Studi Mengenai Algoritma Skipjack dan Penerapannya

M. Auriga Herdinantio – NIM : 13506056

Program Studi Teknik Informatika, Institut Teknologi Bandung

Jl. Ganesha 10, Bandung

E-mail : if16056@students.if.itb.ac.id

Abstrak

Salah satu hal yang penting untuk menjamin kerahasiaan data dan informasi adalah enkripsi. Enkripsi menggunakan suatu algoritma yang dapat mengkodekan semua aliran data (*stream*) bit dari sebuah pesan menjadi kriptogram yang tidak dimengerti (*unintelligible*).

Skipjack merupakan algoritma elektronik codebook 64-bit yang merubah 64-bit blok masukan menjadi 64-bit blok keluaran. Parameter yang digunakan untuk enkripsi adalah 80-bit kunci, dan mempunyai 32 putaran untuk proses enkripsi dan dekripsi. Algoritma ini dapat digunakan pada salah satu dari empat modus ditetapkan dalam FIPS 81 untuk digunakan dalam Data Encryption Standard (DES).

Algoritma Skipjack dikembangkan oleh Badan Keamanan Nasional Amerika Serikat dan karena tidak banyak yang mengetahui algoritma Skipjack ini, algoritma ini digolongkan rahasia oleh pemerintah Amerika Serikat. Algoritma ini cocok untuk melindungi semua tingkat klasifikasi data.

Algoritma yang akan digunakan untuk kriptografi dikatakan aman bila memenuhi tiga kriteria. Ketiga kriteria tersebut adalah persamaan matematis yang digunakan harus kompleks, sehingga algoritma sulit dipecahkan secara analitik, biaya yang digunakan untuk memecahkan chipertext melampaui nilai informasi, dan waktu yang diperlukan untuk memecahkan chipertext melampaui lamanya waktu informasi.

Kekuatan algoritma enkripsi apapun tergantung pada kemampuan untuk menahan serangan yang ditujukan untuk menentukan key atau uncrypted communication. Secara garis besar ada dua jenis serangan, yaitu brute-force dan shortcut. Oleh karena itu pada makalah ini akan dibahas sejauh mana algoritma Skipjack menjamin keamanan suatu informasi yang disembunyikan.

Makalah ini juga akan membahas spesifikasi dari algoritma Skipjack. Selain itu di dalam makalah ini juga akan dibahas proses enkripsi dan dekripsi menggunakan algoritma Skipjack. Penerapan dari algoritma Skipjack juga akan coba dibahas dalam makalah kali ini.

Kata kunci : Skipjack, block chiper, enkripsi

1. PENDAHULUAN

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.

Salah satu algoritma yang tengah dikembangkan adalah Algoritma Skipjack. Skipjack adalah algoritma yang dikembangkan pada tahun 1987. Skipjack merupakan represantasi dari *family of encryption algorithms* yang dikembangkan pada tahun 1980 sebagai bagian dari algoritma Tipe I yang dikembangkan Badan Kemanan Nasional Amerika Serikat. Algoritma Tipe I sangat aman dan sangat rahasia.

Skipjack digunakan dalam dua *encryption device*, yaitu : the Clipper Chip dan Fortezza PC card. Perangkat ini memiliki banyak kegunaan dan digunakan oleh lembaga seperti FBI. Kriptanalisis pada Skipjack dapat dilakukan secara ekstensif dan tidak memiliki kelemahan. Tidak dikenal adanya serangan *shortcut* yang dapat merusak Skipjack. Namun, ukuran kunci yang kecil membuat algoritma lebih inferior pada kandidat terbaru untuk Advanced Encryption Standard (AES) kompetisi yang diadakan oleh NIST.

Walaupun masih kurang memuaskan, Skipjack menyediakan keamanan yang sangat kuat dan dapat bertahan selama bertahun-tahun sebelum algoritma ini dapat dirusak oleh *bruteforce attack*. Badan Keamanan Amerika juga tidak berniat untuk menjadikan Skipjack sebagai kandidat untuk AES. Seperti Triple DES, Skipjack merupakan solusi

sementara yang akan digunakan sampai AES terselesaikan dan dapat dilaksanakan secara luas, sehingga dapat dikatakan Skipjack menawarkan alternatif yang aman untuk DES tanpa harus bergantung pada AES.

Skipjack bekerja dalam 64-bit blok dan menggunakan 80-bit kunci. Dibutuhkan 64-bit blok plaintext sebagai masukan dan keluaran 64-bit blok ciphertext. Skipjack memiliki 32 putaran, yang berarti algoritma diulang 32 kali untuk menghasilkan ciphertext.

Telah ditemukan bahwa jumlah putaran berbanding secara eksponensial dengan jumlah waktu yang diperlukan untuk menemukan sebuah kunci menggunakan *brute force* attack sehingga apabila jumlah putaran meningkat, keamanan dari algoritma juga meningkat secara eksponensial.

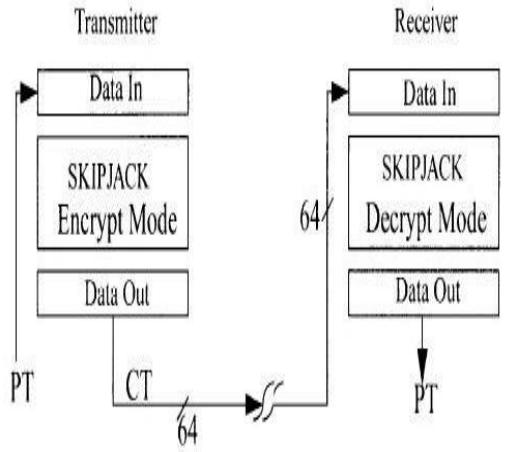
Deskripsi rinci Algoritma Skipjack terdapat dalam spesifikasi resmi yang disediakan oleh Badan Keamanan Nasional Amerika setelah algoritma tidak dirahasiakan.

2. MODE OPERASI SKIPJACK

Mode Operasi Skipjack merupakan subset dari deskripsi mode FIPS-81 untuk DES. Mode-mode tersebut adalah sebagai berikut :

A. ECB (Electronic Code Book)

Pada mode ECB, data dibagi menjadi 64-bit blok dan setiap blok dienkripsi satu per satu. Enkripsi dilakukan terpisah terhadap masing-masing blok Artinya, jika data yang dikirimkan melalui jaringan atau saluran telepon, transmisi kesalahan hanya akan mempengaruhi blok yang bersangkutan.



Gambar 1. Mode ECB

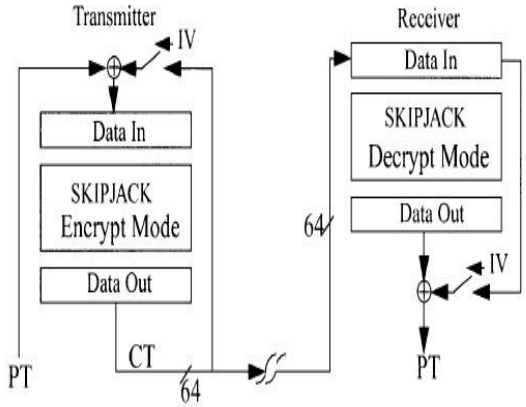
ECB merupakan metode terlemah karena tidak ada langkah-langkah keamanan tambahan yang diimplementasikan selain algoritma dasar Skipjack. Namun, ECB adalah yang tercepat dan termudah untuk diimplementasikan.

Tidak ada banyak informasi mengenai modus operasi yang banyak digunakan, tetapi untuk kebanyakan jenis ciphers blok, ECB adalah yang paling umum diterapkan. Modus operasi ini digunakan oleh Private Encryptor.

B. CBC (Cipher Block Chaining)

Dalam modus operasi ini, setiap blok ciphertext yang dienkripsi dengan modus ECB di-XOR dengan plaintext berikutnya yang akan dienkripsi, sehingga seluruh blok bergantung pada blok sebelumnya.

Untuk mencari plaintext dari blok tertentu, diperlukan ciphertext, kunci, dan ciphertext untuk blok sebelumnya. Blok pertama yang akan dienkripsi tidak memiliki ciphertext sebelumnya, sehingga plaintext untuk blik pertama adalah hasil XOR dengan angka 80-bit yang disebut initialization Vector, atau IV.



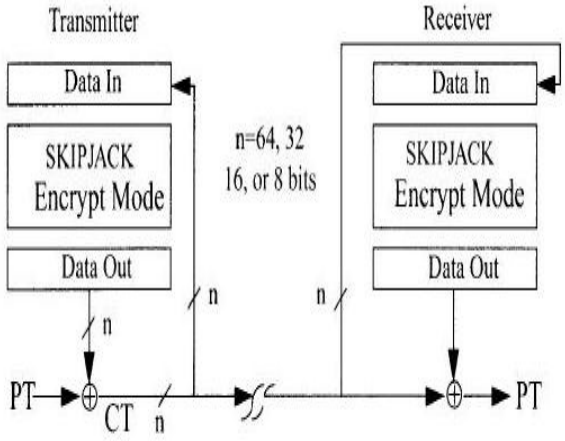
Gambar 2. Mode CBC

Jadi jika terdapat kesalahan transmisi dalam data yang dikirimkan melalui jaringan atau saluran telepon, kesalahan akan dibawa ke semua blok sampai blok terakhir. Modus operasi ini lebih aman daripada ECB karena langkah XOR tambahan menambahkan satu lapisan lagi ke proses enkripsi.

C. CFB (Cipher Feed Back)

Dalam mode ini, blok plaintext yang kurang dari 64 bit dapat dienkripsi. Biasanya, proses yang khusus digunakan untuk menangani file yang ukurannya tidak utuh 8 byte. Private Encryptor menangani kasus ini dengan menambahkan beberapa dummy byte ke akhir file sebelum melakukan enkripsi.

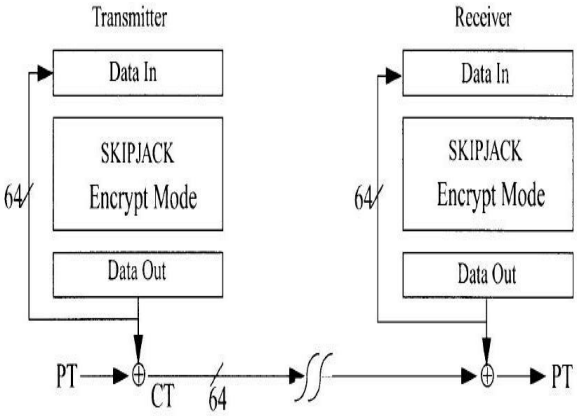
Plaintext sebenarnya tidak melalui proses Skipjack, tetapi hanya di-XOR dengan output blok. Proses ini dilakukan dengan cara 64-bit yang disebut blok Shift Register digunakan sebagai masukan plaintext pada Skipjack. Blok ini dibentuk untuk beberapa nilai yang berubah-ubah dan dienkripsi dengan algoritma Skipjack.



Gambar 3. Mode CFB

Kemudian ciphertext di-pass melalui sebuah komponen tambahan yang disebut kotak-M yang hanya memilih left-most M bits dari ciphertext, M adalah jumlah bit dalam yang ingin dienkripsi. Nilai ini di-XOR dengan plaintext dan hasilnya adalah final ciphertext.

Akhirnya ciphertext di-fedback pada Shift Register, dan digunakan sebagai *seed* untuk blok berikutnya yang akan dienkripsi. Seperti pada mode CBC, kesalahan dalam satu blok akan mempengaruhi semua blok selama pengiriman data. Modus operasi ini sama dengan CBC dan sangat aman, tetapi lebih lambat dari ECB karena kompleksitas yang ditambahkan.



Gambar 4. Mode OFB

D. OFB (Output Feed Back)

Mode ini mirip dengan mode CFB. Namun pada OFB, hasil ciphertext dari proses enkripsi di-fedback ke dalam Shift Register. Shift Register tersebut di-assign pada sebuah nilai awal yang dapat berubah-ubahdan dijadikan sebagai parameter dalam algoritma Skipjack.

Hasil dari proses Skipjack di-pass melalui M-Box, kemudian di-fedback ke dalam Shift Register untuk mempersiapkan blok berikutnya. Nilai ini kemudian di-XOR dengan plaintext yang panjangnya kurang dari 64 bit dan hasilnya adalah ciphertex akhir.

Tidak seperti CFB dan CBC, kesalahan dalam satu blok dalam mode OFB tidak akan mempengaruhi blok yang lain karena setelah blok penerima memiliki Shift Register awal, Shift Register yang baru akan terus di-generate tanpa masukan input data lagi. Namun, modus ini kurang aman daripada mode CFB karena hanya ciphertext dan hasil ciphertext Skipjack yang diperlukan untuk menemukan plaintext dari blok terakhir. Informasi tentang kunci pun tidak dibutuhkan.

3. SPESIFIKASI SKIPJACK

A. Notasi dan terminologi

- Vn : kumpulan semua nilai n-bit
- word : V16, nilai 16-bit
- byte : V8, nilai 8-bit
- permutasi Vn : fungsi permutasi pada Vn
- X ⊕ Y : operasi XOR X dan Y
- X ‖ Y : X di-concat dengan Y. X‖Y = X x 28 + Y (Y adalah word). X adalah high-order byte dan Y adalah low-order byte

B. Struktur Umum

Skipjack merupakan iterated blok cipher dengan 32 putaran yang memiliki dua tipe, yaitu Rule A dan Rule B. Setiap putaran dibuat dalam bentuk feedback linear shift register dengan tambahan permutasi G non-linear.

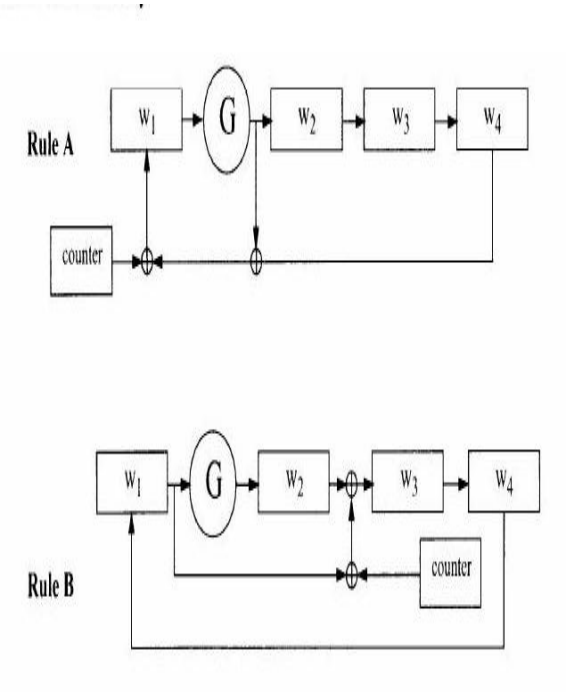
Rule B pada dasarnya adalah kebalikan dari Rule A dengan beberapa perbedaan pada pengaturan posisi. Skipjack melakukan delapan putaran terhadap Rule A, setelah itu melakukan delapan putaran Rule B, setelah itu melakukan delapan putaran lain dari Rule A dan delapan putran lain dari Rule B.

Definisi dari Rule A dan Rule B diberikan pada Gambar . Counter merupakan angka yang berurut dari 1 sampai 32. G adalah permutasi Feistel 4 putaran dimana fungsi F ditetapkan sebagai bit 8x8

S-box (disebut sebagai tabel F). Setiap putaran dari G dienkripsi dengan kunci 8-bit.

Langkah – langkah Rule A :

a. Fungsi G melakukan permutasi terhadap w1.
b. w1 yang baru adalah hasil XOR dari G, counter, dan w4.
c. w2 dan w3 menggeser satu register ke kanan dan menjadi w3 dan w4.
d. w2 yang baru adalah keluaran dari G
e. counter bertambah



Gambar 5. Rule A dan Rule B

Rule B berjalan seperti Rule A.

C. Proses Dekripsi dan Enkripsi

Proses enkripsi

Saat proses dimulai, counter bernilai 1. Algoritma melakukan Rule A sebanyak 8 kali, lalu berganti melakukan Rule B sebanyak 8 kali. Berganti melakukan Rule A sebanyak 8 kali lagi dan menyelesaikan enkripsi dengan 8 putaran Rule B. Counter bertambah setiap menyelesaikan satu langkah.

ENCRYPT

Rule A	Rule B
$w_1^{k+1} = G^k(w_1^k) \oplus w_4^k \oplus counter^k$	$w_1^{k+1} = w_4^k$
$w_2^{k+1} = G^k(w_1^k)$	$w_2^{k+1} = G^k(w_1^k)$
$w_3^{k+1} = w_2^k$	$w_3^{k+1} = w_1^k \oplus w_2^k \oplus counter^k$
$w_4^{k+1} = w_3^k$	$w_4^{k+1} = w_3^k$

Gambar 6. Proses Enkripsi

Proses Dekripsi

Algoritma dimulai dengan nilai counter 32. Algoritma melakukan Rule B ^ -1 untuk 8 langkah, Rule A ^ -1 untuk 8 langkah, Rule b ^ -1 selama 8 langkah lagi dan akhirnya Rule ^ -1 untuk 8 langkah lain. Counter berkurang satu setiap langkahnya.

DECRYPT

Rule A^{-1}	Rule B^{-1}
$w_1^{k-1} = [G^{k-1}]^{-1}(w_2^k)$	$w_1^{k-1} = [G^{k-1}]^{-1}(w_2^k)$
$w_2^{k-1} = w_3^k$	$w_2^{k-1} = [G^{k-1}]^{-1}(w_2^k) \oplus w_3^k \oplus counter^{k-1}$
$w_3^{k-1} = w_4^k$	$w_3^{k-1} = w_4^k$
$w_4^{k-1} = w_1^k \oplus w_2^k \oplus counter^{k-1}$	$w_4^{k-1} = w_1^k$

Gambar 7. Proses Dekripsi

C. Permutasi – G

Permutasi-G menggunakan cryptovariable 10-byte. Fungsi-G adalah Feistel 4 putaran. Setiap putaran melakukan fungsi substitusi dengan Tabel F. Setiap putaran menggunakan satu byte cryptovariable. Karakteristik dari fungsi G adalah sebagai berikut :

- Secara rekursif : Gk(w=g1||g2) dmana gi = F (gi-1⊕cv 4k+i-3) ⊕ gi-2. K adalah urutan langkah(langkah pertama adalah 0), F adalah tabel substitusi dan cv4k+i-3 adalah byte ke (4k+i-3) pada cryptovariable schedule

$$g_3 = F(g_2 \oplus cv_{4k}) \oplus g_1$$

$$g_4 = F(g_3 \oplus cv_{4k+1}) \oplus g_2$$

$$g_5 = F(g_4 \oplus cv_{4k+2}) \oplus g_3$$

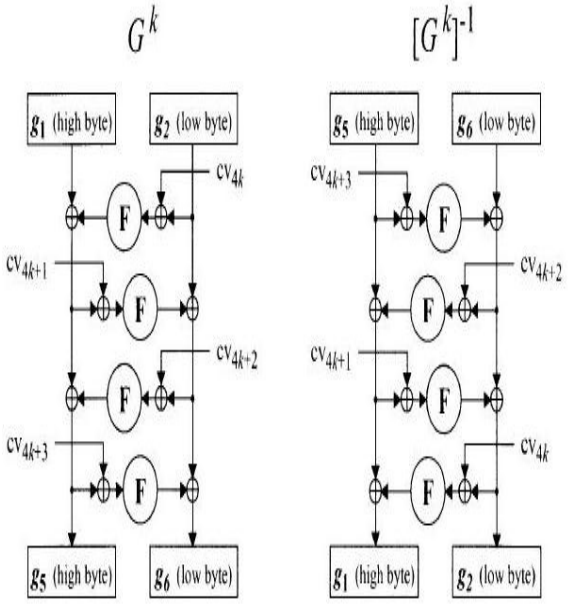
$$g_6 = F(g_5 \oplus cv_{4k+3}) \oplus g_4$$

Similarly, for the inverse, $[G^k]^{-1}(w = g_5 \| g_6) = g_1 \| g_2$ where

$$g_{i-2} = F(g_{i-1} \oplus cv_{4k+i-3}) \oplus g_i.$$

Gambar 8. Proses Permutasi

- Secara skema :



Gambar 9. Skema permutasi

Cryptovariable schedule memiliki panjang 10 byte. Penjadwalan kunci berbentuk siklus. Urutan dari 4 byte sub-kunci diulang setiap lima putaran dan hanya ada lima yang serupa seperti itu.

Berhubung byte kunci dibagi dalam 2 set : byte ganjil dan byte genap. Byte genap selalu memasuki putaran yang genap dari permutasi-G dan byte ganjil selalu masuk urutan putaran ganjil.

D. Tabel-F

Tabel-F di bawah ini diberikan dalam notasi heksadesimal. High-order 4 bit dari index input sebagai baris dan low order 4 bits sebagai kolom index.

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	a3	d7	09	83	f8	48	f6	f4	b3	21	15	78	99	b1	af	f9
1x	e7	2d	4d	8a	ce	4c	ca	2e	52	95	d9	1e	4e	38	44	28
2x	0a	df	02	a0	17	f1	60	68	12	b7	7a	c3	e9	fa	3d	53
3x	96	84	6b	ba	f2	63	9a	19	7c	ae	e5	f5	f7	16	6a	a2
4x	39	b6	7b	0f	c1	93	81	1b	ee	b4	1a	ea	d0	91	2f	b8
5x	55	b9	da	85	3f	41	bf	e0	5a	58	80	5f	66	0b	d8	90
6x	35	d5	c0	a7	33	06	65	69	45	00	94	56	6d	98	9b	76
7x	97	fc	b2	c2	b0	fe	db	20	e1	eb	d6	e4	dd	47	4a	1d
8x	42	ed	9e	6e	49	3c	cd	43	27	d2	07	d4	de	c7	67	18
9x	89	cb	30	1f	8d	c6	8f	aa	c8	74	dc	c9	5d	5c	31	a4
Ax	70	88	61	2c	9f	0d	2b	87	50	82	54	64	26	7d	03	40
Bx	34	4b	1c	73	d1	c4	fd	3b	cc	fb	7f	ab	e6	3e	5b	a5
Cx	ad	04	23	9c	14	51	22	f0	29	79	71	7e	ff	8c	0e	e2
Dx	0c	ef	bc	72	75	6f	37	a1	ec	d3	8e	62	8b	86	10	e8
Ex	08	77	11	be	92	4f	24	c5	32	36	9d	cf	f3	a6	bb	ac
Fx	5e	6c	a9	13	57	25	b5	e3	bd	a8	3a	01	05	59	2a	46

Gambar 10. Tabel F

4. ANALISIS

Pada serangan *Brute force*, penyerang mencoba setiap key yang mungkin pada potongan cipher text sampai terjemahan plaintext diperoleh dengan jelas. Umumnya, setengah dari semua kemungkinan kunci harus dicoba untuk mencapai kesuksesan. Hal ini juga sering disebut exhaustive search.

Resource yang diperlukan untuk melakukan exhaustive search tergantung pada panjang kunci. Kunci dengan panjang N bit memiliki 2 ^ N kemungkinan. Skipjack menggunakan kunci 80-bit, yang berarti terdapat 2 ^ 80 (sekitar 10 ^ 24) atau lebih dari 1 triliun kemungkinan kunci.

Sebagai contoh, implementasi algoritma Skipjack dioptimalkan pada satu prosesor 8-Cray YMP yang melakukan enkripsi sekitar 89.000 per detik. Proses itu akan membutuhkan 400 miliar tahun untuk mencoba seluruh kunci.

Dengan asumsi penggunaan delapan prosesor akan mengurangi wakt satu miliar tahun. Dengan 100.000 RISC prosesor, masing-masing yang mampu melakukan 100000 enkripsi per detik, masih akan memakan waktu sekitar 4 juta tahun.

DES memenuhi properti yang diberikan untuk pasangan plaintext dan ciphertext dan kunci yang terkait. Enkripsi salah satu komplemen plaintext degan salah satu komplemen kunci melengkapi komplemen dari ciphertext.

5. PENERAPAN SKIPJACK

Fortezza adalah sistem keamanan informasi yang berbasis pada PC Card. Setiap individu yang berhak untuk melihat informasi yang dilindunginya diberikan Fortezza Card yang menyimpan private key dan data lainnya yang dibutuhkan untuk mendapatkan akses. Fortezza berisi sebuah keamanan microprocessor yang disebut Capstone (MYK-80). Capstone menerapkan algoritma Skipjack.

Fortezza Card telah digunakan di pemerintahan, militer, dan aplikasi perbankan untuk melindungi data yang sensitif. Card dapat dipertukarkan dengan berbagai jenis peralatan yang mendukung Fortezza. Kunci dan program dapat dibuat ulang

Fortezza dikembangkan untuk proyek Clipper Chip Pemerintah AS dan telah digunakan oleh Pemerintah AS di berbagai aplikasi. Fortezza Card (KOV-8) yang asli adalah produk Type II yang berarti tidak dapat digunakan untuk informasi yang rahasia.

Produk yang paling banyak digunakan adalah produk Type I (KOV-12) yang digunakan secara luas untuk Defense Message System (DMS).. Sebuah versi yang lebih baru, yang disebut KOV-14 atau Fortezza Plus, menggunakan Krypton microprocessor. Pada gilirannya KOV-14 akan diganti dengan PC CARD KSV-21yang lebih baru dengan algoritma yang lebih modern dan kemampuan tambahan.

Enkripsi suara dan data NSA's Secure Terminal Equipment menggunakan Fortezza Plus Card yang diproduksi oleh Mykotronx Corporation dan Spyrus.

6. KESIMPULAN

Dari hasil pemaparan di atas, kesimpulan yang di dapat adalah sebagai berikut :

Dengan asumsi biaya pemrosesan dibagi dua setiap delapan belas bulan, maka akan dibutuhkan 36 tahun sebelum biaya memecahkan Skipjack oleh exhaustive search akan sama dengan biaya memecahkan DES saat ini.

Dengan demikian, tidak ada risiko yang signifikan bahwa Skipjack akan dipecahkan dalam 30-40 tahun ke depan, sehingga dapat dikatakan bahwa Algoritma Skipjack sangat aman untuk diterapkan.

DAFTAR PUSTAKA

[1] http:// www.cspr.org/program/clipper/ skipjack_interim_review.html Tanggal akses: 31 Maret 2009 pukul 20:00

[2] http://www.cs.technion.ac.il/ ~biham / Reports / skipjack/ Tanggal akses: 31 Maret 2009 pukul 20:15

[3] Munir, Rinaldi. (2004). Bahan Kuliah IF5054 Kriptografi. Departemen Teknik Informatika, Institut Teknologi Bandung.

[4] http://www.funet.fi/pub/Crypt/criptography /symmetric/skipjack/ Tanggal akses: 31 Maret 2009 pukul 20:45