STUDI DAN ANALISIS ALGORITMA KRIPTOGRAFI PADA JARINGAN SELULER

Andri Rizki Aminulloh – NIM: 13506033

Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail : if16033@students.if.itb.ac.id

Abstrak *- Perkembangan zaman dan ekonomi menuntut manusia untuk melakukan kegiatannya dengan cepat dan tepat. Tuntutan ini membawa manusia pada era pertukaran informasi yang*

Telepon genggam adalah salah satu alat yang paling menunjang tuntutan hidup zaman modern. Kini penggunaan telepon genggam sudah menjadi gaya hidup, bahkan sudah mulai bergeser menjadi sebuah kebutuhan. Banyak hal yang dapat dilakukan telepon genggam untuk membantu manusia mempercepat pekerjaannya. Apalagi teknologi seluler generasi ketiga memungkinkan penggunanya untuk mengakses internet dimana saja dan kapan saja, dan yang membuatnya lebih menakjubkan adalah kecepatan aliran data melalui jaringan seluler saat ini lebih cepat daripada melalui jaringan kabel.

Dalam makalah ini kan dibahas 3 algoritma kriptografi dalam teknologi seluler yaitu A5/1,A5/2, dan A5/3(KASUMI). Penulis akan membandingkan kelebihan dan kekurangan masing-masing algoritma dan membahas serangan-serangan yang telah dan mungkin dapat dilakukan terhadap ketiga algoritma diatas. Penulis juga akan menganalisis penyebab kebocoran algoritma A5/1, A5/2.

Kata Kunci : *algoritma A5, GSM, jaringan seluler*

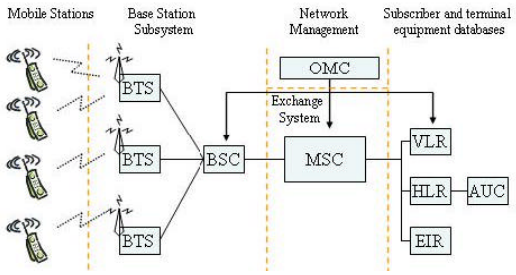
1. PENDAHULUAN

Teknologi seluler semkin popular di kalangan masyarakat kita. Seiring perkembangannya, penggunaannya pun semakin luas dengan pesat. Jumlah perusahaan penyedia jasa layanan seluler pun semakin menjamur di Indonesia, baik perusahaan local maupun perusahaan asing. Jenis layanannya pun semakin beragam, mulai dari jasa telepon, SMS (*short message service*), internet, video call, dsb.

Saat ini terdapat dua sistem jaringan seluler yang paling populer dan digunakn luas oleh masyarakat, yaitu GSM (Global Sistem on Mobile Communication) dan CDMA (Code Division Multiple Access).

Jaringan seluler adalah jaringan yang penggunaannya tidak memakai kabel, oleh karena itu jaringan ini sangat rentan akan serangan, jauh lebih rentan daripada jaringan telepon biasa. Hal ini menimbulkan kebutuhan akan sistem keamanan yang kuat untuk menjaga kerahasiaan data yg mengalir di udara. Beberapa langkah telah dilakukan untuk menjaga kerahasiaan percakapan maupun data yang dimiliki oleh pengguna telepon genggam dan salah satunya adalah dengan mengembangkan algoritma enkripsi untuk setiap data yang dikirim.

Berikut adalah arsitektur jaringan GSM



Gambar 1 Arsitektur GSM

Dari gambar di atas, dapat dilihat bahwa dalam berkomunikasi, telepon seluler, *mobile station* (MS), memanfaatkan layanan jaringan melalui *base station subsystem* yang terdiri dari beberapa *base transceiver station (BTS) dan sebuah base station controller (BSC).*

Pada sistem GSM terdapat 3 langkah pengamanan yang dilakukan:

- Otentikasi pengguna layanan seluler – berhubungan dengan kemampuan telepon genggam untuk membatasi akses hanya pada akun tertentu di sebuah operator.
- Keamanan pengiriman data – keamanan terhadap serangan –serangan yang dilakukan untuk menerima data milik pengguna secara illegal, dalam hal ini bagaimana menyandikan data dan sinyal yang dikirim.
- Kerahasiaan identitas pengguna – terkait kerahasiaan informasi mengenai pengirim sinyal sehingga tidak semua orang yang

berusaha menangkap sinyal ini dapat mengetahui identitas atau informasi mengenai si pengirim. Untuk isu keamanan ini digunakan IMSI (International Mobile Subscriber Identity) yang unik untuk setiap pengguna layanan telepon seluler

Otentikasi pengguna layanan.
Otentikasi pengguna dibutuhkan untuk mencegah pengguna yang tidak berhak memasuki jaringan dan mengklaim bahwa ia adalah *subscriber*. Jika hal ini terjadi, maka dengan mudahnya ada kemungkinan untuk membajak *account* seseorang dan berkedok seolah-olah ia adalah *account* tersebut.

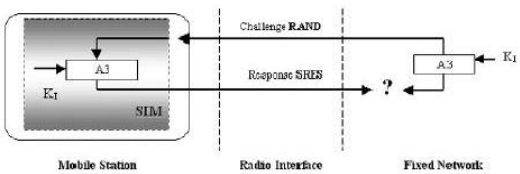
Otentikasi pengguna dilakukan agar hanya pengguna yang terdaftar dan berhak saja yang dapat menggunakan layanan operator jaringan. Selain itu, digunakan agar tagihan dikenakan pada pengguna yang tepat, yang memang memanfaatkan layanan jaringan.

Algoritma yang digunakan dalam proses otentikasi pengguna layanan pada jaringan
GSM adalah algoritma A3. Algoritma ini tidak bersifat publik sehingga hanya antarmuka eksternalnya saja yang dispesifikasikan dalam GSM. Keamanan algoritma ini tergantung pada kunci rahasia *user Ki* yang beririsan antara *mobile phone* dan jaringan GSM. GSM sendiri tidak menspesifikasikan panjang nilai *Ki* sehingga penentuan panjang nilai *Ki* biasanya diserahkan sepenuhnya kepada pihak operator masing-masing. Namun, biasanya panjang kunci yang biasa digunakan oleh operator adalah 128 bit.Adapun proses otentikasi pengguna menggunakan algoritma A3 adalah sebagai berikut :

a. Jaringan mengirim tantangan acak (RAND) kepada MS.
b. MS melakukan enkripsi terhadap RAND dengan algoritma A3 dan kunci Ki, menghasilkan respon SRES. Proses ini dilakukan pula oleh jaringan.
c. MS mengirim SRES kepada jaringan.
d. Jaringan membandingkan SRES

yang dihasilkannya dengan SRES yang diterima. Jika cocok, otentikasi berhasil.

Skema otentikasi pengguna dapat dilihat pada gambar di bawah.



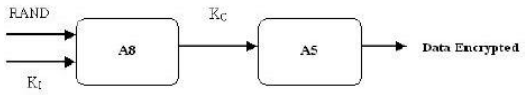
Gambar 2 Skema Otentikasi Pengguna

Dapat dilihat bahwa tidak ada pengiriman Ki karena mensubmit Ki ke jaringan tidaklah aman.Karena hal itulah proses otentikasi tidak dilakukan dengan membandingkan nilai Ki di SIM dan Auc, tapi dengan prosedur otentikasi yang telah dijelaskan. Jika otentikasi gagal untuk pertama kalinya dan TMSI telah digunakan, maka jaringan akan memilih untuk kembali mengulangi proses otentikasi menggunakan IMSI.

Enkripsi Data

Mekanisme enkripsi data adalah sebagai berikut:

1. Memproses RAND, yang diterima pada saat akan melakukan otentikasi pengguna, dengan algoritma A8 dan Ki untuk menghasilkan kunci enkripsi Kc (*ciphering key*).
2. Mengenkripsi *plaintext* dengan algoritma A5 dan kunci Kc untuk menghasilkan *ciphertext*, yang akan ditransmisikan melalui jaringan. Skema untuk enkripsi dapat dilihat pada gambar berikut :



Gambar 6 Skema Enkripsi dalam GSM

Ketika algoritma A3 dijalankan pada proses otentikasi pengguna, sebenarnya pada saat yang sama algoritma A8 juga dijalankan.

Pada makalah ini hanya akan dibahas algoritma A5.

2. ALGORITMA CHIPER A5

Algoritma chiper A5 adalah algoritma enkripsi yang digunakan untuk mengamankan komunikasi dan aliran data pada system GSM pada jarinagn seluler. Algoritma ini sudah dikenal memiliki banyak kelemahan dan tidak jarang serangan terhadapnya berhasil dilakukan.

Algoritma ini pertama kali dikembangkan pada tahun 1987 ketika penggunaan GSM hanya terbatas di wilayah Eropa. Pada saat itu algoritma ini bernama A5/1. Kemudian pengembangan terus dilakukan untuk meningkatkan keamanan algoritma hingga pada tahun 1989 A5/2 berhasil dikembangkan. Kedua algoritma ini pada saat itu sangat dirahasiakan oleh pihak-pihak yang mengembangkannya dan oleh penyedia layanan GSM sampai pada akhirnya informasi mengenai algoritma ini dapat diketahui oleh masyarakat umum pada tahun 1994 dan berhasil di *reverse*

engineered oleh Marc Briceno pada tahun 1999 melalui sebuah telepon genggam biasa.

Walaupun demikian algoritma A5/1 akhirnya tetap digunakan untuk mengamankan komunikasi suara pada tahun 2000 ketika layanan telepon selular mulai berkembang pesat dan penggunaannya semakin luas. Sampai saat ini pengembangannya sudah sampai pada generasi ketiga yaitu algoritma A5/3.

Berbeda dengan algoritma A5/1 dan A5/2, algoritma A5/3 merupakan algoritma *block chipper* yang diadaptasi dari algoritma KASUMI.

KASUMI dibuat oleh Secure Algorithms Groups of Experts (SAGE, bagian dari ETSI, sebuah badan standar di Eropa. KASUMI sendiri adalah pengembangan dari algoritma MISTY1.

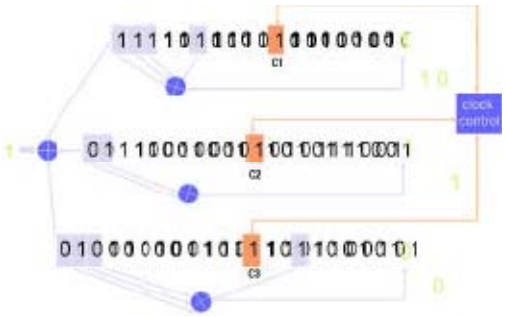
Algoritma A5/1

Transaksi data pada sitem GSM diatur sebagai *sequence of burst*. Sebuah burst dikirim melalui sebuah jalur dan searah setiap 4.615 milisekon dan mengandung informasi sebesar 114 bits. Algoritma A5/1 digunakan untuk menenkripsi setiap burst dengan membuat sebuah *keystream* berukuran 114 bit yang kemudian di XOR dengan tiap *burst*.

A5/1 terdiri dari 3 buah LFSR (Linear Feedback Shift Register) dengan irregular clocking dan aturan sebagai berikut

Panjang dalam bit	Fungsi feedback	Bit yg diberi tap	Clocking Bit
19	X18+x17+x16+x13+1	13,16,17,18	8
22	X21+x20+1	20,21	10
23	X22+x21+x20+x7+1	7,20,21,22	0

Bit-bit yang dibentuk diberi indeks dengan LSB 0.



Gambar 3 Algoritma A5/1

Masing-masing register memiliki aturan clocking sendiri. Sebuah register akan di clock jika bit tengahnya sama dengan majority bit dari bit tengah tiap-tiap register. Bit mayoritas akan bernilai "1" jika dan hanya jika 2 atau lebih bit tengah bernilai "1", jika tidak maka akan bernilai "0".

Bit tengah R1 = bit 8, kita misalkan x_1
Bit tengah R2 = bit 10, kita misalkan x_2
Bit tengah f R3 = bit 10, kita misalkan x_3

kita misalkan y_1 sebagai kondisi clocking R1
kita misalkan y_2 sebagai kondisi clocking R2
kita misalkan y_3 sebagai kondisi clocking R3
jika y_1 ="1" maka R1 di clock, dan seterusnya untuk R2 dan R3.

Proses tersebut dapat digambarkan dengan table karnaugh-map berikut

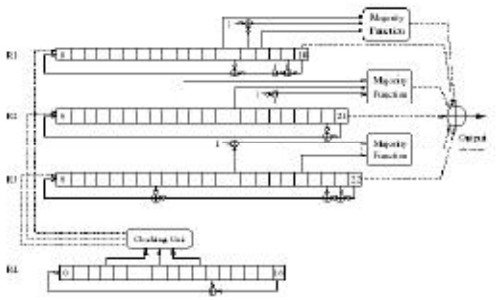
Middle Bit			Majority	Clock		
R1	R2	R3		R1	R2	R3
0	0	0	0	1	1	1
0	0	1	0	1	1	0
0	1	0	0	1	0	1
0	1	1	1	0	1	1
1	0	0	0	0	1	1
1	0	1	1	1	0	1
1	1	0	1	1	1	0
1	1	1	1	1	1	1

Gambar 4 Tabel Clocking

Algoritma A5/2

Algoritma A5/2 terdiri dari empat LFSR (Linear Feedback Shift Register) dengan panjang maksimum yakni : R1, R2, R3, dan R4. Register-register tersebut memiliki panjang 19 bit, 22 bit, 23 bit, dan 17 bit. Setiap register memiliki *tap* dan fungsi *feedback* dan untuk ploynomial tiap-tiap register adalah x19 + x5 + x2 + x +1, x22 + x +1, x23 + x15 + x2 + x +1, dan x17 + x5 +1.

Untuk representasi register-register tersebut digunakan notasi [2, 4, 5, 17] dimana bit-bit dalam register yang terurut secara terbalik berkorespondensi dengan sebuah *tap* dengan indeks *len-i*-1, dimana *len* adalah ukuran register. Contoh: ketika R4 dikunci berdasarkan mekanisme penguncian (*clocking*), nilai XOR R4[17-0-1=16] dan R4[17-5-1=11] dihitung, baru kemudian registernya digeser satu bit ke kanan dan nilai hasil XOR tersebut ditempatkan di R4.



Gambar 5 Algoritma A5/2

Pada algoritma A5/2, R1, R2, dan R3 dikunci dilakukan berdasarkan mekanisme penguncian (*clocking*) dengan aturan seperti yang dijelaskan pada gambar yakni R4 mengontrol penguncian (*clocking*) R1, R2, dan R3. Ketika penguncia terhadap R1, R2, danR3 dilakukan, bit-bit R4[3], R4[7], dan R4[10] merupakan input dari unit penguncian. Unit pengujian ini melakukan sebuah fungsi mayoritas pada bit-bit yang ada. R1 dikunci jika dan hanya jika R4[10] sesuai dengan mayoritas. R2 dikunci jika dan hanya jika R4[3] sesuai dengan mayoritas. R3 dikunsi jika dan hanya jika R4[7] sesuai dengan mayoritas. Setelah penguncianpenguncian terhadap register R1, R2, dan R3 dilakukan, baru kemudian R4 dikunci.

Setelah proses penguncian dilakukan, satu bit output sudah siap untuk dihasilkan pada A5/2. bit output merupakan fungsi non-linier dari status internal R1, R2, dan R3. Setelah dilakukan inisialisasi 99 bit output dibuang dan 228 bit berikutnya digunakan sebagai *output key-stream*. Adapun proses inisialisasi status internal dilakukan sebagai berikut:

- ubah nilai seluruh LFSRs dengan nilai 0

- for 1:=0 to 63 do
1. kunci seluruh LFSR
2. R1[0]¬R1[0] Å *Kc[i]*
3. R2[0]¬R2[0] Å *Kc[i]*
4. R3[0]¬R3[0] Å *Kc[i]*
5. R4[0]¬R4[0] Å *Kc[i]*

- for i:=0 to 21 do
1. kunci seluruh LFSR
2. R1[0]¬R1[0] Å *f[i]*
3. R2[0]¬R2[0] Å *f[i]*
4. R3[0]¬R3[0] Å *f[i]*
5. R4[0]¬R4[0] Å *f[i]*

Dimana nilai *i* menunjukkan bit ke-*i* dari *session key Kc*[i] dengan panjang 64 bit, bit ke-*i* dari register dari register *Rj*[*i*], dan bit ke-*i* dari jumlah *frame* yang bersifat public *f*[*i*].

Sedangkan proses pembangkitan *key-stream* adalah:

1. inisialisasi status internal dengannilai *Kc* dan jumlah *frame*
2. Isikan nilai bit-bit R1[15], R2[16],R3[8], dan R4[10] dengan 1
3. jalankan algoritma A5/2 untuk 99 *clocks* dan abaikan outputnya
4. Jalankan algoritma A5/2 untuk 228 *clocks* berikutnya dan gunakan aoutputnya sebagai *key-stream*

3. SERANGAN TERHADAP ALGORITMA A5

Sejumlah serangan terhadap algoritma A5/1 sudah berhasil dilakukan.dan dipublikasikan. Beberapa serangan tersebut memerlukan proses yang memakan biaya yang cukup besar dengan waktu yang dibutuhkan hanya dalam hitungan menit atau detik.

Kelemahan algoritma A5/1 dapat dieksploitasi dengan menggunakan serangan known-plaintext secara pasif. Kelemahan yang lebih seirus dapat dieksploitasi dengan skenario chipertext-only

Serangan Known Plaintext

Pada tahun 1997, Golic memperlihatkan sebuah serangan yang berbasis himpunan penyelesaian dari persamaan linear yang memiliki kompleksitas waktu $\mathbf{2}^{40.16}$ (unit yang digunakan berdasar pada solusi persamaan linear yang dibutuhkan). Penyelesaian himpunan dari persamaan linear yang memiliki waktu.

Pada tahun 2000, Alex Biryukov, Adi Shamir dan David Wagner menunjukkan bahwa A5/1 dapat dikriptanalis secara real time menggunakan *time memory tradeoff attack*. Sebuah tradeoff memungkinkan seorang penyerang untuk merekonstruksi kunci dalam beberapa menit dari 2 detik percakapan (plainteks yang diketahui), namun si penyerang harus melalukan terlebih dahulu preprocessing stage yang membutuhkan data 300 GB dan 2^{48} waktu komputas. Beberapa tradeoff dari preprocessing, data requirement, kompleksitas memori dan waktu serangan dimungkinkan..

Chipertext-only Scenario

Serangan pada algoritma A5 ini juga dapat dilakukan dengan metode serangan pada cipherteks. Namun hal ini harus dibarengi dengan data statistik plainteks untuk meningkatkan kemungkinan sukses serangan.

Apabila statistik yang dimiliki cukup bagus, maka serangan jenis ini dapat dilakukan dengan panjang

cipherteks hanya beberapa ribu bytes. Untuk melakukan serangan pada cipherteks, dapat dimulai dengan mengidentifikasi kemungkinan kata berisi tujuh karakter yang mungkin muncul pada plainteks. Contoh kata yang mungkin adalah "login: ", dan banyak kata lain. Kemudian string yang mungkin tersebut kemudian digeser – geser posisinya pada cipherteks untuk menemukan padanannya dengan pesan teks yang benar.

Apabila plainteks yang mungkin tadi diletakkan secara benar, maka kemudian didapatkan potongan keystream dengan panjang sama dengan panjang plainteks tersebut. Kemudian cara known – plainteks diatas dapat diterapkan pada potongan keystream ini.

Apabila setiap jalur tebakan tidak ada yang benar pada N=7 byte dari teks yang diketahui, maka tampaknya peletakan plainteks yang mungkin diatas salah. Apabila sebaliknya, dapat disimpulkan bahwa telah ditemukan padanan yang valid. Ini tampaknya terlalu optimistis, namun kemudian akan ditunjukkan bahwa kemungkinan error sebenarnya cukup kecil. Dengan prosedur ini, padanan yang salah seharusnya cukup jarang, sebab jalur – jalur yang salah pada tiap tebakan seharusnya cukup dangkal. Setelah menganalisa byte pertama, ada 216 kemungkinan byte tinggi dari tiap register LFSR.

Setelah menganalisis byte kedua, maka kemungkinan terjadinya kesalahan yang tidak terdeteksi adalah hanya 0,0459. setelah byte ketiga, juga ada 0,0459 kemungkinan kesalahan. Ini berarti hanya 216 x (0,0459)6 atau 0,00061 peluang kesalahan dapat terjadi setelah pengecekan byte ketujuh.

Oleh karena itu, dengan lebih dari seribu byte cipherteks, dapat diharapkan bahwa tidak mungkin ada kesalahan pemadanan untuk kata yang mungkin sepanjang minimal tujuh karakter.

Penggunaan kata yang lebih panjang akan lebih meminimalisasi kemungkinan terjadinya kesalahan. Untuk mengecek posisi cipherteks untuk padanan yang mungkin dibutuhkan 216 percobaan, dengan panjang cipherteks yang kurang dari seribu byte, maka total usaha komputasi yang digunakan untuk mengetes kata yang mungkin kuranga dari 226, dan pengecekan terhadap kamus kata – kata yang mungkin menjadi mudah.

Sepanjang dimiliki panjang cipherteks dan dapat mengidentifikasi himpunan kata – kata yang mungkin, seharusnya mudah untuk menemukan dua atau tiga padanan dan kemudian merecover seluruh kunci algoritma A5 ini. Dengan kata lain, serangan cipherteks pada algoritma A5 memiliki tingkat kompleksitas yang rendah, ketika beberapa pengetahuan dari statistic plainteks diketahui.

Komputasi yang dibutuhkan tidak terlalu banyak dan jumlah cipherteks yang dibutuhkan juga tidak terlalu panjang. Sehingga serangan – serangan sejenis ini tampak sangat mungkin.

4. ANALISIS

Para pembuat algoritma A5 sedemikian rupa mencoba untuk merahasiakannya. Seprtinya langkah ini dilakukan untuk menghindari serangan-serangan terhadap algoritma tersebut. Namun yang terjadi adalah pada akhirnya algoritma-algoritma A5 tersebut diketahui secara luad bahakan beberapa orang berhasil melakukan reverse Engineering terhadap algoritma tersebut. Daripada mencoba memperbaiki performa algoritma A5, para pembuat algoritma ini lebih sibuk untuk menjaganya agar tetap rahasia. Startegi seperti ini sia-sia terbukti dari kondisi yang terjadi saat ini. Algoritma A5 yang digunakan untuk system GSM saat ini sangat rentan terhadap serangan, pembicaraan melalui jaringan seluler dengan mudah disadap.

Pengembangan lebih lanjut algoritma A5 yang didasarkan pada algoritma A5/1 dan A5/2 masih dapat dilakukan. Dengan membuat proses enkripsi menjadi lebih kompleks, algoritma A5 masih dapat diselamatkan.
Hal ini sudah mulai direalisasikan dengan pembuatan algoritma A5/3 yang berbasis block chipper. Namun lagi-lagi para pembuat lebih sibuk merahasiakannya ketimbang berusaha semaksimal mungkin mengembangkannya dan segera menggunakannya pada proses enkripsi pada system GSM, karena sejak pengembangannya algoritma ini masih belum digunakan.

Ide dasar dari algoritma A5/1 dan A5/2 sebenarnya sudah cukup baik, mengingat enkripsi dilakukan di level bit. Untuk pengembangan kedepannya akan sangat baik jika diimplementasikan konsep-konsep dari block chipper. Hal ini mungkin sudah diimplementasikan pada pengembangan algoritma A5/3. Tetapi yang mungkin menjadi sedikit kekurangan adalah prosedur yang memaksa algoritma untuk membentuk sebuah blok terlebih dahulu. Akan menjadi lebih baik jika konsep stream chipper tetap dipertahankan untuk efektifitas dan efisiensi algoritma A5, mengingat penggunaannya pada system GSM yang membutuhkan aliran data yang kontinu. Dengan mempertahankan prinsip-prinsip stream chipper diharapkan delay yang terjadi pada saat pengiriman data atau sinyal dapat diminimalisir.

Ide lain yang dapat diimplementasikan adalah dengan menggunkan kunci dinamis. Maksud dari kunci dinamis disini adalah kunci dibentuk dari sebuah informasi yang selalu berubah dan tidak mungkin berulang. Kunci ini digunakan digunakan bersama dengan kunci lain yang sudah diketahui. Dengan penggunaan kunci dinamis ini digabung dengan pengguunaan konsep jaringan feistel pada block chipper, diharapkan serangan terhadap jaringan GSM akan lebih sulit untuk dilakukan.

5. KESIMPULAN

Dari pembahasan yang dilakukan di atas mengenai algoritma A5 dalam penggunannya pada system GSM, maka dapat disimpulkan bahwa

1. Belum ditemukan algoritma yang aman untuk melakukan enkripsi pada jaringan seluler khususnya sistem GSM.
2. Para pengembang algoritma A5 terlalu sibuk merahasiakan algoritmanya ketimbang mengembangkan algoritma yang lebih baik
3. Algoritma A5/3 belum dapat digunkan secara luas
4. Jaringan GSM sangat mudah disadap sehingga informasi yang dikirimkan melalui jaringan ini menjadi kurang terpercaya

DAFTAR PUSTAKA

1. E. Dawson and L. Nielsen. Automated cryptanalysis of XOR plaintext strings.*Cryptologia*, volume XX Number 2, pages165{181. April 1996.
2. B. Goldburg, E. Dawson and S. Sridharan.The automated cryptanalysis of analogspeech scramblers. *Advances in Cryptology EUROCRYPT'91*, volume 547 of*Lecture Notes in Computer Science*, pages422{430. Springer-Verlag, 1991.
3. M. Briceno, I. Goldberg and D. Wagner. GSM cloning. 20 April, 1998. http://www.isaac.cs.berkeley.edu/isaac/gsm.ht
4. J. Dj. Goli_c. Cryptanalysis of alleged A5 stream cipher. *Advances in Cryptology - EUROCRYPT'97*, volume 1233 of *Lecture Notes in Computer Science*, pages 239{255. Springer-Verlag, 1997.
5. http://www.gsm-security.net/gsm-security-papers.shtml, diakses tanggal 25 Maret 2009
6. http://en.wikipedia.org/wiki/A5/1 diakses 25 Maret 2009
7. Barkan, Elad; Biham, Eli; Keller, Nathan. *Instant Ciphertext-Only Cryptanalysis of GSM Encrypted Communication.*