Studi Stream Cipher A5/2 dan Serangan-serangan terhadap Stream Cipher A5/2

Darwin - 13508102
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
dawn0689@aol.com

ABSTRAK

Makalah ini akan membahas mengenai algoritma *Stream Cipher A5/2* dan serangan yang digunakan untuk mendekripsi ekripsi A5/2 ini. Adapun kegunaan dari *Stream Cipher A5/2* ini, di negara-negara di Eropa dan Amerika Serikat, algoritma ini digunakan untuk mendekripsi pembicaraan yang dilakukan melalui jaringan GSM yang digunakan. Enkripsi ini digunakan untuk menjamin privasi dari orang yang berbicara di telepon karena pada pembicaraan di telepon, hal yang dikirimkan merupakan bit sehingga kita dapat menyadap pesan tersebut dan mengkonversikannya ke suara untuk mendegarkan pesan yang disampaikan. Serangan terhadap enkripsi A5/2 ini dapat dilakukan untuk mendapatkan pesan yang diinginkan oleh penyadap. Akan dibahas mengenai serangan yang dapat dilakukan pada enkripsi A5/2 pada makalah ini. Teknik serangan yang akan dibahas adalah mengenai teknik serangan *known-plaintext attack* dan *ciphertext-only attack*.

Kata kunci: GSM, *Stream Cipher*, A5/2, *known-plaintext attack, ciphertext-only attack*.

I. PENDAHULUAN

Privasi merupakan hal sangat penting bagi setiap orang. Untuk menjamin privasi dari setiap orang, maka kita memerlukan suatu teknik enkripsi agar pesan atau media tersebut tidak dapat oleh orang lain. Pemrosesan enkripsi ini dapat dilakukan melalui dua macam teknik yaitu teknik penyisipan sebelum pesan tersebut dikirimkan dan pengenkripsian yang dilakukan pada saat pesan tersebut dikirimkan. Dengan adanya privasi tersebut, setiap orang dapat melakukan hal yang mereka inginkan tanpa diketahui orang lain.

Di setiap negara, penanganan yang dilakukan terhadap privasi setiap orang berbeda-beda terutama pada jaringan telepon GSM. Pada negara seperti Eropa dan Amerika Serikat, untuk menjamin privasi dari setiap pengguna jaringan telepon, setiap pesan yang dikirimkan pada pembicaraan dilakukan, pesan tersebut dienkripsikan terlebih dahulu agar tidak dapat disadap oleh orang lain. Tetapi pada negara seperti Indonesia dan Arab, hal tersebut dilarang oleh pemerintah. Alasan kedua negara tersebut melarang adanya enkripsi terhadap pesan yang dilakukan pada jaringan telepon adalah karena apabila pesan tersebut dienkripsikan oleh penyedia layanan jaringan maka pemerintah tidak dapat melakukan pencarian terhadap adanya tindakan terorisme yang dilakukan pada negara tersebut.

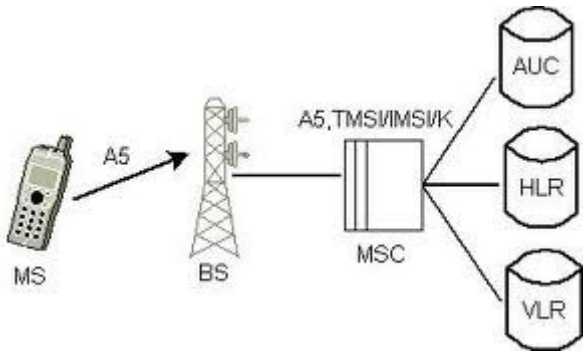
Pada negara seperti Eropa dan Amerika Serikat, enkripsi yang dilakukan adalah *Stream Cipher* A5/2. Sebelum digunakan *Stream Cipher* A5/2, digunakan terlebih dahulu Stream Cipher A5/1 tetapi karena adanya kebocoran informasi mengenai algoritma dari A5/1 tersebut maka algoritma tersebut tidak lagi digunakan. Untuk mengganti A5/1 yang sudah diketahui algoritmanya, maka digunakan A5/2 sebagai penggantinya.

Akan tetapi hal ini tidak berlangsung lama. Karena serangan terhadap enkripsi ini dapat dilakukan sehingga enkripsi ini juga dikategorikan menjadi tidak aman. Untuk mengganti teknik enkripsi yang sudah tidak aman tersebut, maka diciptakan algoritma jenis baru yang bernama A5/3 atau KASUMI. Meski algoritma ini sudah ada, tetapi algoritma ini masih belum diterapkan secara langsung pada jaringan GSM yang ada.

Algoritma A5/1 yang merupakan dasar dari enkripsi pada jaringan GSM ini pertama kali di kriptanalisis oleh Golic. Golic berhasil mendekripsi informasi tersebut dikarenakan adanya gambaran besar mengenai informasi A5/1 yang beredar.

Untuk algoritma A5/2, algoritma ini pertama kali dianalisa oleh Goldberg, Wagner dan Green. Mereka berhasil melakukan kriptanalis terhadap algoritma tersebut dalam waktu yang relatif cepat. Cara kriptanalisis yang digunakan oleh mereka yaitu dengan mendapat 2 buah plainteks dan dengan menggunakan *reverse engineering*.

Adapaun cara lain yang ditawarkan oleh Petrovi'c dan F'uster-Sabater untuk melakukan kriptanalisis terhadap algoritma A5/2 adalah dengan menggunakan sebuah rumus quadratic yang mana variabelnya akan mendeskripsikan mengenai keadaan dari A5/2 tersebut. Dengan menggunakan cara ini, kita tidak akan dapat mendapatkan kunci untuk mendekripsi tersebut, melainkan kita hanya dapat mendekripsi sisa dari informasi yang terdapat pada suatu komunikasi tertentu.



II. KRIPTANALISIS

A. Definisi Kriptanalisis

Kriptanalisi berasal dari bahasa Yunani *kryptos* yang berarti tersembunyi dan *analyein* yang berarti melepaskan. Kriptanalisis merupakan sebuah studi mengenai teknik yang digunakan untuk mendapatkan suatu informasi yang terenkripsi. Informasi yang diperoleh tersebut dapat diperoleh seorang kriptanalis tanpa memiliki kunci untuk mengakses informasi yang dibutuhkan tersebut. Studi mengenai kripanalis ini melibatkan pengetahuan mengenai bagaimana sistem bekerja dan menemukan sebuah kunci rahasia. Seiring dengan berkembangnya ilmu dan teknik untuk melakukan enkripsi, metode untuk melakukan kriptanalisi juga semakin berkembang. Kompleksitas dari metode kriptanalisis berkembang seiring dengan berkembangnya kompleksitas dari metode enkripsi yang ada.

B Metode-metode Kriptanalisis

Adapun beberapa metode yang dapat digunakan oleh kriptanalisis untuk melakukan dekripsi terhadap informasi yang telah berhasil dienkripsi adalah sebagai berikut :

1. Ciphertext-only
   Kriptanalas tersebut hanya memiliki cipherteks yang dapat digunakan untuk mencari informasi yang diinginkan.
2. Known-plaintext
   Kriptanalis memiliki cipherteks dan juga plainteks yang mana merupakan informasi dari cipherteks tersebut.
3. Chosen-plaintext
   Kriptanalis memiliki cipherteks dan juga plainteks sesuai dengan pilihan yang diinginkan oleh kriptanalis.
4. Adaptive chosen-plaintext
   Serupa dengan chosen-plainteks, kecuali penyerang dapat memilih plainteks berdasarkan pada informasi yang didapatkan dari enkripsi sebelumnya.
5. Related-key attack
   Metode yang digunakan pada bagian ini serupa dengan chosen-plainteks. Tetapi pada metode ini, kriptanalis mendapatkan dua hasil cipherteks dari dua kunci yang berbeda sehingga relasi diantaranya dapat diketahui.

C. Sumber Daya dan Usaha

Adapun sumber daya dan usaha yang diperlukan untuk melakukan kriptanalisis terhadap suatu teknik enkripsi dapat dibagi menjadi beberapa jenis, antara lain :

1. Waktu
   Waktu yang dibutuhkan untuk mencari kunci atau lamanya waktu yang dibutuhkan untuk mendapatkan informasi yang diinginkan.
2. Memori
   Kapasistas penyimpanan yang dibutuhkan untuk mencari seluruh kemungkinan yang dapat digunakan untuk mendekripsi informasi tersebut.
3. Data
   Data yang diperlukan untuk melakukan usaha dekripsi tersebut. Misalnya : banyaknya plainteks yang dimiliki yang berkoresponden dengan cipherteks yang dimiliki.

Usaha dan sumber daya yang diperlukan untuk melakukan dekripsi tersebut tidak dapat ditentukan secara pasti. Hal ini disebabkan usaha dan sumber daya tersebut sangat bergantung terhadap hasil analisis dari seorang kriptanalis dan juga kesulitan dari tesknik enkripsi itu sendiri.

III. ALGORITMA A5/2

Algoritma A5/2 merupakan algoritma enkripsi *stream cipher*. *Stream cipher* memiliki arti bahwa enkripsi dilakukan bersamaan dengan dikirimkannya pesan/informasi tersebut. Pada penerapan enkripsi jaringan GSM, pesan/informasi yang dikirimkan merupakan suara dari pengguna pada saat berbicara di telepon. Penerapan rumus umum pada *stream cipher* sebagai berikut :

$$\sigma_{i+1} = F(\sigma_i; x_i; K)$$
$$y_i = f(\sigma_i; x_i; K)$$

Dimana :
K = kunci yang digunakan
x_i = plainteks
y_i = cipherteks
σ_i = keadaan pada saat i
F = fungsi *next-state*
f = fungsi output

Pada algoritma A5/2 ini, digunakan kunci yang memiliki panjang 64-bit dan juga *Initial Value* yang berukuran 22-bit. *Inititial Value* yang digunakan ini bersifat publik sehingga dapat diketahui oleh siapa saja. Pada umumnya, *Initial Value* yang digunakan ini bernama COUNT yang

mana merupakan angke frame.

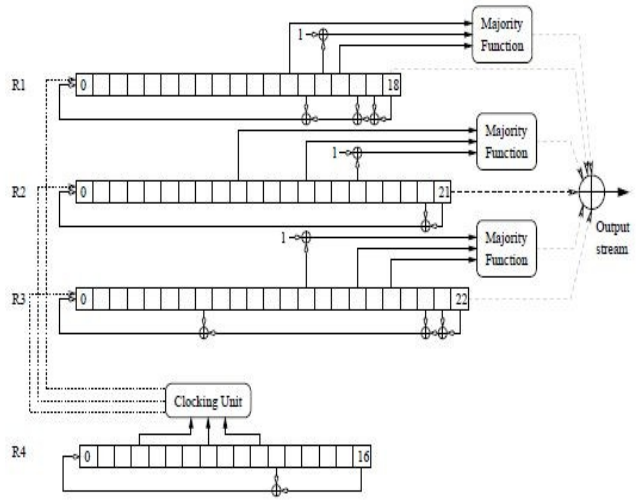


Fig. 1. The Internal Structure of A5/2

Gambar 1. Struktur algoritma A5/2

Seperti yang dapat dilihat pada gambar 1, algoritma A5/2 ini memiliki 4 *shift register*. 4 *shift register* tersebut dilakukan melaui variabel R1, R2, R3 dan R4. Setiap variabel tersebut memiliki panjang bit tersendiri. R1 mempunyai 17-bit, R2 mempunyai 23-bit, R3 mempunyai 23-bit dan R4 mempunyai 17-bit. Sebelum sebuah register mengalami waktu *clocking* yang baru, dilakukan perhitungan terlebih dahulu terhadap data yang dimiliki kemudian disisipkan pada bagian paling kiri sehingga menyebabkan seluruh bit yang ada digeser ke kanan 1 petak.

1. Set $R1 = R2 = R3 = R4 = 0$.
2. For $i = 0$ to 63
   - Clock all four registers.
   - $R1[0] \leftarrow R1[0] \oplus K_c[i]$; $R2[0] \leftarrow R2[0] \oplus K_c[i]$; $R3[0] \leftarrow R3[0] \oplus K_c[i]$; $R4[0] \leftarrow R4[0] \oplus K_c[i]$.
3. For $i = 0$ to 21
   - Clock all four registers.
   - $R1[0] \leftarrow R1[0] \oplus f[i]$; $R2[0] \leftarrow R2[0] \oplus f[i]$; $R3[0] \leftarrow R3[0] \oplus f[i]$; $R4[0] \leftarrow R4[0] \oplus f[i]$.
4. Set the bits $R1[15] \leftarrow 1$, $R2[16] \leftarrow 1$, $R3[18] \leftarrow 1$, $R4[10] \leftarrow 1$.

Fig. 2. The Key Setup of A5/2

Gambar 2. Pengaturan pada algoritma A5/2

Seperti yang dapat kita lihat pada gambar 2, pengaturan awal terhadap kunci dari algoritma ini memiliki 4 tahapan. Tahapan-tahapan tersebut juga menjelaskan mengenai bagaimana shift register tersebut diacak. Pengacakan shift register tersebut berjalan sesuai dengan *clocking* yang digunakan. Pada gambar 2, K(i) merupakan kunci ke i dan f(i) merupakan hasil dari COUNT.

Algoritma A5/2 bekerja secara siklik. Siklik memiliki arti bahwa setiap satu putaran berhasil dilakukan maka akan menghasilkan sebuah bit baru yang akan dikirimkan ke R1,R2 dan R3. Hasil tersebut bergantung dari R4 sehingga R4 juga menjadi penentu dalam menghasilkan enkripsi yang diinginkan.

Pada fungsi yang digunakan untuk menghasilkan keystream generator, digunakan sistem siklik untuk mendapatkan hasil yang diingikan. Tahapan yang dilakukan untuk mendapatkan keystream tersebut sebagai berikut :

1. Melakukan inisiasi seperti pada gambar 2.
2. Mengaktifkan siklik tersebut selama 99 siklik dan membuang hasilnya.
3. Mengaktifkan kembali siklik tersebut selama 228 siklik dan gunakan sebagai keystream untuk enkripsi.

Setelah mendapatkan fungsi keystream yang diinginkan, maka kita melakukan pembagian terhadap bit-bit tersebut menjadi 2 bagian yang sama banyak. Sebanyak 114 bit pertama akan kita gunakan untuk mengenkripsi dari jaringan ke telepon. Sisa 114 bit lainnya akan kita gunakan untuk mengenkripsi jalur telepon ke jaringan.

Perlu juga kita ketahui bahwa algoritma A5/2 dibangun berdasarkan pada arsitektural algoritma A5/1. Fungsi kembalian dari R1, R2 dan R3 mirip dengan fungsi yang dimiliki oleh algoritma A5/1. Proses inisialisasi yang dilkakukan oleh algoritma A5/2 ini juga serupa dengan algoritma A5/1. Hal yang berbeda adalah pada fungsi R4 yang kita gunakan. Pada algoritma A5/2 digunakan fungsi R4 untuk mendapatkan fungsi *clocking* yang akan mempengaruhi fungsi R1, R2 dan R3. Pada algoritma A5/1, *clocking* yang digunakan berasal dari R1, R2 dan R3 tersebut sendiri.

Salah satu penyebab lemahnya algoritma A5/2 adalah karena algoritma A5/2 menggunakan arsitektural dari algoritma A5/1. Kemiripan ini mudah untuk ditemukan oleh para kriptanalis sehingga menyebabkan algoritma A5/2 menjadi sangat lemah.

IV. SERANGAN KNOWN-PLAINTEXT ATTACK PADA ALGORITMA A5/2

Berikut akan dijelaskan mengenai teknik yang akan digunakan untuk mendekripsi algoritma A5/2. Teknik kriptanalisis yang akan dibahas pada makalah ini adalah teknik mendekripsi dengan menggunakan *Known-plaintext attack* dan *Known-ciphertext attack*. Serangan yang

dilakukan ini digunakan untuk mendapatkan kunci session sehingga kita dapat menghasilkan informasi yang kita inginkan.

A. Goldberg, Wagner, and Green's Known Plaintext Attack pada A5/2

Teknik serangan yang diberikan oleh Goldberg, Wagner, dan Green memberikan kita hasil dari observasi mereka bahwa R4[10] akan dipaksa untuk menghasilkan "1" setelah inisialisasi. R4 akan memiliki nilai yang sama setelah inisialisasi dilakukan tanpa memperdulikan apakah hasil dari f[10] tersebut nol atau satu.

Karena R4 mengatur clocking pada R1,R2 dan R3 maka ketiga variabel tersebut bersifat independen terhadap fungsi f[10]. Dengan menggunakan informasi ini maka mereka melakukan perhitungan permutasi terhadap terhadap TDMA (*Time Division Multiple Access*) dan COUNT yang mana merupakan hasil derivasi dari banyak frame yang terdapat pada TDMA maka, diperoleh bahwa dua frame yang memiliki total 26.51 = 1326 TDMA frames (sekitar 6 detik) diperlukan untuk mendapatkan f[10] yang memiliki nilai nol. Penyerang tidak dapat menggunakan f[10] yang memiliki nilai satu karena TDMA akan bertambah secara periodik sehingga menyebabkan angka COUNT tersebut akan selalu berubah.

Melalui informasi tersebut, Goldberg, Wagner dan Green melakukan observasi lebih lanjut bahwa perbedaan dari hasil keluaran tersebut dapat direpresentasikan dalam fungsi linear. Fungsi tersebut dapat direpresentasikan sebagai berikut :

$$g_1(L1_{l1} \cdot R1_1) \text{ XOR } g_1(L1_{l1} \cdot R1_1 \text{ XOR } \delta_1) \text{ XOR } g_2(L2_{l2} \cdot R2_1) \text{ XOR } g_2(L2_{l2} \cdot R1_2 \text{ XOR } \delta_2) \text{ XOR } g_3(L3_{l3} \cdot R3_1) \text{ XOR } g_3(L3_{l3} \cdot R1_3 \text{ XOR } \delta_3) = g_{\delta 1}(L1_{l1} \cdot R1_1) \text{ XOR } g_{\delta 2}(L2_{l2} \cdot R2_1) \text{ XOR } g_{\delta 3}(L3_{l3} \cdot R3_1)$$

Dengan memperoleh fungsi tersebut maka kita dapat mengetahui Kc dengan mencoba R4 dan juga k1 XOR k2. Hasil tersebut dapat diperoleh dengan menyelesaikan fungsi linear tersebut. Dengan menyelesaikan fungsi linear tersebut maka kita akan mendapatkan $R1_1$, $R2_1$ dan juga $R3_1$. Setelah mendapatkan ketiga variabel tersebut maka kita dapat mengembalikannya ke bentuk inisialisasi awal pembentukan kunci tersebut. Dikarenakan R4 merupakan suatu jenis *clocking* yang tidak diketahui maka kita perlu melakukan percobaan terhadap seluruh kemungkinan R4 untuk mendapatkan hasil yang cocok sehingga kita dapat mengambil kembali kunci asalnya.

Salah satu solusi yang lebih cepat dapat diperoleh yaitu dengan memfiltrasi nilai dari R4. Cara filtrasi dari R4 tersebut dapat dilakukan dengan menggunakan eliminasi Gauss.

Meskipun untuk melakukan serangan seperti yang dijelaskan oleh Goldberg, Wagner, and Green membutuhkan proses pendek yang relatif banyak untuk menyelesaikan persamaan tersebut tetapi hasil tersebut dapat kita peroleh dengan melakukan pemrosesan yang terdapat pada suatu komputer pribadi selama beberapa menit saja.

B. Ciphertext-only Attack pada A5/2

Salah satu teknik serangan yang dapat digunakan untuk mendekripsi algoritma A5/2 adalah dengan menggunakan metode *Ciphertext-only attack*.

Pada suatu jaringan GSM, pengkoreksian terhadap kesalahan yang mungkin terjadi. Hanya saja, ketika transmisi dilakukan suatu pesan terlebih dahulu disisipkan dengan kode untuk memperbaik kesalahan yang ada sehingga menyebabkan ukuran dari pesan tersebut relatif menjadi lebih besar. Pada saat itulah pesan tersebut dienkripsi dan dikirimkan.

Jalur transmisi yang digunakan tersebut bertentangan dengan prinsip enkripsi yang digunakan. Pada prinsip yang digunakan, seharusnya pesan tersebut dienkripsi terlebih dahulu kemudian baru disisipkan kode untuk memperbaiki kesalahan.

Melalui jenis enkripsi inilah dapat kita gunakan metode *ciphertext-only attack*. Serangan dapat kita lakukan terhadap kode perbaikan yang terdapat pada setiap pesan yang dikirimkan.

Operasi kode dan operasi pesan tersebut dapat dimodelkan dalam bentuk perkalian pesan (direpresentasikan sebagai 184-bit vector biner dengan lambang P) dengan ukuran matriks 456x184 yang mana diberikan lambing G dan dilakukan operasi XOR terhadap sebuah vector konstan yang dilambangkan dengan g. Hasil dari vector tersebut adalah :

$$M = (G.P) \text{ XOR } g$$

Vector M tersebut kemudian dibagi menjadi 4 data frame yang berbeda. Pada proses enkripsi yang diperlukan, setiap data frame dilakukan operasi XOR terhadap hasil keluaran *keystream* dari A5/2 tersebut.

Kunci observasi yang perlu diperhatikan pada cipherteks yang dimiliki adalah dengan mencari persamaan linear pada *keystream* bit tersebut. Keystream bit tersebut, yang dapat dilambangkan dengan k, merupakan hasil gabungan dari keempat frame keystream yang dimiliki dimana k = k1 || k2 || k3 || k4 (dimana || mengartikan bahwa konkatenasi). Hasil observasi tersebut dapat

menghasilkan sebuah operasi yang mirip dengan C XOR g, yang mana memiliki bentuk seperti berikut :

$$H \cdot (C \text{ XOR } g) = H \cdot (M \text{ XOR } k \text{ XOR } g) = H \cdot (M \text{ XOR } g) \text{ XOR } H \cdot k = 0 \text{ XOR } H \cdot k = H \cdot k.$$

Variabel H merupakan suatu variabel yang digunakan untuk melakukan parity check. Karena kita telah mengetahui cipherteks dan g merupakan suatu nilai yang konstan dan diketahui maka kita mempunyai suatu persamaan yang akan dicari bit k yang digunakan.

Dari rumus persamaan linear tersebut, kita dapat melihat bahwa rumus tersebut bersifat independen terhadap variabel P yang telah kita definisikan sebelumnya. Variabel yang mempengaruhi persamaan linear tersebut hanyalah variabel k yang mana merupakan keystream yang digunakan.

Setelah persamaan tersebut dilakukan substitusi, maka kita hanya perlu melakukan uji coba terhadap kunci *keystream* k tersebut. Untuk memperkecil kemungkinan yang akan dicari maka kita dapat menggunakan Gauss elimination untuk mendapatkan hasil tersebut. Hasil akhir dari kunci tersebut dapat diperoleh dengan melakukan inversi terhadap standar inisialisasi awal yang terdapat pada penjelasan proses sebelumnya.

C. Perbedaan Serangan dengan Menggunakan Known Plaintext-only Attack dan Ciphertext-only Attack

Perbedaan utama dari serangan dengan menggunakan known plaintext-only attack dan ciphertext-only attack terdapat pada *keystream* bit yang digunakan. Pada known plaintext-only attack keystream bit tersebut dapat diperoleh dengan mencocokkan waktu terhadap TDMA yang sedang berjalan. Sedangkan pada ciphertext-only attack, keystream tersebut tidak diketahui, yang kita ketahui hanyalah nilai dari persamaan garis dari *keystream* tersebut. Persamaan tersebut dapat kita peroleh melalui cipherteks yang ada dan juga melalui kode perbaikan yang disisipkan dalam setiap pesan yang ada.

Perbedaan lain yang terdapat pada known plaintext-only attack dan ciphertext-only attack terdapat pada pencarian yang dilakukan. Pada known plaintext-only attack, banyaknya frame plainteks yang diperlukan untk menghasilkan persamaan yang dapat merepresentasikan keystream enkripsi tersebut. Sedangkan pada ciphertext-only attack, dibutuhkan paling tidak 8 frame cipherteks untuk dapat menghasilkan persamaan yang dapat merpresentasikan keystream enkripsi tersebut. Hal tersebut disebabkan karena pada ciphertext-only attack, dari 456 bit cipherteks yang ada, kita hanya dapat menarik 272 persamaan. Konsekuensi yang diperoleh dengan menggunakan 8 frame dibandingkan dengan 4 frame adalah optimasi yang dapat dilakukan sehingga konstrain yang terdapat pada perbedaan XOR pada frame yang dimiliki semakin kuat. Dengan semakin pastinya konstrain yang kita miliki, maka kita dapat mengetahui bagaimana kita perlu mencari kemungkinan yang lainnya.

V. PENGEMBANGAN ALGORITMA YANG DAPAT DILAKUKAN

Algoritma A5/2 ini dapat dikembangkan agar serangan-serangan tersebut tidak dapat dilakukan lagi terhadap algoritma tersebut. Salah satu pengembangan yang dilakukan adalah dengan mengacak penggunaan keystream tersebut. Pengacakan keystream tersebut dapat dilakukan dengan menggunakan sebuah fungsi permutasi seperti yang dilakukan pada DES. Pengacakan tersebut dilakukan dengan sebuah fungsi permutasi yang telah dikembangkan dan akan berubah setiap kali clocking tersebut dilakukan. Dengan menggunakan fungsi ini, maka serangan untuk mencari persamaan dari cipherteks ataupun plainteks tidak dapat dilakukan.

```
PROCEDURE

Byte[] PermutationShuffle(input Keystream : Byte[])
{This procedure is used to reshuffle the keystream
according to algorithm defined in this procedure. The
result given will be used to do the encryption in the main
algorithm}
```

VI. KESIMPULAN

Pengembangan algoritma A5/2 sangatlah lemah karena dibangun di atas arsitektural algoritma A5/1. Hal ini disebabkan karena pengembangan algoritma A5/2 disebabkan oleh berhasil dipecahkannya cara enkripsi dari algoritma A5/1. Dengan pengembangan algoritma A5/2 di atas arsitektur algoritma A5/1 maka kelemahan dari A5/2 tersebut akan semakin jelas dan dapat dieksploitasi oleh para kriptanalis. Penggunaan *ciphertext-only attack* seperti yang dibahas tersebut dapat dilakukan karena kita mengetahui bahwa setiap pesan tersebut disisipkan kode untuk melakukan koreksi kemudian baru dilakukan enkripsi sehingga setiap pesan yang dikirimkan tersebut secara tidak langsung mengandung informasi yang sama. Melalui informasi tersebut maka kita dapat melakukan serangan terhadap algoritma tersebut secara langsung.

Dalam pengembangan algoritma enkripsi yang digunakan, kita perlu menerapakan prinsip *diffusion* dan *confusion* dari Shannon untuk memperoleh hasil yang lebih efektif dibandingkan hanya dengan menggunakan pergeseran bit. Pada stream cipher kita juga dapat menerapkan *iterative encryption* agar hasil yang kita dapatkan lebih kuat enkripsinya.

Permasalahan yang ada pada *stream cipher* adalah diperlukannya sistem enkripsi yang cepat karena data yang diberikan berjalan secara terus menerus. Permasalahan lain yang terdapat pada stream cipher adalah pengiriman kunci yang digunakan. Karena sifatnya aktif maka kunci harus dapat dihasilkan secara random setiap kali melakukan hubungan komunikasi. Akan tetapi masalah yang terjadi ialah penerima atau penelepon tidak mengetahui kunci apa yang perlu digunakan untuk melakukan dekripsi tersebut. Pengiriman kunci melalui jaringan memiliki resiko dibaca oleh orang lain. Hal tersebut masih menjadi permasalahan yang belum dapat dipecahkan dan dicari solusinya.

Pengembangan enkripsi stream cipher terutama pada jaringan telepon GSM akan sangat membantu untuk menjaga privasi setiap orang dalam berbicara ataupun menyebarkan informasi melalui jaringan GSM sehingga masih perlu dilakukan pengembangan terhadap algoritma yang mangkus untuk mengenkripsi pesan/informasi yang dimiliki.

REFERENSI

[1] http://www.cs.technion.ac.il/users/wwwb/cgi-bin/tr-get.cgi/2006/CS/CS-2006-07.pdf. Accessed 1 March 2011.

[2] http://en.wikipedia.org/wiki/Stream_cipher. Accessed 1 March 2011.

[3] http://en.wikipedia.org/wiki/Known-plaintext_attack. Accessed 1 March 2011.

[4] http://www.cert-ist.com/eng/ressources/Publications_ArticlesBulletins/Veilletechnologique/200912_gsmcracking/. Accessed 1 March 2011.

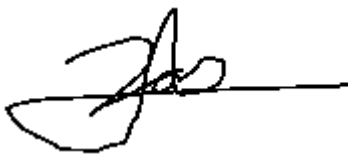
[5] http://en.wikipedia.org/wiki/A5/2. Accessed 1 March 2011.

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

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, 21 Maret 2011



Darwin - 13508102