Kopel Dav Vinci

Ilden Abi Neri [1)]

1) Jurusan Teknik Informatika Sekolah Teknik Elektro dan Informatika ITB, Bandung, email: if14145@students.if.itb.ac.id

Abstract – *Kopel Dav Vinci merupakan salah satu algoritma kriptografi klasik dimana dalam proses enkripsi dan dekripsinya memiliki dua tahap utama. Pertama, memanfaatkan teknik variasi vigenere cipher dan kedua, mengaplikasikan persamaan linear dalam proses enkripsi dan dekripsi. Pada akhirnya, algoritma ini berhasil mematahkan serangan terhadap cipherteks dengan memanfaatkan teknik kasiski. Algoritma ini memiliki peluang untuk terus dikembangkan terutama dengan memperluas ruang lingkup karakter dari plainteks yang akan dienkripsi.*

Kata Kunci*: Kopel Dav Vinci, kriptografi klasik enkripsi, dekripsi,, vigenere cipher, persamaan linear.*

1. PENDAHULUAN

Kopel Dav Vinci adalah singkatan dari kombinasi persamaan linear dan variasi *vigenere cipher*. Dari singkatannya, dapat kita simpulkan bahwa algoritma ini menggunakan persamaan linear dan bentuk lain dari *vigenere cipher* dalam melakukan proses enkripsi dan dekripsi data. Pada awalnya ide ini muncul karena adanya pemikiran dari penulis untuk memberikan salah satu solusi agar algorima *vigenere cipher* yang sudah dapat dipecahkan dengan teknik kasiski, bisa dimodifikasi sedemikian rupa sehingga teknik kasiski tidak dapat lagi digunakan untuk memecahkan algoritma hasil modifikasi tersebut. Akhirnya diperolehlah ide variasi *vigenere* tersebut tapi, karena penulis menilai terlalu sederhana maka penulis mengkombinasikannya dengan memasukkan persamaan linear kedalam algoritma ini.

Sehingga dalam melakukan proses enkripsi dan dekripsi dengan menggunakan algoritma ini, terdapat dua tahap yang harus dilalui yakni pertama, dengan menggunakan kombinasi *vigenere* cipher kemudian yang kedua, dengan menerapkan persamaan linear. Untuk lebih lengkap dan jelasnya akan dijelaskan pada bagian selanjutnya.

2. PENJELASAN KOPEL DAV VINCI

Untuk menjelaskan algoritma ini, penulis membagi menjadi dua tahap yang pertama, tahap enkripsi dan yang kedua, tahap dekripsi.

2.1. Tahap Enkripsi

Dalam proses enkripsi terdapat dua langkah yang harus dilakukan. Pertama, adalah dengan menerapkan variasi *vigenere cipher* dan yang kedua, menerapkan persamaan linear. Sebagai penyederhaan, karakter yang akan dienkripsi dibatasi hanya pada abjad A sampai Z dimana A diberi label dengan 0, B dengan 1 dan seterusnya sampai Z diberi label 25. Untuk lebih jelasnya, penulis memberikan contoh sebagai berikut.

Plainteks : ABCDEABCDE
Kunci : ILDEN

Plainteks	A	B	C	D	E	A	B	C	D	E
(label)	0	1	2	3	4	0	1	2	3	4
Rantai Kunci	I	L	D	E	N	I	L	D	E	N
(label)	8	11	3	4	13	8	11	3	4	13

Langkah I adalah dengan mengekripsi dengan variasi *vigenere* cipher. Berikut ilustrasinya:

Plainteks	A	B	C	D	E	A	B	C	D	E
Rantai Kunci	I	L	D	E	N	I	L	D	E	N
Chiperteks ke - 0 = (0 + (8 mod 26)) mod 26 = 8 ➔ I										
Plainteks	A	B	C	D	E	A	B	C	D	E
Rantai Kunci	I	L	D	E	N	I	L	D	E	N
Chiperteks ke - 1 = (1 + ((8+11) mod 26)) mod 26 = (1 + 19) mod 26 = 20 ➔ U										
Plainteks	A	B	C	D	E	A	B	C	D	E
Rantai Kunci	I	L	D	E	N	I	L	D	E	N
Chiperteks ke - 2 = (2 + ((8+11+3) mod 26)) mod 26 = (2 + 22) mod 26 = 24 ➔ Y										
	A	B	C	D	E	A	B	C	D	E
	I	L	D	E	N	I	L	D	E	N
Chiperteks ke - 3 = (3 + ((8+11+3+4) mod 26)) mod 26 = (3 + 0) mod 26 = 3 ➔ D										

Jika proses enkripsi pada langkah pertama diteruskan terhadap semua plainteks, maka didapat cipherteks untuk langkah pertama sebagai berikut :

IUYDRVHLQE

I	U	Y	D	R	V	H	L	Q	E
8	20	24	3	17	21	7	11	16	4

Untuk langkah pertama ini kita dapat menurunkan rumusnya sebagai berikut :

C'[i] = (P[i] + (K[0] + … + K[i] mod 26)) mod 26 … persamaan (1)

Keterangan "persamaan (1)" :
C'[i] : Label huruf cipherteks tahap I yang ke-i
P[i] : Label huruf plainteks yang ke-i
K[i] : Label huruf rantai kunci yang ke-i

Langkah II adalah dengan menerapkan persamaan linear dalam proses enkripsi.

Bentuk umum dari persamaan linear yang diaplikasikan kedalam proses enkripsi adalah sebagai berikut :

C"[i] = ((pk . K[i]) mod 26 + C'[i]) mod 26 … persamaan (2)

Keterangan "persamaan (2)" :
C"[i] : Label huruf cipherteks tahap II yang ke-i
pk : Panjang kunci
K[i] : Label huruf rantai kunci yang ke-i
C'[i] : Label huruf cipherteks tahap I yang ke-i

Bentuk penerapan "persamaan (2)" terhadap contoh adalah sebagai berikut : (panjang kunci = 5 → ILDEN)

C"[0] = ((5 . 8) mod 26 + 8) mod 26
= (14 + 8) mod 26
= 22 mod 26
= 22 → W

C"[1] = ((5 . 11) mod 26 + 20) mod 26
= (3 + 20) mod 26
= 23 mod 26
= 23 → X

C"[2] = ((5 . 3) mod 26 + 24) mod 26
= (15 + 24) mod 26
= 39 mod 26
= 13 → N

C"[3] = ((5 . 4) mod 26 + 3) mod 26
= (20 + 3) mod 26
= 23 mod 26
= 23 → X

Jika proses enkripsi pada langkah kedua diteruskan terhadap semua cipherteks hasil langkah pertama, maka didapat cipherteks untuk langkah kedua sebagai berikut :

WXNXEJKAKR

W	X	N	X	E	J	K	A	K	R
22	23	13	23	4	9	10	0	10	17

2.2. Tahap Dekripsi

Untuk melakukan proses dekripsi, sederhananya kita cukup membalikkan proses enkripsi saja dimana terlebih dahulu dimulai dengan membalikkan persamaan linear untuk langkah pertama kemudian membalikkan variasi *vigenere cipher* untuk langkah kedua.

Langkah I adalah membalikkan persamaan linear pada proses enkripsi langkah kedua dengan rumusan sebagai berikut :

C'[i] = ((C"[i] + 26) – ((pk . K[i]) mod 26)) mod 26 … persamaan (3)

Keterangan untuk "persamaan (3)" sama dengan "persamaan (2)".

Berikut aplikasinya terhadap contoh sebelumnya :

C'[0] = ((22 + 26) – ((5 . 8) mod 26)) mod 26
= (48 – 14) mod 26
= 34 mod 26
= 8 → I

C'[1] = ((23 + 26) – ((5 . 11) mod 26)) mod 26
= (49 – 3) mod 26
= 46 mod 26
= 20 → U

C'[2] = ((13 + 26) – ((5 . 3) mod 26)) mod 26
= (39 – 15) mod 26
= 24 mod 26
= 24 → Y

C'[3] = ((23 + 26) – ((5 . 4) mod 26)) mod 26
= (49 – 20) mod 26
= 29 mod 26
= 3 → D

Jika proses dekripsi langkah pertama ini diteruskan terhadap semua cipherteks hasil proses enkripsi langkah kedua, maka akan didapat cipherteks yang persis sama dengan cipherteks hasil enkripsi langkah pertama, yakni :

IUYDRVHLQE

Langkah II adalah membalikkan variasi *vigenere cipher* pada proses enkripsi langkah pertama dengan rumusan sebagai berikut :

P[i] = ((C'[i] + 26) – ((K[0]+ ... + K[i]) mod 26)

) mod 26 … persamaan (4)

Keterangan untuk "persamaan (4)" sama dengan "persamaan (1)".

Berikut aplikasinya terhadap hasil yang diperoleh dari proses dekripsi langkah pertama :

P [0] = ((8 + 26) – ((8) mod 26)) mod 26
= (34 – 8) mod 26
= 26 mod 26
= 0 → A

P [1] = ((20 + 26) – ((8 + 11) mod 26)) mod 26
= (46 – 19) mod 26
= 27 mod 26
= 1 → B

P [2] = ((24 + 26) – ((8 + 11 + 3) mod 26)) mod 26
= (50 – 22) mod 26
= 28 mod 26
= 2 → C

P [2] = ((3 + 26) – ((8 + 11 + 3 + 4) mod 26)) mod 26
= (29 – 0) mod 26
= 29 mod 26
= 3 → D

Jika proses dekripsi langkah kedua ini diteruskan terhadap semua teks hasil proses dekripsi langkah pertama, maka akan didapat plainteks yang persis sama dengan plainteks semula, yakni :

ABCDEABCDE

Dengan demikian proses enkripsi dan dekripsi telah selesai dilakukan untuk contoh kasus diatas.

Untuk lebih mempersingkat proses enkripsi dan dekripsi, dapat dilakukan dengan menjadikan proses enkripsi dan dekripsi yang semula dilakukan masing-masing dalam dua langkah menjadi satu langkah. Hal ini dimungkinkan dengan mengubah rumusan proses enkripsi dengan mensubtitusikan "persamaan (1)" kedalam "persamaan (2)", menjadi sebagai berikut :

C"[i] = ((pk . K[i]) mod 26 + ((P[i] + (K[0] + … + K[i] mod 26)) mod 26)) mod 26 … persamaan (5)

dan untuk proses dekripsi rumusannya diubah dengan mensubtitusikan "persamaan (3)" kedalam "persamaan (4)" menjadi sebagai berikut :

P[i] = (((((C"[i] + 26) – ((pk . K[i]) mod 26)) mod 26) + 26) – ((K[0]+ ... + K[i]) mod 26)) mod 26 … persamaan (6)

3. ANALISIS TERHADAP KOPEL DAV VINCI

Seperti yang telah dijelaskan sedikit sebelumnya, Kopel Dav Vinci pada awalnya ditujukan untuk memperbaharui algoritma *vigenere cipher* sedemikian sehingga tidak dapat dipecahkan dengan teknik kasiski.

Menurut contoh :
Plainteks : ABCDEABCDE
Cipherteks (enkripsi dengan algoritma Kopel Dav Vinci) : WXNXEJKAKR

Dari contoh dapat kita tarik beberapa kesimpulan bahwa :

1. Huruf plainteks yang sama belum tentu pada cipherteksnya juga sama.

   P[0] = P[5] = A
   C[0] = W, C[5] = J → C[0] /= C[5]

2. Huruf plainteks yang berbeda mungkin saja dienkripsi menjadi huruf yang sama pada cipherteksnya.

   P[6] = B, P[8] = D → P[6] /= P[8]
   C[6] = C[8] = K

3. Blok huruf pada plainteks yang berulang dengan indeks posisi yang bersesuaian dengan indeks posisi blok huruf kunci pada rantai kunci, belum tentu menghasilkan blok huruf pada cipherteks yang berulang pula.

   Blok I : ABCDE → WXNXE
   Blok II : ABCDE → JKAKR

4. Keamanan dari cipherteks sangat bergantung pada kunci. Baik itu huruf-huruf penyusun kunci maupun panjang dari kunci.

Dengan adanya poin pertama dan kedua, analisis statistik terhadap hasil enkripsi atau cipherteks jelas tidak akan dapat membantu dalam memecahkan cipherteks yang ada sedikitpun. Hal ini jelas sangat berperan sekali dalam hal menjadikan cipherteks menjadi lebih aman dari serangan kriptanalis.

Berikutnya dengan poin ketiga, penyerangan dengan teknik kasiski tidak akan berpengaruh terhadap keamanan dari cipherteks. Tentunya hal ini sesuai dengan tujuan awal dari algoritma ini dibuat.

Sementara itu, poin keempat memberikan pengaruh ganda bagi keamanan dari cipherteks. Maksudnya, disatu sisi dengan kebergantungan keamanan algoritma terpusat pada kunci yang digunakan merupakan sesuatu hal yang baik tapi, disisi lain hal ini bisa menjadi kelemahan yang cukup berarti apabila

panjang kunci diketahui. Karena dengan panjang kunci maka pemecahan terhadap cipherteks akan menjadi mudah dilakukan oleh kriptanalis, seperti yang telah dijelaskan diatas panjang kunci menjadi salah satu komponen yang digunakan untuk melakukan proses enkripsi dan dekripsi.

4. PELUANG PENGEMBANGAN KOPEL DAV VINCI

Dalam makalah ini, penulis melakukan penyederhanaan agar penjelasan algoritma menjadi lebih mudah. Penulis menilai terdapat beberapa peluang untuk pengembangan algoritma ini lebih jauh. Diantaranya adalah sebagai berikut :

1. Memperluas cakupan karakter plainteks yang akan dienkripsi. Misalnya, dengan memasukkan angka, karakter-karakter khusus dan sebagainya. Dan tentunya hal ini akan memberikan pengaruh terhadap fungsi modulo yang dipakai dimana besaran yang dipakai tergantung kepada jumlah karakter yang digunakan.

2. Plainteks yang akan dienkripsi bisa berupa barisan blok-blok bit 0, 1 layaknya algoritma kriptografi modern. Berikut ilustrasinya :

   Diketahui :
   Panjang kunci = 6
   Panjang blok bit = 4.
   Plainteks : 0001 0101 0111 0101
   Rantai kunci : 1110 0100 1011 0100

   Misal proses enkripsi untuk blok kedua plainteks :

Plainteks	0001	0101	0111	0101
Rantai Kunci	1110	0100	1011	0100

   Enkripsi blok bit kedua plainteks caranya adalah :

   Langkah I
   Blok bit kedua plainteks di-XOR kan dengan blok bit kedua rantai kunci kemudian hasilnya di XOR kan lagi dengan blok bit pertama rantai kunci dan hasilnya adalah 1111

   Langkah II
   Hasil enkripsi langkah pertama dalam bilangan desimal adalah 15 dan blok bit kedua rantai kunci dalam desimal adalah 4.

   Hasil enkripsi akhir blok kedua = ((4 . 6 mod 16) + 15) mod 16 = (8 + 15) mod 16 = 7 → 0111. Di modulo 16 karena panjang blok bit = 4.

5. KESIMPULAN

Algoritma Kopel Dav Vinci atau kombinasi persamaan linear dan variasi *vigenere cipher* adalah salah satu contoh algoritma yang penulis kembangkan untuk memodifikasi algoritma *vigenere cipher* sedemikian sehingga tidak dapat dipecahkan dengan teknik kasiski. Setelah pengembangan selesai dilakukan, penulis meyakini tujuan awal ini telah tercapai sekaligus juga aman dari jenis serangan klasik lainnya seperti analisis statistik.

Untuk teknik atau jenis serangan lain terhadap cipherteks hasil enkripsi dengan algoritma Kopel Dav Vinci, penulis tidak menjelaskan di makalah ini.

Secara keseluruhan dapat dilihat bahwa algoritma ini tidak menggunakan fungsi-fungsi yang rumit dalam proses enkripsi ataupun dekripsi sehingga algoritma ini sangat mudah diimplementasikan.

Untuk kategori, Kopel Dav Vinci termasuk kedalam algoritma kriptografi klasik karena karakter pada plainteks yang dienkripsi disederhanakan khusus untuk abjad A sampai Z. Tetapi, algoritma Kopel Dav Vinci memiliki peluang untuk terus dikembangkan menjadi lebih baik dari yang ada saat sekarang ini.

Bagi siapapun yang ingin atau tertarik untuk mempelajari algoritma ini kemudian menemukan kelemahan-kelemahan didalamnya, memperbaiki, mengembangkan lebih jauh maka, penulis akan berterima kasih sekali dan harapan penulis semoga makalah ini bisa memberikan manfaat dan kontribusi walaupun sangat sedikit terutama bagi perkembangan ilmu kriptografi kedepannya.

Akhir kata penulis mengucapkan puji syukur kepada Allah SWT dan terima kasih kepada semua pihak baik secara langsung maupun tidak langsung sehingga makalah ini bisa diselesaikan.

DAFTAR REFERENSI

[1] Munir, Rinaldi, “Diktat Kuliah IF5054 Kriptografi”, Institut Teknologi Bandung, 2006.
[2] Piper Fred & Sean Murphy, “Cryptography: A Very Short Introduction”, Oxford, 2002.