiga pada bagian sebelumnya.
2. Hasil kasus uji 3
   Hasil enkripsi merupakan semua karakter ASCII dan dapat dikembalikan lagi (dekripsi) seperti semula.
3. Hasil kasus uji 4.
   Hasil uji enkripsi tunggal dari plainteks "abcdefg" menghasilkan cipherteks "dkuhxqm" dan enkripsi ganda menghasilkan cipherteks "jvnlpzp". Pada enkripsi tunggal keterkaitan karakter-karakter masih dengan mudah dapat ditemukan dengan serangan known plainteks attack, sedangkan untuk hasil enkripsi ganda sulit untuk mencari hubungan antara plainteks dengan cipherteks.
4. Hasil kasus uji 5
   Hasil pada enkripsi ganda pada alfabet 26 menghasilkan cipher yang juga terdiri dari 26 karakter sedangkan untuk ASCII terdiri dari 256 karakter ASCII, shingga dari segi keamanan karakter ASCII lebih aman karena mencari jumlah kemungkinan menjadi lebih besar daripada 26 karakter.
5. Hasil kasus uji 8,9,10 dan 11

Tingkat keamanan pada cipher substitusi ganda lebih baik jika dibandingkan dengan caesar cipher dan substitusi abjad tunggal. Contoh pada kasus uji 8 pada plainteks huruf yang paling banyak muncul dari yang terbanyak berturut-turut e, r, n, a, i. Sedangkan pada cipherteksnya hampir memiliki kemunculan huruf yang rata untuk semua huruf. Berikut tabel perbandingan karakter plainteks dan cipherteks pada file kasus uji 8.

Kar	Plain	Cip	Kar	Plain	Cip
a	67	36	n	71	24
b	8	23	o	52	35
c	36	22	p	50	32
d	34	14	q	1	32
e	87	28	r	73	40
f	4	23	s	28	31
g	19	42	t	50	37
h	29	24	u	16	36
i	54	32	v	8	34
j	2	28	w	13	43
k	9	49	x	0	27
l	40	22	y	28	12
m	18	40	z	2	43

Untuk kasus uji 9 dengan faktor pengali sesuai dengan pilihan pada program ini deret bilangan ganjil atau genap (bisa juga dengan deret bilangan lain), maka cipherteks selain tidak dapat diserang dengan analisis frekuensi juga tidak ditemui adanya pasangan huruf berulang seperi pada vigenere cipher, karena dengan faktor pengali tersebut, panjang kunci menjadi sepanjang plainteks, sehingga sama seperti OTP. Tetapi dengan pemilihan faktor pengali seperti tersebut jika kriptanalisis berhasil menemukan faktor pengali maka mudah juga untuk menemukan kunci.

Kasus uji 10 merupakan penggunaan faktor pengali yang menyesuaikan (costum). Dengan menggunakan faktor pengali seperti ini maka keteraturan faktor pengali seperti pada uji kasus 9 dapat teratasi. Tetapi tipe faktor pengali seperti ini merepotkan, baik pada waktu proses enciphering dan deciphering, maupun distribusi kunci. Meskipun demikian panjang faktor pengali ini tidak harus sepanjang plainteks seperti pada OTP.

6. Hasil kasus uji 13

Kebutuhan ruang untuk cipher substitusi rantai segitiga memang lebih besar dibandingkan dengan cipher substitusi. Untuk cipher substitusi jika plainteks terdiri dari N karakter, maka hanya dibutuhkan ruang sebanyak N. Tetapi untuk rantai segitiga dibutuhkan ruang berukuran N X N yang pada pelaksanaan nya akan diisi sebanyak (1+N) . N/2. Untuk substitusi ganda maka diperlukan ukuran dua kali substitusi tunggal yaitu (N^2 + N). Sedangkan kebutuhan waktu berbanding lurus dengan jumlah substitusi yang dilakukan. Untuk substitusi abjad tunggal, untuk plainteks dengan panjang N, hanya dibutuhkan N operasi substitusi, sedangkan pada cipher segitiga dibutuhkan (1+N) . N/2 untuk segitiga abjad tunggal dan (N^2 + N) untuk substitusi abjad majemuk.

5. Kesimpulan

Kesimpulan yang dapat diambil dari studi dan implementasi cipher rantai segitiga ini adalah :

1. Kriptografi algoritma klasik memiliki operasi dasar berupa operasi substitusi yaitu mengganti huruf yang akan disandikan menjadi huruf tertentu dan operasi substitusi yaitu mempertukarkan huruf-huruf yang akan disandikan dengan huruf-huruf itu sendiri, sehingga keteraturan huruf-huruf tersebut menjadi acak.
2. Cipher substitusi rantai segitiga merupakan cipher yang prinsip dasarnya adalah caesar cipher dengan cara melakukan substitusi berulang kali serta berantai sehingga membentuk suatu cipherteks yang acak.
3. Standar minimal agar cipher rantai segitiga dikatakan aman adalah dengan melakukan enkripsi ganda sehingga cipher rantai segitiga yang standar adalah cipher rantai segitiga ganda.
4. Cipher rantai segitiga memiliki tingkat keamanan yang lebih baik dari caesar cipher dan substitusi abjad tunggal. Cipher ini tahan dari serangan known-plainteks attack, cipherteks-only attack dan analisis frekuensi.
5. Cipher rantai segitiga memiliki kebutuhan ruang dan waktu yang lebih besar dari cipher abjad tunggal, hal ini disebabkan operasi substitusi pada cipher rantai segitiga terjadi berulang-ulang sedangkan pada substitusi abjad tunggal hanya satu kali operasi substitusi. Dengan demikian kebutuhan waktu cipher substitusi rantai segitiga membutuhkan waktu yang lebih banyak dibanding substitusi abjad tunggal.

DAFTAR PUSTAKA

[1] Munir, Rinaldi. (2006). Diktat Kuliah IF5054 Kriptografi. Program Studi Teknik Informatika Sekolah Teknik Elektro dan Informatika , Institut Teknologi Bandung.

[2] Bishop, David. (2002). Introduction to Cryptography with Java Aplets. Jones and Batrlet Publisher.

[3] Douglas J.Riesenberg . Question about classic chryptography (Double Encryption with Affine cipher, Non-relative Prime Affine Decryption, Hill Cipher Decryption, Up the Hill Down the Hill, First-Timer LSFR Sequence) EC 575 Data Security and Cryptography

[4] *Substitution CipherEnsiklopedi http://eng..wikipedia.org*. Tanggal akses: 20 September 2006 pukul 15.00.