Analisis Boomerang Attack dan Sandwich Attack untuk Memecahkan Enkripsi Pengamanan Jaringan GSM 3G

Muhammad Ghufron Mahfudhi / 13508020
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
if18020@students.if.itb.ac.id

***Abstrak*—Jaringan telekomunikasi pada telepon seluler yang marak digunakan saat ini adalah jaringan GSM 3G. Pada jaringan tersebut digunakan enkripsi A5/3 atau yang lebih dikenal dengan nama Kasumi. Tidak lama setelah enkripsi ini disebarkan, ternyata ada pihak dari Israel yang dapat memecahkan enkripsi tersebut dengan menggunakan *boomerang attack* dan *sandwich attack*. Untuk dapat mendesain enkripsi yang lebih aman, kita harus mempelajari serangan-serangan terlebih dahulu. Oleh karena itu, dengan adanya analisis terhadap serangan tersebut dapat dihasilkan algoritma enkripsi yang lebih baik.**

***Kata Kunci*—Boomerang Attack, Sandwich Attack, Kasumi, Block Cipher.**

I. PENDAHULUAN

Seiring dengan perkembangan zaman, dunia telekomunikasi juga berkembang pesat. Saat ini, kita telah mengenal yang namanya telepon seluler. Dengan telepon seluler ini kita dapat berkomunikasi dengan orang lain tanpa memandang jarak yang ada. Telepon seluler pun selalu berkembang dan menyediakan banyak fitur-fitur baru yang berguna bagi konsumen.

Telepon seluler membutuhkan jaringan untuk bisa berkomunikasi dengan telepon seluler lainnya. Jaringan telepon seluler yang telah berkembang saat ini yaitu jaringan GSM. Karena jaringan ini tidak menggunakan kabel, jaringan ini menjadi lebih rentan terhadap serangan daripada jaringan pada telepon kabel. Oleh karena itu, perlu dilakukan penerapan sistem pengacakan guna mengamankan privasi jalinan komunikasi antara perangkat ponsel dengan menara BST pada jaringan telepon seluler GSM. Sekarang ini, pengamanan tersebut dilaksanakan dengan sejenis sistem enkripsi pengacakan stream ciphers atau terkadang dikenal dengan A5/1 dan A5/2 yang masa operasinya telah memasuki lama penggunaan 20 tahun lebih. Sistem A5/1 & A5/2 ini merupakan varian *cryptosystem* yang dikenal dengan nama "Misty". Menurut rencana sistem enkripsi ini pada jaringan seluler GSM masa depan bermuatan konektivitas 3G akan digantikan dengan varian *cryptography* sistem *block cipher* 64-bit berkode A5/3 yang dikenal dengan nama "Kasumi" dan sebenarnya adalah sebuah versi modifikasi dari *cryptosystem* "Misty".

Ternyata sistem kriptografi yang bernama Kasumi itu telah berhasil dipecahkan oleh Nathan Keller dan timnya dengan menggunakan *boomerang attack* dan *sandwich attack*. Oleh karena itu, diperlukan sistem pengamanan yang lebih kuat dari enkripsi Kasumi. Untuk mendapatkan cara enkripsi yang lebih baik, terlebih dahulu kita harus bisa mempelajari cara pemecahan yang telah dilakukan. Oleh karena itu, dalam makalah ini akan dibahas mengenai analisis *boomerang attack* dan *sandwich attack* untuk memecahkan enkripsi pengamanan jaringan GSM 3G tersebut.

II. ENKRIPSI PADA JARINGAN GSM 3G

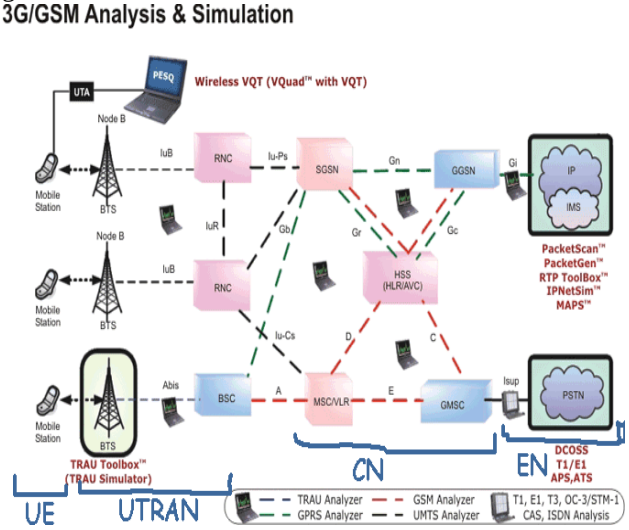
A. *Jaringan GSM 3G*

GSM (*Global System for Mobile communication*) adalah suatu teknologi yang digunakan dalam komunikasi *mobile* dengan teknik *digital*. GSM telah memberikan alernatif berkomunikasi baru bagi dunia telekomunikasi yang lebih kuat. Dengan menggunakan sistem sinyal digital dalam transmisi datanya, membuat kualitas data maupun *bit rate* yang dihasilkan menjadi lebih baik dibanding sistem *analog*. Teknologi GSM saat lebih banyak digunakan untuk komunikasi seluler dengan berbagai macam layanannya.

UMTS (*Universal Mobile Telecommunication* System) merupakan suatu revolusi dari GSM yang mendukung kemampuan generasi ketiga (3G). UMTS menggunakan teknologi akses WCDMA dengan system DS-WCDMA (*Direct Seqence Wideband Code Division Multiple Access*). Terdapat dua mode yang digunakan dalam WCDMA dimana yang pertama menggunakan FDD (*Frequency Division Duplex*) dan kedua dengan menggunakan TDD (*Time Division Duplex*). FDD dikembangkan di Eropa dan Amerika sedangkan TDD dikembangkan di Asia. Pada WCDMA FDD, digunakan sepasang frekuensi pembawa 5 MHz pada *uplink* dan *downlink* dengan alokasi frekuensi untuk *uplink* yaitu 1945 MHz – 1950 MHz dan untuk downlink yaitu 2135 MHz – 2140 MHz. Perbandingan antara *spreading rate* (kecepatan chip tiap detik) terhadap *user data rate* (kecepatan simbol data user tiap detik) dikenal sebagai *spreading factor*. Hal ini menandakan bahwa semakin tinggi *chip rate*, maka semakin banyak user yang

dapat ditampung. Pengertian lainnya adalah dalam menentukan jumlah user, semakin besar jumlah *chip rate*, maka semakin tinggi kecepatan data yang diperoleh masingmasing user. Dalam WCDMA, *chip rate* yang digunakan sebesar 3,84 Mbps.

Arsitektur pada UMTS-WCDMA dapat dilihat pada gambar berikut ini:



Gambar 1. Arsitektur Jaringan GSM 3G

Berdasarkan gambar di atas, jaringan UMTS-WCDMA pada intinya terdiri dari empat bagian, yaitu:

1. UE (*User Equipment*)
   UE adalah nama yang berhubungan dengan *terminal* atau *mobile*. *Terminal mobile* terhubung ke *Mobile Station* untuk membangun koneksi. Untuk terhubung dengan jaringan, terminal mobile membutuhkan kartu UMTS. Pada intinya, UE ini merupakan perangkat pada sisi pelanggan yang berupa headset untuk mengirim dan menerima informasi.
2. UTRAN (UMTS *Terresterial Radio Access Network*)
   UTRAN merupakan jaringan akses radio *terestrial* pada UMTS. Jaringan akses radio ini menyediakan koneksi antara *terminal mobile* dan *Core Network*. UTRAN terdiri dari satu atau lebih Jaringan Sub-Sistem Radio (RNS). Sebuah RNS merupakan suatu sub-jaringan dalam UTRAN dan terdiri dari Radio Network Controller (RNC) dan satu atau lebih Node B. RNS dihubungkan antar RNC melalui suatu Iur *Interface* dan Node B dihubungkan dengan satu Iub *Interface*.
3. CN (*Core Network*)
   CN merupakan jaringan inti yang telah dibangun sebelum adanya UMTS seperti GSM dan GPRS. CN menggabungkan fungsi kecerdasan dan transport. CN mendukung pensinyalan dan transportasi informasi dari trafik, termasuk peringanan beban trafik. Dengan melewati CN, UMTS juga dihubungkan dengan jaringan telekomunikasi lain, jadi sangat memungkinkan tidak hanya antara pengguna UMTS mobile, tetapi juga dengan jaringan yang lain.
4. EN (*External Network*)
   EN merupakan jaringan luar yang akan diakses oleh *User Equipment*. EN dapat berupa jaringan lain, seperti PS *Domain* (internet) atau CS *Domain* (PLMN, PSTN, ISDN, dll).

B. *Enkripsi Block Cipher*

Block Cipher merupakan jenis algoritma kriptografi moderen dengan menggunakan kunci simetris. Maksud dari kunci simetris tersebut yaitu dalam setiap proses enkripsi atau dekripsi digunakan kunci yang sama.

Block Cipher ini akan membagi bit-bit plainteks menjadi blok-blok bit dengan panjang yang sama. Panjang kunci enkripsi harus sama dengan panjang blok. Enkripsi dilakukan terhadap blok bit plainteks menggunakan bit-bit kunci. Algoritma enkripsi ini akan menghasilkan blok cipherteks yang panjangnya sama dengan blok plainteks.

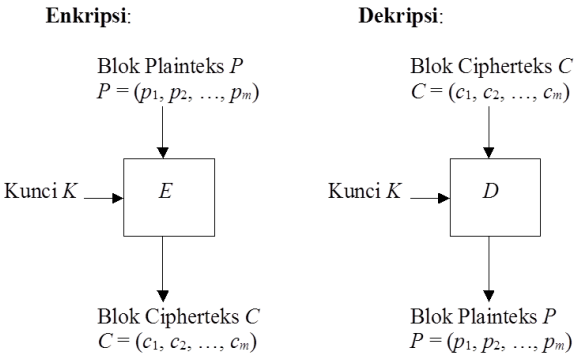
Jika diketahui blok plainteks (P) berukuran m bit:

$$P = (p_1, p_2, ..., p_m), p_i \in \{0,1\}$$

dan blok cipherteks (C) berukuran m bit:

$$C = (c_1, c_2, ..., c_m), c_i \in \{0,1\}$$

Maka skema umum proses enkripsi dan dekripsi tampak pada gambar berikut:

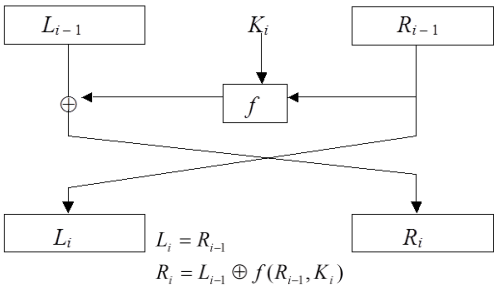


Gambar 2. Skema Umum Enkripsi dan Dekripsi *Cipher Block*

Dalam merancang algoritma cipher block, kita harus memperhatikan:

1. Prinsip *diffusion* dan *confussion* dari Shannon
   Tujuan *diffusion* yaitu untuk menyembunyikan hubungan apapun yang ada antara plainteks, cipherteks, dan kunci dengan membuat hubungan statistik antara plainteks, cipherteks, dan kunci menjadi sangat rumit. Sedangkan *confussion* menyebarkan pengaruh satu bit plainteks atau kunci ke sebanyak mungkin cipherteks, sehingga menghasilkan perubahan pada cipherteks yang tidak dapat diprediksi.
2. *Cipher* berulang (*iterated cipher*)
   Cipher berulang menggunakan fungsi transformasi sederhana yang mengubah plainteks menjadi cipherteks diulang sejumlah kali. Pada setiap putaran digunakan upa-kunci (*subkey*) atau kunci putaran (*round key*) yang dikombinasikan dengan plainteks.

3. Jaringan Feistel (*Feistel Network*)
   Skema jaringan Feistel tampak pada gambar di bawah ini:



Gambar 3. Skema Umum Jaringan Feistel

4. Kunci lemah (*weak key)*
   Kunci lemah adalah kunci yang menyebabkan tidak adanya perbedaan antara enkripsi dan dekripsi. Dekripsi terhadap cipherteks tetap menghasilkan plainteks semula, namun enkripsi dua kali berturut-turut terhadap plainteks akan menghasilkan kembali plainteksnya.
5. Kotak-S (*S-box*)
   Kotak-*S* adalah matriks yang berisi substitusi sederhana yang memetakan satu atau lebih bit dengan satu atau lebih bit yang lain. Masukan dari operasi *look-up table* dijadikan sebagai indeks kotak-*S*, dan keluarannya adalah *entry* di dalam kotak-*S*.

C. Enkripsi A5/3 (Kasumi)

Kasumi merupakan enkripsi *block cipher* yang diterapkan pada jaringan sistem komunikasi GSM 3G (UMTS). Kasumi dibuat oleh *Security Algorithms Group* of Experts (SAGE) untuk *3rd Generation Partnership Project* (3GPP) yang akan digunakan pada jaringan UMTS.

Kasumi menerima 64 bit plainteks dan 128 bit kunci, kemudian mengeluarkan 64 bit cipherteks. Kasumi menggunakan delapan putaran jaringan Feistel. Fungsi putaran pada jaringan Feistel merupakan transformasi jaringan yang mirip jaringan Feistel yang tidak bisa dikembalikan. Dalam setiap fungsi putaran menggunakan *round key* yang terdiri dari delapan buah upakunci berukuran 16 bit yang didapatkan dari 128 bit kunci asli dengan menggunakan fungsi penjadwalan kunci.

Pada fungsi penjadwalan kunci, pada mulanya 128 bit kunci dibagi menjadi 16 bit upakunci sebagai berikut:

$$K = K_1 \parallel K_2 \parallel K_3 \parallel K_4 \parallel K_5 \parallel K_6 \parallel K_7 \parallel K_8$$

Kemudian juga menggunakan kunci K' yang didapatkan dengan rumus:

$$K' = K \oplus X$$

dengan

$$X = 0 \times 123456789ABCDEFFEDCBA9876543210$$

$$X =$$

Round key didapatkan dari upakunci dengan memutar bit ke kiri pada jumlah tertentu dan dari upakunci yang telah diubah.

Proses penjadwalan kunci dapat disimpulkan seperti gambar di bawah ini:

Round	$KL_{i,1}$	$KL_{i,2}$	$KO_{i,1}$	$KO_{i,2}$	$KO_{i,3}$	$KI_{i,1}$	$KI_{i,2}$	$KI_{i,3}$
1	$K_1 \lll 1$	K'_3	$K_2 \lll 5$	$K_6 \lll 8$	$K_7 \lll 13$	K'_5	K'_4	K'_8
2	$K_2 \lll 1$	K'_4	$K_3 \lll 5$	$K_7 \lll 8$	$K_8 \lll 13$	K'_6	K'_5	K'_1
3	$K_3 \lll 1$	K'_5	$K_4 \lll 5$	$K_8 \lll 8$	$K_1 \lll 13$	K'_7	K'_6	K'_2
4	$K_4 \lll 1$	K'_6	$K_5 \lll 5$	$K_1 \lll 8$	$K_2 \lll 13$	K'_8	K'_7	K'_3
5	$K_5 \lll 1$	K'_7	$K_6 \lll 5$	$K_2 \lll 8$	$K_3 \lll 13$	K'_1	K'_8	K'_4
6	$K_6 \lll 1$	K'_8	$K_7 \lll 5$	$K_3 \lll 8$	$K_4 \lll 13$	K'_2	K'_1	K'_5
7	$K_7 \lll 1$	K'_1	$K_8 \lll 5$	$K_4 \lll 8$	$K_5 \lll 13$	K'_3	K'_2	K'_6
8	$K_8 \lll 1$	K'_2	$K_1 \lll 5$	$K_5 \lll 8$	$K_6 \lll 13$	K'_4	K'_3	K'_7

$(X \lll i)$ — X rotated to the left by i bits

Gambar 4. Penjadwalan Kunci pada Kasumi

Setelah mendapatkan *round key*, algoritma Kasumi membagi 64 bit plainteks menjadi dua buah 32 bit, yaitu kiri (L_i) dan kanan (R_i). Pada setiap putaran, bagian kanan di-XOR-kan dengan keluaran dari fungsi putaran yang sebelumnya telah dipisah dulu. Prosesnya adalah sebagai berikut:

$$L_i = F_i(KL_i, KO_i, KI_i, L_{i-1}) \oplus R_{i-1}$$
$$R_i = L_{i-1}$$

KL_i, KO_i, KI_i merupakan kunci putaran untuk putaran ke-i.

Fungsi putaran untuk putaran ke-bilangan ganjil dan ke-bilangan genap berbeda, tetapi merupakan komposisi dari fungsi FL_i dan FO_i. Pada putaran ganjil menggunakan fungsi:

$$F_i(K_i, L_{i-1}) = FO(KO_i, KI_i, FL(KL_i, L_{i-1}))$$

dan pada putaran genap menggunakan fungsi:

$$F_i(K_i, L_{i-1}) = FL(KL_i, FO(KO_i, KI_i, L_{i-1})).$$

Fungsi FL dan FO di atas membagi 32 bit data masukan menjadi dua buah 16 bit. Fugsi FL tersebut merupakan manipulasi bit yang tidak bisa dikembalikan, sedangkan fungsi FO adalah tiga putaran jaringan yang mirip dengan Feistel yang tidak bisa dikembalikan.

Fungsi FL yang dilakukan dimulai dari membagi 32 bit masukan x dari FL(KL, x) menjadi dua buah 16 bit, kemudian dijalankan fungsi berikut ini:

$$r' = ROL(l \wedge KL_{i,1}, 1) \oplus r$$
$$l' = ROL(r' \vee KL_{i,2}, 1) \oplus l$$

Keluaran dari fungsi FL yaitu hasill konkatenasi dari l'

dan r' ($x' = l' \| r'$).

Fungsi FO yang dilakukan yaitu dimulai degan membagi 32 bit masukan x dari FO(KO,KI,x) dan ibagi menjadi dua buah 16 bit x = l_0 || r_0. Kemudian pada setiap tiga putaran (dengan indeks pertama, kedua, dan ketiga), bagian kiri dimodifikasi untuk mendapatkan kanan yang baru, sedangkan bagian kanan menjadi bagian kiri dari putaran selanjutnya. Untuk lebih mudahnya, dapat dilihat pada rumus berikut:

$$r_j = FI(KI, l_{j-1} \oplus KO_{i,j}) \oplus r_{j-1}$$
$$l_j = r_{j-1}$$

Fungsi FI di atas merupakan jaringan yang mirip dengan Feistel yang tidak beraturan.

Untuk fungsi FI tersebut, pada mulanya 16 bit masukan x dari fungsi FI(KI,x) dibagi menjadi 2 bagian, x = l_0 || r_0 di mana l_0 berukuran 9 bit dan r_0 berukuran 7 bit. Kemudian l_0 diacak dengan Kotak-S yang berukuran 9 bit dan hasilnya di-XOR r_0 yang telah ditambah 0 untuk mendapatkan 9 bit kanan yang baru. Untuk lebih jelasnya, dapat dilihat pada rumus di bawah:

$$r_1 = S9(l_0) \oplus (00 \| r_0)$$

Bit kanan r_0 diacak dengan kotak-S yang berukuran 7 bit , kemudian di-XOR dengan tujuh LSB dari kanan yang baru untuk mendapatkan 7 bit kiri. Untuk lebih jelasnya, dapat dilihat pada rumus di bawah:

$$l_1 = S7(r_0) \oplus LS7(r_1)$$

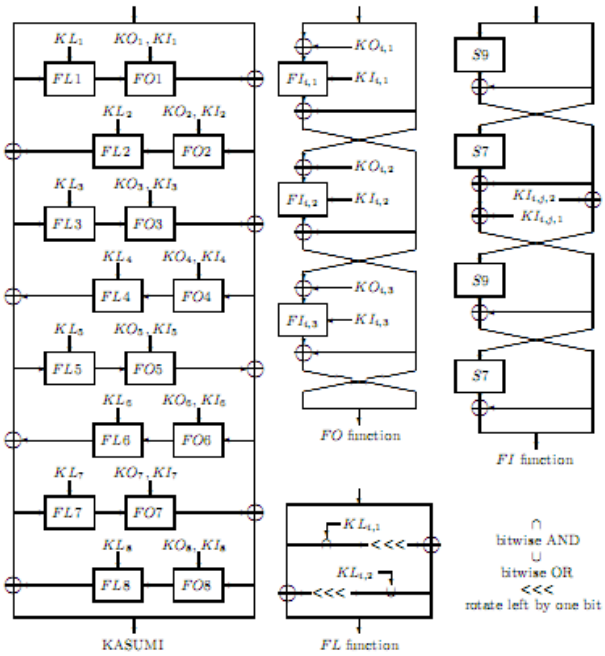
Kemudian menjalankan fungsi di bawah ini:

$$x_2 = KI \oplus x_1$$
$$r_3 = S9(r_2) \oplus (00 \| l_2)$$
$$l_3 S7(l_2) \oplus LS7(r_3)$$

Hasil akhir dari fungsi FI merupakan hasil konkatenasi dari l_3 dan r_3 (x' = l_3 || r_3).

Keluaran dari enkripsi Kasumi merupakan hasil konkatenasi dari bagian kiri dan kanan hasil pemutaran ($output = R_8 \| L_8$).

Secara umum, proses enkripsi Kasumi tampak pada diagram berikut ini:

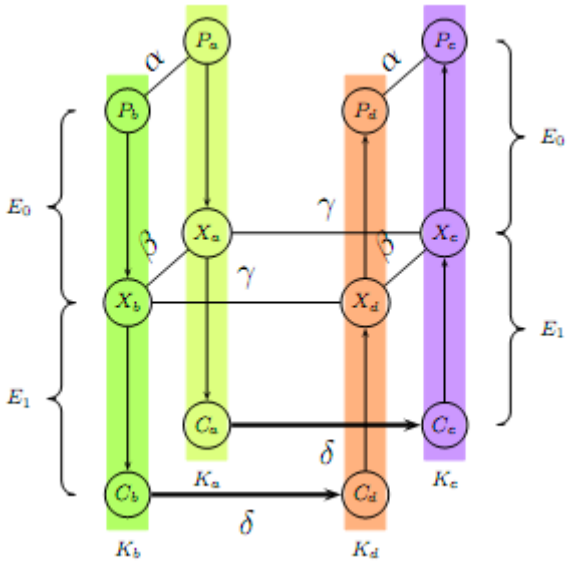


Gambar 5. Diagram Kasumi

III. Serangan pada Enkripsi A5/3 (Kasumi)

A. *Boomerang Attack*

Boomerang attack merupakan metode kriptanalisis terhadap enkripsi *block cipher* berdasarkan kriptanalisis diferensial. Dalam kriptanalisis diferensial, penyerang menganalisis bagaimana perbedaan masukan ke cipher atau plainteksnya dapat memberikan perbedaan pada keluarannya (cipherteksnya). Diferensial dengan kemungkinan terbesar dibutuhkan untuk menutupi semua atau hampir semua ciphernya. *Boomerang attack* mengizinkan diferensial untuk digunakan dalam menutupi hanya bagian dari cipher. Serangan ini digunakan untuk menghasilkan struktur *quartet* pada suatu titik menuju cipherteks.



Gambar 6. *Quartet* dari *Related-key Boomerang Attack*

Pada serangan ini, cipher diperlakukan sebagai aliran dari dua upacipher $E = E_1 \circ E_0$, serta kunci hubungan diferensial dari E_0 dan E_1 digabungkan menjadi suatu pengenal plainteks dan cipherteks yang dipilih untuk E. Misal, kita mengasumsikan bahwa terdapat sebuah diferensial *related-key* $\alpha \rightarrow \beta$ untuk E_0 dalam perbedaan kunci ΔK_{ab} dengan kemungkinan p. Misalnya, $\Pr[E_{0(K)}(P) \oplus E_{0(K \oplus K_{ab})}(P \oplus \alpha) = \beta] = p$, di mana $E_{0(K)}$ mendenotasikan enkripsi melalui E_0 pada kunci K. Kita juga mengasumsikan bahwa terdapat diferensial *related-key* $\gamma \rightarrow \delta$ untuk E_1 dengan perbedaan kunci ΔK_{ac} dengan probabilitas q. Pengenal *related-key boomerang* membutuhkan enkripsi atau dekripsi dengan kunci K_a, dan pada *related-key* $K_b = K_a \oplus \Delta K_{ab}$, $K_c = K_a \oplus \Delta K_{ac}$, dan $K_d = K_c \oplus \Delta K_{ab} = K_b \oplus \Delta K_{ac}$.

Sebuah *quartet boomerang* didapatkan dengan mengambil plainteks P_a secara acak dan meminta enkripsinya pada K_a, yang dinamakan $C_a = E_{K_a}(P_a)$. Kemudian, $P_b = P_a \oplus \alpha$ dienkripsi dengan K_b untuk mendapatkan $C_b = E_{K_b}(P_b)$. Dua cipherteks baru dikomputasi, $C_c = C_a \oplus \delta$ dan $C_d = C_b \oplus \delta$. Kemudian C_c didekripsi dengan K_c dan C_d didekripsi dengan K_d, misalnya $P_c = E_{K_c}^{-1}(C_c)$ dan $P_d = E_{K_d}^{-1}(C_d)$. Jika $P_c \oplus P_d = \alpha$, maka *quartet bomerang* yang benar telah ditemukan. Pada gambar 6 menunjukkan *quartet* dari *related-key boomerang* yang benar. Untuk permutasi acak, kemungkinan kondisi akhir ditemukan adalah 2-n, di mana n adalah ukuran blok. Untuk E, kemungkinan pasangan (P_a, P_b) merupakan pasangan yang benar berdasarkan diferesial yang pertama adalah p. Kemungkinan pasangan (C_a, C_c) dan (C_b, C_d) merupakan pasangan benar berdasarkan diferensial kedua adalah q^2. Jika semuanya merupakan pasangan yang benar, maka $E_1^{-1}(C_c) \oplus E_1^{-1}(C_d) = \beta = E_0(P_c) \oplus E_0(P_d)$. Oleh karena itu, dengan probabilitas p, $P_c \oplus P_d = \alpha$. Dari sini, jumlah kemungkinan *quartet* dari plainteks dan cipherteks yang sesuai dengan kondisi $P_c \oplus P_d = \alpha$ minimal adalah $(pq)^2$. Oleh karena itu, jika $pq \gg 2^{-n/2}$, dengan algoritma di atas dapat memisahkan E dari sebuah permutasi acak dengan kompleksitas $O((pq)^{-2})$.

B. *Sandwich Attack*

Sandwich attack merupakan jenis serangan yang hampir mirip dengan *boomerang attack*. Pada *sandwich attack* cipher diperlakukan sebagai aliran dari tiga upacipher, yaitu $E = E_1 \circ M \circ E_0$. Asumsi yang digunakan pada serangan ini sama dengan asumsi pada *boomerang attack*, yaitu terdapat sebuah diferensial *related-key* $\alpha \rightarrow \beta$ untuk E_0 dalam perbedaan kunci ΔK_{ab} dengan kemungkinan p dan terdapat diferensial *related-key* $\gamma \rightarrow \delta$ untuk E_1 dengan perbedaan kunci ΔK_{ac} dengan probabilitas q. Algoritma penyerangannya juga sama dengan algoritma penyerangan pada *boomerang attack*, dengan menghiraukan upacipher M. Akan tetapi, analisisnya lebih halus dan membutuhkan perhatian besar dalam menganalisis ketergantungan antara berbagai divisi.

Gagasan utama pada *sandwich attack* terketak pada transisi di tengahnya. Dalam *boomerang attack*, jika pasangan (P_a, P_b) merupakan pasangan yang tepat berdasarkan diferensial pertama, serta pasangan (C_a, C_c) dan (C_b, C_d) adalah pasangan yang tepat berdasarkan diferensial kedua, maka kita mempunyai:

$$(X_a \oplus X_b = \beta) \wedge (X_a \oplus X_c = \gamma) \wedge (X_b \oplus X_d = \gamma), \quad (1)$$

dengan X adalah nilai enkripsi perantara P. Dengan demikian,

$$X_c \oplus X_d = (X_c \oplus X_a) \oplus (X_a \oplus X_b) \oplus (X_b \oplus X_d) = \beta \oplus \gamma \oplus \gamma = \beta, \quad (2)$$

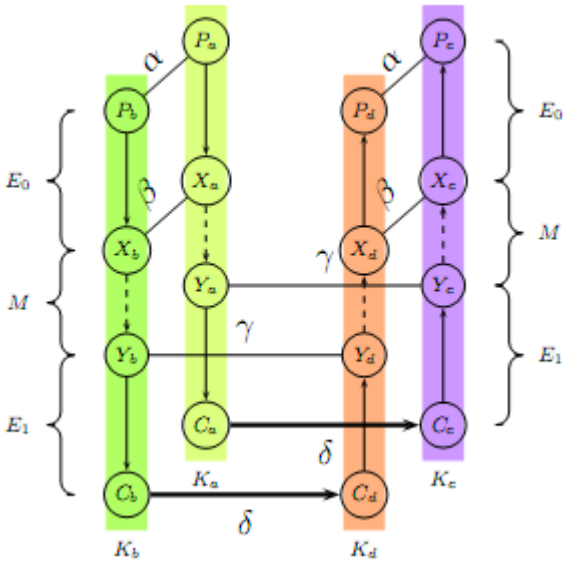
mengakibatkan $P_c \oplus P_d = \alpha$ dengan probabilitas p. Pada *sandwich attack* kita tidak mendapatkan kondisi (1), tetapi:

$$(X_a \oplus X_b = \beta) \wedge (Y_a \oplus Y_c = \gamma) \wedge (Y_b \oplus Y_d = \gamma). \quad (3)$$

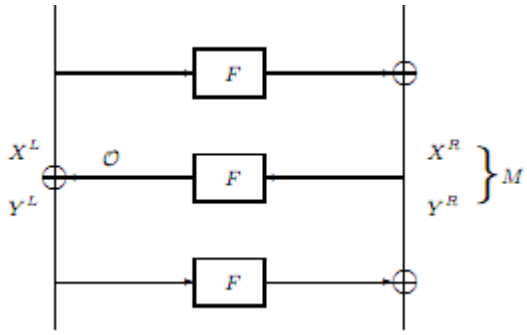
Oleh karena itu, probabilitas dari pengenal tiga layer *related-key boomerang* adalah p^2q^2r, di mana

$$r = \mathbf{Pr}\left[(X_c \oplus X_d = \beta) \middle| (X_a \oplus X_b = \beta) \wedge (Y_a \oplus Y_c = \gamma) \wedge (Y_b \oplus Y_d = \gamma)\right]. \quad (4)$$

Tanpa asumsi lagi, kemungkinan r menjadi sangat rendah, mendekati 2^{-n}. Dengan demikian, pengenal diharapkan gagal. Namun, dalam beberapa kasus perbedaan dalam E_0 dan E_1 dapat dipilih sedemikian rupa sehingga kesalahan probabilitas dalam melalui upacipher tengah adalah satu, yang jauh lebih tinggi dari yang diharapkan.



Gambar 7. *Quartet* dari *Related-key Sandwich Attack*



Gambar 8. Konstruksi Feistel dengan M pada Putaran Kedua

IV. Analisis Boomerang Attack dan Sandwich Attack

Pemecahan enkripsi Kasumi dengan menggunakan *boomerang attack* menghasilkan kompleksitas sebanyak $O((pq)^{-2})$. Pada penyerangan yang dilakukan oleh Bilham pada tahun 2005, serangan ini membutuhkan $2^{54,6}$ plainteks, masing-masing telah dienkripsi pada salah satu dari empat *related key*. Pada proses penyerangan tersebut, memiliki kompleksitas waktu sebanyak $2^{76,1}$ enkripsi Kasumi.

Pada pemecahan enkripsi Kasumi yang dilakukan oleh Nathan Keller, digunakan *related-key sandwich attack*. Kompleksitas data dari serangan tersebut adalah 2^{25} cipherteks terpilih dan 2^{25} plainteks terpilih yang dienkripsi/dekripsi pada satu dari empat kunci. Kompleksitas waktu yang dibutuhkan kira-kira sama dengan 2^{32} enkripsi. Probabilitas kesuksesan kira-kira 76%. Kompleksitas memori dari serangan juga termasuk sedang, butuh menyimpan 2^{26} pasangan plain/cipherteks di mana setiap pasangan membutuhkan 16 byte. Dengan demikian, jumlah memori yang digunakan 2^{30} byte atau kurang lebih 1 GB memori.

Dari data kompleksitas di atas, dapat diketahui bahwa penyerangan dengan *sandwich attack* mengeluarkan hasil yang lebih baik. Dari segi kompleksitas waktu, *sandwich attack* memiliki kompleksitas waktu yang lebih kecil. Jika kita kembali memandang algoritma yang telah dituliskan pada bab sebelumnya, *sandwich attack* menggunakan subcipher tambahan, yaitu M yang diletakkan di tengah. Hal ini justru membuat probabilitas berhasilnya penyerangan lebih tinggi.

Apabila kedua penyerangan tersebut digabungkan, kita dapat mengambil sisi positif dari kedua penyerangan tersebut. Dengan demikian, proses penyerangan juga bisa menjadi lebih baik dengan kompleksitas yang lebih kecil.

V. Kesimpulan dan Saran

Pada penyerangan terhadap enkripsi cipher blok A5/3 atau Kasumi, didapatkan hasil bahwa menggunakan *sandwich attack*-lah yang terbaik. Apalagi kalau dengan menggabungkan kedua penyerangan tersebut, pasti penyerangan bisa dilakukan dengan lebih efektif dan efisien dengan kompleksitas yang lebih rendah.

Sekarang kita sudah tahu bagaimana proses kerja penyerangan pada enkripsi Kasumi. Dalam proses desain enkripsi selanjutnya, seharusnya kita juga mempertimbangkan sisi penyerangan agar kita dapat membuat enkripsi yang aman dari serangan, jadi tidak hanya menyembunyikan proses pembuatannya saja. Dengan begitu, akan didapatkam algoritma enkripsi yang lebih baik dan lebih aman untuk jaringan telekomunikasi kita.

Referensi

http://www.informatika.org/~rinaldi/Kriptografi/2010-2011/kripto10-11.htm

http://sidarejaonline.wordpress.com/2010/01/23/riset-ilmuwan-israel-resep-untuk-membobol-jaringan-seluler-gsm-masa-depan/

http://carireferral.blogspot.com/2010/01/cara-menjebol-jaringan-handphone-gsm.html

http://en.wikipedia.org/wiki/Boomerang_attack

http://docs.google.com/viewer?a=v&q=cache:2XqpjWTuQmgJ:eprint.iacr.org/2010/013.pdf+sandwich+attack&hl=en&pid=bl&srcid=ADGEEShDk1WXmYmL5g1gMH3IRD8VT7Yn7AKqq_wzHPN_PyNyv9dQFBOsiK0l8w93-Q4Zv0W_hUvU6kp4wjCJARh5SfhntGShvXHpqHhHWlc1DDwwhYMg1GTz7C81K45Kl1wYtNwD83Ud&sig=AHIEtbS2bIeVGhtLdFNwOe8y-SFFP8Hc9g

http://www.ma.huji.ac.il/~nkeller/Crypt-conf2.pdf

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, 24 Maret 2011

Muhammad Ghufron Mahfudhi
13508020