Analisis Perbandingan Algoritma LOKI 91 dan International Data Encryption Algorithm(IDEA)

Sidik Soleman 13508101[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]sidik_324@students.itb.ac.id, sidiksoleman@gmail.com

Abstrak*—Ada beberapa algoritma blok cipher yang termasuk dalam ketegori algoritma kriptografi modern. Beberapa algoritma tersebut adalah International Data Encryption Algorithm(IDEA) dan algoritma LOKI 91 yang merupakan pengembangan dari algoritma LOKI 89. Blok cipher termasuk dalam algoritma kunci simetri artinya kunci yang sama digunakan untuk melakukan enkripsi maupun dekripsi. Blok merupakan kumpulan dari bit-bit data yang akan dienkripsi maupun didekripsi. IDEA dan LOKI 91 merupakan algoritma yang direkomendasikan sebagai pengganti dari algoritma DES(Data Encryption Standard) yang sudah tidak aman lagi digunakan. Karena direkomendasikan sebagai pengganti DES, prinsip kedua algoritma ini selain mirip satu sama lain, juga mirip dengan DES. Kedua algoritma memiliki ukuran blok yang sama dan keduanya pula sama-sama menggunakan jaringan feistel serta keduanya melakukan proses enkripsi berkali-kali. Perbedaanya adalah panjang kunci yang digunakan. Salah satu perbedaan kedua algoritma tersebut adalah panjang kunci yang digunakan. Dengan menggunakan jariangan feistel dan proses enkripsi berkali-kali, penentuan *key scheduler* juga sangat menentukan kekuatan algoritma ini. Karena itu penting untuk menganalisis kedua algoritma ini baik dari segi proses algoritma, jenis-jenis serangan yang bisa dilakukan ke algoritma ini dan tentu performasi serta pemilihan key scheduler yang tepat untuk menentukan kekuatan algoritma ini. Hasil cipher teks keduanya memiliki ukuran yang sama dengan plain teks yang dienkripsi.

***Kata Kunci*—IDEA, LOKI 91, LOKI 89, DES, blok cipher, jaringan feistel, *key scheduler*.**

I. PENDAHULUAN

Blok cipher termasuk dalam algoritma kriptografi modern. Operasi pada blok cipher menggunakan operasi bit-bit yang dikelompokkan menjadi satu blok. Ini berarti plainteks yang akan dienkripsi dikelompokkan menjadi beberapa blok-blok. Ada banyak algoritma yang termasuk dalam algoritma kriptografi modern. Dua diantaranya adalah LOKI 91 dan IDEA. Kedua algoritma ini merupakan algoritma yang direkomendasikan untuk menjadi pengganti atau alternatif dari algoritma DES. DES sendiri sudah dianggap tidak aman lagi karena dengan *exhautive search key* bisa dilakukan. Hal ini didukung dengan resource untuk melakukan komputasi tersebut sudah dianggap tidak menjadi persoalan lagi.

Karena direkomendasikan sebagai pengganti DES secara otomatis karateristik dari kedua algoritma ini tidak berbeda dengan DES. Ketiga algoritma tersebut memiliki kemiripan pada jumlah blok yang digunakan yaitu 64-bit. DES dan LOKI 91 memiliki jumlah kunci yang sama yaitu 64-bit sedangkan untuk IDEA memiliki kunci sebanyak 128-bit. Karakteristik kedua dari kedua algoritma ini pula adalah memiliki jaringan feistel. Kedua algoritma ini mengenkripsi cipherteks hasil enkripsi beberapa kali atau yang biasa disebut dengan *n-round times*. Untuk LOKI 91 menggunakan 16 kali putaran, persis sama dengan DES sedangkan IDEA menggunakan 8 kali putaran.

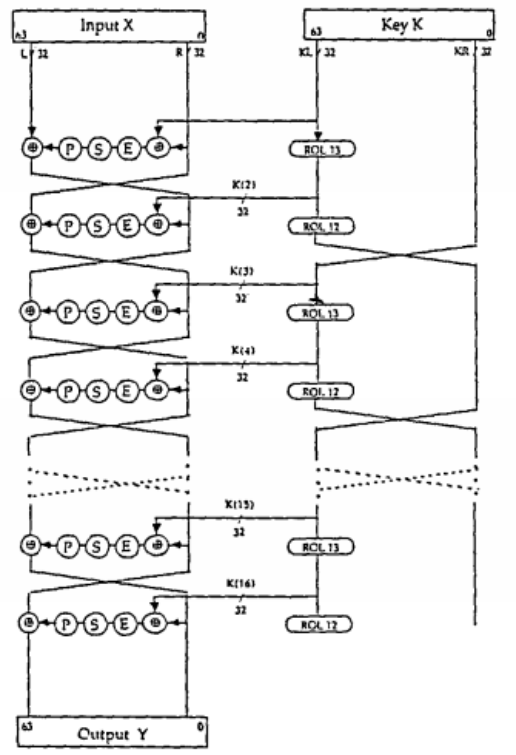
LOKI 91 algoritma juga sama-sama seperti pada DES menggunakan *S-Boxes* (*subtitution box)*. *S-Boxes* merupakan metode pemetaan plainteks menjadi ciperteks. Dengan ini tidak ada fungsi linier yang menghubungkan antara cipherteks dan plainteks. Saat melakukan fungsi putaran, digunakan pula *keyscheduler*. *Keyscheduler* merupakan fungsi yang akan menghasilkan kunci yang akan digunakan dalam pengulangan n-kali dengan kata lain fungsi ini akan memperluas *master key* yang ada. Selain penentuan *S-Boxes*, bagian penting dari algoritma blok cipher adalah pemilihan *keyscheluder*. Pemilihan ini akan menentukan tingkat keamanan dari algoritma blok cipher. Selain itu, performasi juga perlu diperhatikan dalam mendesain algoritma blok cipher. Biasanya dalam algoritma blok cipher untuk memberikan keamanan yang lebih berbanding terbalik dengan efisiensi. Hal ini terlihat dari operasi yang digunakan yaitu operasi bit-bit. Semakin rumit fungsi yang digunakan dan semakin banyak putaran yang digunakan, semakin lama pula proses enkripsi pada algoritma blok cipher. Begitu pula semakin panjang plainteks yang dimasukkan semakin lama pula waktu yang digunakan untuk mengenkripsi. Oleh karena itu, penting untuk menganalisis kedua algoritma ini, selain menarik karena memiliki beberapa kesamaan.

II. ALGORITMA LOKI 91

Algoritma LOKI 91 merupakan penyempurnaan dari LOKI 89. Beberapa perubahan dilakukan dari algoritma LOKI 89. Perubahan tersebut terkait dengan tingkat keamanan yang diberikan dari algoritma ini yaitu agar menghasilkan cipherteks yang kuat. Setelah sebelumnya

LOKI 91 ditunjukkan kelemahanya oleh kriptanalisis. Karena algoritma ini termasuk dalam algoritma *DES-like*, LOKI 91 menggunakan pengulangan 16 kali dalam fungsi yang biasa disebut dengan fungsi f. Ukuran blok dan kunci sama persis dengan ukuran yang digunakan dalam DES yaitu 64-bit. Setiap pengulangan disebut dengan *round*. Karena operasinya bit-bit hasil keluaran plainteks tidak akan bertambah artinya jika plainteks masukannya adalah 512 bit maka keluarnnya pun akan memiliki panjang yang sama.

Pada saat proses enkripsi, input pada setiap *round* dibagi menjadi dua bagian. Bagian satu digunakan didalam fungsi f dengan dioperasikan 32 bit kunci round yang ditentukan oleh *keyschedule*. Output dari fungsi f ini akan ditambahkan dengan modulo 2 dari bagian kedua. Hal ini dilakukan sebanyak 16 kali mengingat LOKI 91 memiliki *round* 16. Untuk pemetaan *S-Boxes* berdasarkan fungsi Galois field(2^8). Secara umum algoritma LOKI 91 terdapat pada skema di bawah ini



Gambar 1. Skema umum algoritma LOKI 91

Dari skema di atas dapat diketahui bahwa proses enkripsi plainteks menjadi cipherteks melalu proses pengulangan ekripsi sebanyak 16 kali yang masing-masing pengulangan dioperasikan dengan 32 bit key. Operasi yang ada adalah permutas, subtitusi, ekslusif or, dan ekspansi.

Penggunaan permutasi adalah untuk meningkatkan ketergantungan antara cipher dengan input. LOKI 91 menggunakan jaringan feistel. Dalam algoritma LOKI 91digunakan fungsi putaran. Fungsi ini merupakan sebagai fungsi pembangkit kunci untuk *round* selanjutnya. Fungsi putaran ini merupakan fungsi putaran dengan sirkular sift bit. Pada LOKI digunakan nilai 12 dan 13 hal ini agar memberikan kemungkinan kunci yang berbeda tiap pembentukan kunci. Kunci yang ukurannya 64 bit, sebelum masuk fungsi putaran ini, dibagi terbagi menjadi dua yang masing-masing berukuran 32 bit. Pembangkitan kunci langsung dilakukan 16 kali sebelum melakukan proses enkripsi. Dengan tiap iterasi pembangkitan membangkitkan 4 buah kunci, sehingga total ada 4 iterasi. Dua iterasi awal menggunakan bagian kunci pertama dengan menggeser 12 dan 13 kali, sedangkan dua bagian selanjutnya menggunakan bagian kunci kedua dengan menggeser 12 dan 13 kali.

Pada saat enkripsi, blok dibagi menjadi dua dengan ukuran yang sama besar kemudian operasi ini diakhiri dengan penukaran bit pada bagian sebelah kiri dan kanan. Pada diagram proses pertama yang dilakukan yaitu melakukan operasi XOR antara kunci dengan bagian sebelah kanan dari plainteks atau bagian pertama plainteks. Hasil operasi ini akan masuk ke dalam fungsi E. Fungsi E akan melakukan ekspansi menjadi 4x12 bit. Selanjutnya hasil ekspansi ini akan masuk ke dalam *S-Boxes*. Keluaran dari *S-Boxes* adalah 4x8 bit akan dipermutasikan agar menghasilkan keluaran 32 bit.

Berikut ini adalah daftar ekspansi pada fungsi E

3	2	1	0	31	30	29	28	27	26	25	24
27	26	25	24	23	22	21	20	19	18	17	16
19	18	17	16	15	14	13	12	11	10	9	8
11	10	9	8	7	6	5	4	3	2	1	0

31	23	15	7	30	22	14	6
29	21	13	5	28	20	12	4
27	19	11	3	26	18	10	2
25	17	9	1	24	16	8	0

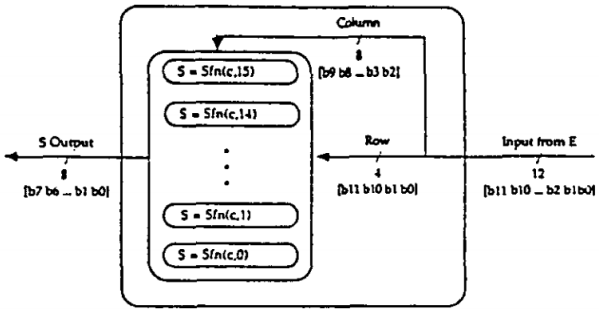
Row	gen_{row}	e_{row}
0	375	31
1	379	31
2	391	31
3	395	31
4	397	31
5	415	31
6	419	31
7	425	31
8	433	31
9	445	31
10	451	31
11	463	31
12	471	31
13	477	31
14	487	31
15	499	31

Gambar 2. Nilai pengekspansi kunci, Nilai permutasi P, dan *S-Boxes*

S-Boxes pada algoritma ini terdiri dai 16 fungsi. *S-Boxes* memiliki input 12 dan memiliki ouput 8. Persamaan yang digunakan adalah

$$Sfn(row, col) = (col + ((row * 17) \oplus ff_{16}) \& ff_{16})^{31} mod\, g_{row}$$

Berikut ini adalah skema pada *S-Boxes* nya



Gambar 3. Skema pada fungsi *S-Boxes*

Pada proses dekripsi, mirip dengan proses dekripsi, hanya saja dengan putaran yang dibalik. Pertama cipherteks yang akan didekripsi dibagi menjadi dua seperti pada proses enkripsi, yaitu bagian kanan dan bagian kiri. Setelah itu masing-masing dimasukkan ke dalam fungsi yang sama dengan proses enkripsi tapi memiliki putaran yang dibalik dan proses terakhir dari dekripsi pada LOKI 91 adalah melakukan penukaran antara bagian kanan dan bagian kiri.

III. ALGORITMA INTERNATIONAL DATA ENCRYPTION ALGORITHM(IDEA)

International Data Encryption Algorithm(IDEA) merupakan algoritma lain yang direkomendasikan sebagai pengganti DES. Algoritma ini dipublikasikan pada tahun 1991 bersamaan dengan algoritma LOKI 91 juga. IDEA memiliki ukuran blok sebanyak 64 bit dan panjang kunci masukan adalah 128 bit. Jumlah putaran atau *round* pada IDEA sebanyak 8 kali yang disertai dengan transformasi output.

Putaran r menggunakan enam dari 16 kunci yang telah dibangkitkan $K_i^{(r)}$ untuk $1 \leq i \leq 6$. Putaran ini ditujukan untuk mentranformasikan 64-bit masukan menjadi keluaran yang berukuran 16 bit, yang selanjutnya digunakan sebagai input pada putaran berikutnya. Artinya proses ini dilakukan sebanyak 8 kali untuk masing-masing blok yang ada. Pada putaran terakhir, dilakukan transformasi setelah keluaran dari putaran. Proses ini menggunakan empat kunci yang telah dibangkitkan sebelumnya.

Keamanan pada IDEA didasari pada percampuran tiga buah operasi. Operasi tersebut adalah penambahan dengan modulo 2^{16}, perkalian dengan modulo $2^{16}+1$, dan operasi ekslusif OR. Ketiga operasi ini bekerja pada satu blok yang berukuran 16 bit.

Proses enkripsi pada IDEA dimulai dengan input 64 bit blok dan 128 bit kunci. Keluaran dari fungsi ini adalah 64 bit blok cipherteks sebanyak empat buah. Hal pertama adalah dalam proses enkripsi adalah membangkitkan kunci atau subkey (*K*) untuk digunakan pada putaran dan digunakan pada transformasi output.

Plainteks dikelompokkan menjadi empat buah dalam empat variabel yang masing-masing berukuran 16 bit. Misal keempat variabel tersebut adalah X_1,X_2,X_3,X_4. Selanjutnya untuk putaran dari satu sampai ke delapan dilakukan operasi

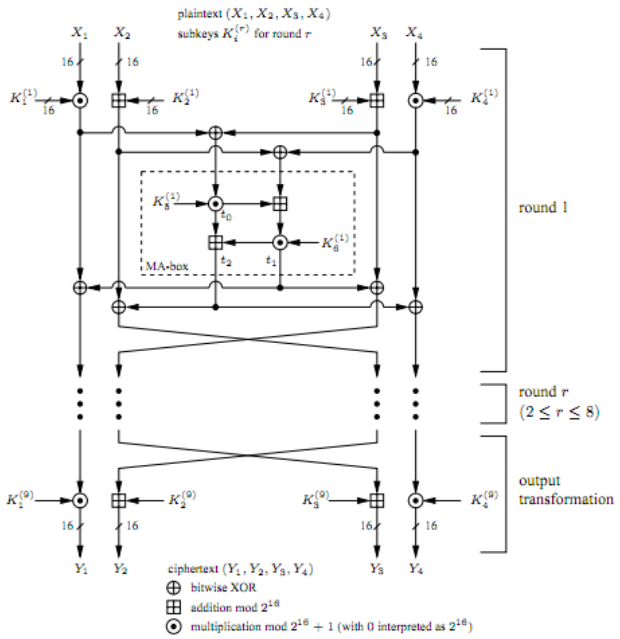
$$
\begin{aligned}
(a)\quad & X_1 \leftarrow X_1 \odot K_1^{(r)} \\
& X_4 \leftarrow X_4 \odot K_4^{(r)} \\
& X_2 \leftarrow X_2 \boxplus K_2^{(r)} \\
& X_3 \leftarrow X_3 \boxplus K_3^{(r)}. \\
(b)\quad & t_0 \leftarrow K_5^{(r)} \odot (X_1 \oplus X_3) \\
& t_1 \leftarrow K_6^{(r)} \odot (t_0 \boxplus (X_2 \oplus X_4)) \\
& t_2 \leftarrow t_0 \boxplus t_1. \\
(c)\quad & X_1 \leftarrow X_1 \oplus t_1 \\
& X_4 \leftarrow X_4 \oplus t_2 \\
& a \leftarrow X_2 \oplus t_2 \\
& X_2 \leftarrow X_3 \oplus t_1 \\
& X_3 \leftarrow a.
\end{aligned}
$$

Selanjutnya untuk transformasi output dilakukan dengan menggunakan operasi berikut ini

$$
\begin{aligned}
Y_1 &\leftarrow X_1 \odot K_1^{(9)} \\
Y_4 &\leftarrow X_4 \odot K_4^{(9)} \\
Y_2 &\leftarrow X_3 \boxplus K_2^{(9)} \\
Y_3 &\leftarrow X_2 \boxplus K_3^{(9)}.
\end{aligned}
$$

$K^{(9)}$ merupakan kunci yang dibangkitkan setelah membangkitkan kunci untuk putaran. Dalam kunci ini terdapat 4 buah kunci.

Secara umum skema dalam algoritma IDEA dapat digambarkan sebagai berikut ini



Gambar 4. Skema umum algoritma IDEA

Fungsi *keyschedule* juga terdapat pada algoritma ini, seperti halnya pada LOKI 91. Fungsi ini menggunakan

masukan kunci yang panjangnya 128 bit atau dua kali lebih panjang dari kunci pada algoritma LOKI 91. Keluaran kunci ini adalah 52 kunci yang ukurannya masing-masing 16-bit. Kunci ini digunakan untuk proses dalam putaran dan proses transformasi. Kunci-kunci tersebut disusun sebagai berikut $K_1^{(1)}$.... $K_6^{(1)}$,..., $K_1^{(8)}$.... $K_6^{(8)}$, $K_1^{(1)}$.... $K_4^{(1)}$.

Untuk mendapatkan kunci ini sebelumnya kunci dipartisi menjadi blok yang masing-masing berukuran 16-bit. Selanjutnya nilai ini dimasukkan kedelapan nilai subkey pertama. Untuk menghasilkan subkey selanjutnya dilakukan dengan operasi penggeseran kunci sebanyak 25 bit secara sirkular. Setelah digeser kemudian dipartisi menjadi blok yang berukuran 16 bit dan masukan nilai ini pada subkey berikutnya.

Untuk proses dekripsi dilakukan dengan langkah yang mirip dengan proses dekripsi. Perbedaanya hanya ada proses pembangkitan yang dilakukan oleh *keyschedule*. Pembangkitan subkey menggunakan fungsi yang sesuai pada tabel berikut ini

round r	$K'^{(r)}_1$	$K'^{(r)}_2$	$K'^{(r)}_3$
$r = 1$	$(K_1^{(10-r)})^{-1}$	$-K_2^{(10-r)}$	$-K_3^{(10-r)}$
$2 \le r \le 8$	$(K_1^{(10-r)})^{-1}$	$-K_3^{(10-r)}$	$-K_2^{(10-r)}$
$r = 9$	$(K_1^{(10-r)})^{-1}$	$-K_2^{(10-r)}$	$-K_3^{(10-r)}$

round r	$K'^{(r)}_4$	$K'^{(r)}_5$	$K'^{(r)}_6$
$r = 1$	$(K_4^{(10-r)})^{-1}$	$K_5^{(9-r)}$	$K_6^{(9-r)}$
$2 \le r \le 8$	$(K_4^{(10-r)})^{-1}$	$K_5^{(9-r)}$	$K_6^{(9-r)}$
$r = 9$	$(K_4^{(10-r)})^{-1}$	—	—

Gambar 5. Tabel menentukan subkey

Pada tabel diatas $-K_i$ merupakan additive inverse modulo 216 dari K_i yang merupakan bilangan interger yang terletak diantara $0 \le u \le 216$. Selanjutnya nilai ini dimasukkan sebagai subkey yang baru. Lambang multiplicatiove inverse (modulo 2^{16} +1) dari K_i merupakan nilai yang dapat diturunkan dariperluasan algoritma *euclidean*.

VI. KRIPTANALISIS DAN PERFORMANSI PADA ALGORITMA LOKI 91

Karena LOKI 91 merupakan penyempurnaan dari LOKI 89, penyempurnaan ini lebih ditujukan untuk tujuan akademis terkait dengan kriptanalisis yang telah dilakukan pada LOKI 89 menggunakan differential kriptanalisis. Differential kriptanalisis merupakan suatu metode kriptanalisis yang menyerang sebuah cipher dengan dinamis. Penyerangan ini dilakukan dengan menggunakan pemilihan pasangan plainteks, dengan menggunakan analisis statistik pada cipherteksnya. Secara umum metode ini lebih cepat dibandingkan dengan *exhaustive search*. Metode ini pertama dilakukan untuk menganalisis DES.

Pada differensial kirptanalisis, digunakan sebuah tabel yang menunjukkan pasangan antara plainteks dan cipherteks. Tabel ini digunakan untuk menujukkan statistik pengamatan antara output dan plainteks. Untuk cipher yang berulang maka prediksi kemungkinan kemunculan untuk tiap round juga akan dicatat dan kemungkinan kemunculanya. Jika ada 7 putaran maka setiap putaran dilakukan prediksi kemunculan dan kemungkinan kemunculan tersebut. Oleh karena itu, operasi XOR yang semula muncul pada LOKI 89 tidak digunakan lagi dalam LOKI 91. Dengan meningkatkan jumlah putaran dari lima menjadi delapan membuat ketegantungan cipher bit pada LOKI 91 meningkat, selain itu, *keyschedule* yang digunakan dalam algoritma ini juga mampu menangani untuk menghasilkan kunci yang berbeda.

Selain itu, penambahan kunci yang dilakukan sebelum ekspansi, hal ini akan menambah sulit untuk dilakukan mencari karakteristik pada zero round(putaran saat input masih sama dengan input yang sesungguhnya). Sehingga hal ini akan membuat banyak kemungkinan pada penebakan cipher pada round berikutnya. Dengan ini LOKI 91 resistan terhadap serangan kirptanalisis dengan differensial kriptanalisis. Dengan *S-Boxes* yang dimiliki oleh LOKI 91, sangat kecil kemungkinan untuk menghasilkan output 0 dari keempat output yang ada. Pasti dari salah satu keempat tersebut ada yang bernilai tidak 0. Hal ini adalah salah satu kelebihan dari algoritma LOKI 91.

Walau begitu, pada algoritma ini juga ditemui pasangan kunci yang lemah. Berikut ini daftar kunci(*kunci bertanda '*' merupakan pasangan yang berhasil direduksi dari LOKI 89) tersebut

Encrypt Key	Decrypt Key
0000000000000000	0000000000000000 *
00000000aaaaaaaa	aaaaaaaa00000000
0000000055555555	5555555500000000
00000000ffffffff	ffffffff00000000
aaaaaaaa00000000	00000000aaaaaaaa
aaaaaaaaaaaaaaaa	aaaaaaaaaaaaaaaa *
aaaaaaaa55555555	55555555aaaaaaaa
aaaaaaaaffffffff	ffffffffaaaaaaaa
5555555500000000	0000000055555555
55555555aaaaaaaa	aaaaaaaa55555555
5555555555555555	5555555555555555 *
55555555ffffffff	ffffffff55555555
ffffffff00000000	00000000ffffffff
ffffffffaaaaaaaa	aaaaaaaaffffffff
ffffffff55555555	55555555ffffffff
ffffffffffffffff	ffffffffffffffff *

Gambar 6. Pasangan kunci lemah dan semi lemah pada algoritma LOKI 91

Serangan lain yang mungkin dilakukan pada LOKI 91 adalah menggunakan plainteks pilihan(*choosen paintext*). Metode ini akan membantu pada pencarian melalu *exhaustive search*. Algoritma untuk melakukan ini adalah sebagai berikut :

1. Ambil P random, kemudian dibagi menjadi dua bagian yaitu Pkanan dan Pkiri. Kemudian lakukan proses enkripsi sehingga diperoleh hasil C, dan C*.

C merupakan hasil enkripsi dengan *P* dan C* adalah hasil enkripsi dari $\overline{P}$.

2. Untuk a dari 0 sampai dengan 2^{32}-1 :
   $P(a) \rightarrow E_2(P,a)$. $P(a)=P_{kanan}(a) \,||\, P_{kiri}(a)$, dimana
   $P_{kiri}(a) = F(P_{kanan,}a) \; XOR \; P_{kiri}$
   $P_{kanan}\,(a) = F(P_{kiri}(a), Rol_{13}(a)) \; XOR \; P_{kanan}$
3. Ambil semua hasil enkripsi dari *C(a), C*(a)* untuk *P(a)* dan $\overline{P(a)}$.

Untuk proses selanjutnya lakukan dengan kriptanalisis dengan *exhaustive search* untuk masing-masing kunci. Dengan menggunakan metode ini untuk memecahkan algoritma LOKI 91 di estimasikan

Estimates for	Time	Space	Chosen plaintexts
	1.07×2^{62}	$2^{33}+2$	$2^{33}+2$

Gambar 7. Waktu estimasi pada kriptanalisis LOKI 91 menggunakan *chosen plaintext* dan *exhaustive search*.

Metode lain yang digunakan setelah *chosen plaintext* adalah dengan melakukan enumerasi pada kunci yang mungkin dibangkitkan selama putaran dikombinasikan dengan *chosen plaintext*.

Dengan menggunakan ini, pada langkah kelima jumlah iterasi yang terjadi adalah mendekati 2^{62}. Sedangkan waktu yang dihabiskan untuk menyelesaikan selama 16 putaran adalah 1.07 x 2^{62}. Dengan metode *chosen plaintext* saja mampu menurunkan faktor pencarian empat kali dibandingkan dengan murni menggunakan *exhaustive search*.

V. KRIPTANALISI DAN PERFORMANSI PADA IDEA

IDEA merupakan salah satu algoritma yang berhasil menghilangkan kelemahan linier pada cipher yang dihasilkannya. Namun, lemah dalam penggunaan *keyschedule* yang cenderung lebih sederhana. Selain itu, walaupun berhasil menghilangkan efek linier, tapi ternyata algoritma ini masih bisa diserang dengan serangan linier. Efek linier ini ternyata masih dapat ditemui dari tiga buah fungsi operasi utama yang dimiliki IDEA. Untuk putaran ke-5 IDEA menggunakan 2^{19} plainteks yang diketahui memiliki kompleksitas waktu pemecahan sebesar 2^{103}. Karena lemahnya *keyschedule* dapat dilakukan dengan serangan *related-key*.

Serangan yang mungkin dilakukan untuk hal itu adalah differensial kiptanalisis dengan menggunakan chosen *plaintext*. Untuk serangan dengan putarak ke lima dibutuhkan 2^{24} *chosen plaintext* yang telah dienkripsi dan memiliki kompleksitas waktu 2^{126}. Untuk serangan yang menggunakan *related-key* membutuhkan $2^{57,8}$ *chosen plaintext* yang telah dienkipsi menggunakan emapt *related-key* dan kompleksitas waktu yang digunakan adalah $2^{88.1}$.

Untuk pasangan *weak key* berhasil diungkap ada sekitar 2^{64} kunci. Untuk melakukan proses ini dibutuhkan 2^{16} *adaptive chosen plaintext*.

VI. ANALISIS KEYSCHEDULE

Dari beberapa serangan yang ditujukan pada kedua algoritma tersebut, kebanyakan penyerangan yaitu memanfaatkan *keyschedule* yang lemah. Beberapa kelemahan biasanya adalah adanya hubungan linier dari fungsi yang digunakan dan hubungan kunci dari *keyshcedule* dalam menghasilkan kunci-kunci putaran.Jika hal ini terjadi secara otomatis, akan dapat dilakukan kriptanalisis dengan metode baik *chosen plaintext, related-key, exhaustive search,* maupun dengan differensial kriptanalisis.

IDEA masih belum bisa menghidari hubungan keterkaitan linier dalam fungsi-fungsi yang digunakan dalam proses enkripsi sebaliknya LOKI 91 berhasil menghilangkan hubungan linier dengan memanfaatkan *S-Boxes* yang dimilikinya. Berkat *S-Boxes* tersebut juga mampu menghindari keluarnya output 0. Kelebihan lain yang dimiliki oleh LOKI 91 adalah operasi penambahan dengan kunci sebelum masuk ke dalam fungsi. Teknik ini berhasil mengecilkan kemungkinan untuk melakukan kiptanalisis dengan menggunakan differensial kriptanalisis karena hubungan pada *zero round* sudah tidak berhubungan lagi dengan plainteks inputan semula.

Untuk keyschedule yang digunakan keduanya, masih memiliki hubungan antara kunci yang satu dengan kunci yang lain. Pada LOKI 91 keyschedule masih diproduksi menggunakan operasi shift bit, begitu pula pada IDEA yang masih menggunakan operasi-operasi bit, yang tentu akan menambah unsur linier dalam pembangkitan kunci antara kunci yang satu dengan yang lainnya.

Dan tentu untuk kedua algoritma ini masih bisa dipecahkan menggunakan *exhaustive search* dengan kombinasi beberapa metode kriptanalisis. Walaupun waktu estimasi penyelesaian dan jumlah *resource* yang dibutuhkan banyak, namun dengan seiring perkembangan dari teknologi, masalah *resource* tidak akan menjadi perosalan yang cukup berat.

Untuk pemilihan *keyschedule* yang cukup meningkatkan keamanan adalah dengan mengkombinasikan dengan fungsi hash. Sehingga hubungan linier antara kunci dalam satu putaran dengan putaran yang lain tidak diketahui hubungan liniernya.

VII. KESIMPULAN

Dari analisis perbandingan antara algoritma LOKI 91 dan IDEA dapat disimpulkan beberapa hal berikut ini:

1. Algoritma LOKI 91 yang merupakan penyempurnaan tidak terlalu banyak mengalami perubahan hanya membuang beberapa operasi yang

membuat algoritma ini lemah yaitu operasi XOR sebelum masuk kedalam fungsi.

2. Operasi penjumlahan dengan kunci sebelum masuk kedalam fungsi putaran mampu meningkatkan kekuatan algoritma dengan mengurangi kemungkinan diserang menggunakan diffesensial kriptanalisis.
3. *Keyschedule* yang lemah yaitu yang memiliki unsur ke linier dalam membangkitkan kunci-kunci yang digunakan dalam putaran.
4. Penggunaan *S-boxes* mampu meningkatkan keamanan dalam kriptografi karena bisa menghilangkan hubungan linier dan dapat menangani output yang mengeluarkan 0 yang bisa dimanfaatkan pada kriptanalisis seperti yang dilakukan pada algoritma LOKI 91.
5. Algortima LOKI 91 dan IDEA sama-sama sudah tidak aman untuk digunakan karena keduanya dapat diserang dengan *exhaustive search* dengan mengkombinasikan beberapa analisis kriptografi seperti *chosen plaintext*.
6. Algoritma IDEA lebih lemah lagi karena bisa dilakukan differesial kriptanalisis.

REFERENCES

[1] Biham, Eli, Adi Shamir, "Differential Cryptanalysis of DES-like Cryptosystem".The Weizmann Institute of Science Departement of Applied Mathematics.

[2] Brown, Lawrence, Mattew Kwan, Josef Pieprzyk, Jennifer Sebberry, *"Improving Resitance toDIfferential Crypanalysis and the Redesign of LOKI"*.Departement of Computer Science, University College, UNSW, Austalian Defence Force Academy.Canberra ACT 2006. Australia.

[3] Biham, Eli, Orr Dunkelman, Nathan Keller."New Cyptanalytic Result on IDEA. Israel.

[4] Kesley, Johan, Burce Sneier, David Wagner, "*Keyschedule Cyptanalysis of IDEA, G-DES, GOST, SAFER, and Triple-DES*" Berkeley.

[5] Knudsen,Lars Ramkilde, "Cryptanalysis of LOKI 91" . Departement of Computer Science University of Aarhus.Denmark.

[6] Brown, Dr Lawry.1999.LOKI 89 and LOKI 91.http://www.unsw.adfa.edu.au/~lpb/research/loki91/loki.html.

[7] Menezes, A, et all. Handbook of Applied Cryptorgaphy.1996.CRC Press.

[8] Biham, Eli, Orr Dunkelman, Nathan Keller.A New Attack On 6-Round IDEA.Computer Science Departement Technion Haifa Iswael, Enistein Institute of Mathematics Hebrew university, Katholiek e Universsitetit Leuven Dept. of Elcetrical Enginering

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 29 April 2010

Sidik Soleman -13508101