Sistem Keamanan SMS (Short Message Service) pada Jaringan Selular dengan Peningkatan Fungsionalitas Menggunakan Internet

Mukhamad Ikhsan – 13503033
Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl Ganesha 10, Bandung
E-mail : Ikhsan_only@yahoo.co.uk

Abstraksi :

Jaringan selular adalah sebuah komponen yang sangat penting dalam perekonomian dan kehidupan sosial saat ini. Selain layanan suara, pesan teks merupakan layanan yang sangat sering digunakan oleh pelanggan. Selain itu seiring dengan perkembangan jaman, banyak perusahaan telekomunikasi yang memperluas layanan SMS-nya selain dapat diakses pada jaringan internalnya yang merupakan jaringan selular tetapi juga memanfaatkan jaringan internet, sehingga para pengguna dapat menggunakan antar muka internet untuk menggunakan layanan SMS.

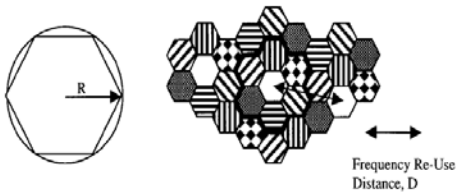
Secara umum SMS tidak menjamin kerahasiaan dan keutuhan pesan yang dikirimkan oleh pengguna. Oleh karena pesan-pesan teks yang dikirim pengguna terkadang merupakan pesan yang rahasia dan pribadi, sehingga kerahasiaan pesan menjadi sangat penting untuk dijaga dari orang-orang yang tidak berhak mendapatkannya. Sehingga dibutuhkan suatu sistem keamanan dalam menyampaikan pesan tersebut. Terlebih karena keterbatasan memori dan juga karakter yang berbeda dalam penyampaian data melalui sistem sinyal jaringan selular dibandingkan dengan jaringan internet, maka dibutuhkan sebuah mekanisme pengamanan yang unik dalam menangani permasalahan tersebut agar besar data yang ditransmisikan diusahakan seminimal mungkin dan tetap menjaga kerahasiaanya.

Kata Kunci : SMS, *Encryption*, *Security Mechanism*, *Cellular Network*, *Mobile Communication*

1. Pendahuluan

Jaringan selular beroperasi dengan membagi-bagi akses layanan berdasarkan jangkauan daerah, yang dibagi kedalam zona-zona tertentu, yang setiap zonanya memiliki sumber daya dan jalur tersendiri. Zona layanan tersebut hanya dapat diakses oleh pengguna yang berada dalam jangkauan zona tersebut. Sehingga jika pengguna berpindah tempat ke daerah lain, walaupun menggunakan layanan yang sama, bisa jadi pengguna tersebut pusat layanan yang berbeda.

Jaringan selular mempunyai keterbatasan dalam hal ketersediaan *bandwith*, karena pada dasarnya jaringan selular beroperasi dengan penggunaan ulang frekuensi yang secara teknis frekuensi-frekuensi tersebut dihasilkan oleh BTS (*base transceiver station*). Hal tersebut mengakibatkan penggunaan frekuensi yang sama untuk beberapa daerah yang berbeda, dengan aturan tidak terjadi interferensi frekuensi antara satu daerah dengan yang lainnya.



Gambar 1 Penggunaan Ulang Tujuh Frekuensi

Untuk menambah kapasitas dari jaringan selular, dapat dilakukan tiga cara :

1. Menambah jumlah jalur komunikasi, dan hal ini tergantung dari kemampuan frekuensi yang digunakan.
2. Menambah efisiensi modulasi dan menerapkan teknik *multiple access*.

3. Penggunaan ulang jalur komunikasi yang sama, dipisahkan oleh jarak yang tidak akan menimbulkan interferensi satu sama lain.

Berdasarkan perkembangan kemampuannya jaringan selular dapat dibagi ke dalam fase 1G yang memiliki bandwith 10Kbits/detik. Selanjutnya dilanjutkan dengan 2G yang secara teoritis dapat memiliki bandwith hingga 270Kbits/detik. Kemudian dilanjutkan dengan teknologi 3G yang memilili bandwith 2Mbits/detik dan dimungkinkan hingga 155Mbits/detik untuk suatu lingkungan yang khusus. Selain itu kini juga sedang dikembangkan teknologi 4G dimana akan diterapkan paradigma *All IP-Environment*.

Sedangkan pada implementasinya jaringan selular adalah sebuah jaringan internal sebuah perusahaan yang menggunakan teknologi *wireless* dengan gelombang berfrekuensi tinggi untuk mengirimkan datanya. Banyak sekali macam-macam teknologi yang dikembangkan untuk komunikasi melalui *mobile device* tersebut, seperti GSM (Global System for *Mobile* Communication), CDMA (Code Division Multiple Access), PHS (Personal Handy Phone System), dan banyak lainnya.

Komunikasi menggunakan jaringan selular dapat mengirimkan data berupa audio maupun teks dan gambar ataupun *content* digital lainnya. Layanan yang paling banyak digunakan dalam komunikasi selular adalah SMS (Short Messaging Service), sebuah layanan yang memungkinkan kita mengirimkan pesan teks ke seseorang secara cepat dan bersifat pribadi.

Seiring dengan perkembangan teknologi, layanan SMS tidak hanya dapat diakses dengan perangkat selular saja seperti *handphone*, tetapi dapat juga diakses melalui internet. Oleh karenanya akan makin banyak sekali celah bagi para pengguna dimana pesan tersebut dapat dibaca oleh orang yang tidak berhak.

Tetapi sistem keamanan seperti yang diterapkan pada email dengan protokol SMTP atau IMAP tidak cocok diterapkan pada layanan SMS, dikarenakan SMS menggunakan jaringan selular untuk mengirimkan datanya. Tidak seperti internet yang memiliki bandwith yang cukup besar dan lebih stabil, pada SMS diusahakan jumlah memori pemrosesan data yang akan diproteksi serta data yang dihasilkan haruslah sekecil mungkin. Jika data yang dikirim terlalu besar, maka kemungkinan kegagalan penyampaian pesan akan semakin besar.

2. Gambaran Umum Jaringan SMS atau Selular

Setiap jaringan selular menggunakan berbagai teknologi yang berbeda seperti GSM dan CDMA, sehingga setiap teknologi menerapkan prosedur operasional yang berbeda pada makalah ini kita akan melihat prosedur pada jaringan GSM yang mirip dengan prosedur operasional pada CDMA, teknologi yang paling banyak digunakan saat ini.

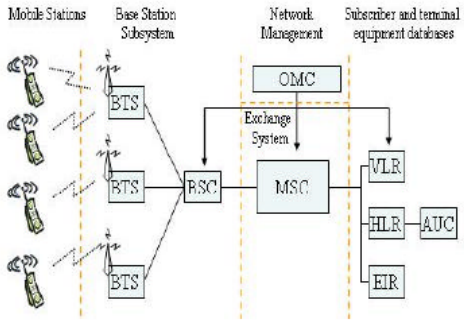
2.1. Mengirimkan Pesan

Ada dua metode untuk mengirimkan pesan teks ke perangkat *mobile*.

1. Menggunakan perangkat *mobile*
2. Menggunakan ESMEs (External Short Messaging Entities).

Mengirim pesan teks dengan menggunakan perangkat *mobile* itu sendiri adalah cara standar yang paling sering digunakan oleh pengguna, Oleh karenanya kita hanya membahas cara kedua.

ESMEs adalah istilah untuk menunjukkan bermacam-macam perangkat dan antar muka aplikasi mulai dari email, portal SMS berbasis web, layana *voice mail*, paging system, dan aplikasi perangkat lunak. ESMEs terhubung dengan jaringan telepon *mobile* melalui internet atau jalur khusus yang lainnya, pesan sebelumnya dikirimkan terlebih dahulu ke server yang mengatur lalu lintas SMS yang disebut juga dengan SMSC (*Short Messaging Service Center*). Sebuah penyedia layanan yang mendukung pengiriman pesan teks harus memiliki setidaknya satu SMSC dalam jaringannya. Tetapi seiring dengan perkembangan popularitas dari layanan SMS via internet, maka penyedia layanan terdorong untuk memiliki lebih dari satu SMSC demi meningkatkan kapasitas layanan.

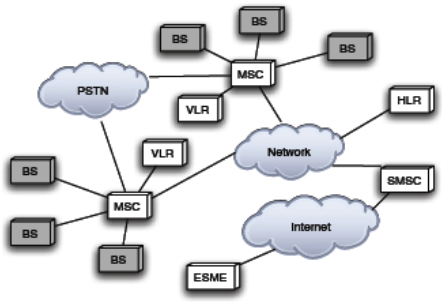


Gambar 2 Arsitektur pada GSM

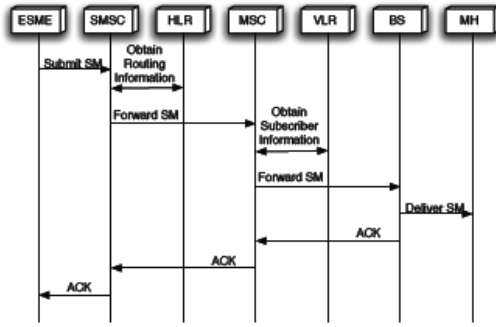
2.2. Perutean Pesan (Routing)

SMSC perlu menentukan jalur pengiriman pesan teks sehingga pesan dapat diterima oleh perangkat *mobile*. SMSC menquery database *Home Location Register* (HLR) yang berisi data pengguna, informasi *subscriber* (info yang berisi *call waiting*, dan pesan teks), data tagihan, *availability* dari pengguna (apakah pesan dapat sampai atau tidak) dan lokasi dari pengguna.

Melalui interaksi-interaksi dengan elemen jaringan-jaringan yang lain, HLR menentukan informasi rute yang dibutuhkan untuk sampai ke tujuan. Jika SMSC menerima balasan bahwa pengguna sedang tidak bisa menerima pesan, maka pesan teks disimpan diserver dan akan dikirim untuk lain waktu. Atau pesan yang dikirm akan mengandung informasi alamat *Mobile Switcing Center* (MSC) penyedia layanan SMS. Berbeda dengan sebelumnya yang memanggil aplikasi agar melakukan *routing* berdasarkan HLC, MSC bertanggung jawab untuk memfasilitasi autentifikasi perangkat *mobile*, manajemen lokasi berdasarkan *base station*. MSC bertindak sebagai gateways ke *Public Switched Telephone Network* (PSTN).



Gambar 3 Jaringan SMS

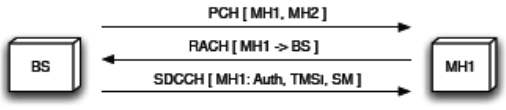


Gambar 4 Flow SMS

Ketika pesan teks sampai dari SMSC, MSC mengambil informasi yang spesifik ke perangkat yang dituju. MSC kemudian melakukan query ke database *Visitor Location Register*, yang akan mengembalikan informasi dari perangkat yang dituju ketika perangkat tersebut berada diluar jangkauan HLR. Setelah itu MSC akan meneruskan pesan kepada *base station* untuk melakukan transmisi pesan melalui media udara.

2.3. Pengiriman Wireless

Pada pengiriman menggunakan medium udara, maka medium udara dibagi kedalam dua bagian, pertama adalah *Control Channels* (CCH), dan kedua adalah *Traffic Channels* (TCH). Kemudian lebih jauh lagi CCH dibagi kedalam dua jalur, *Common CCH* dan *Dedicated CCH*. *Common CCH* mengandung jalur lojik yang terdiri dari *Paging Channel* (PCH) dan *Random Access Channel* (RACH), yang merupakan mekanisme yang digunakan oleh *base station* untuk menginisiasi pengiriman data suara dan SMS. Umumnya, setiap perangkat *mobile* yang terkoneksi akan secara konstan selalu mendengar *Common CCH* untuk sinyal data suara dan SMS.



Gambar 5 Komunikasi SMS lewat medium udara, dimana BS (base station) dan MH (*mobile* host)

Base station mengirim pesan melalui PCH, pesan tersebut mengandung *Temporary Mobile Subscriber ID* (TMSI) yang berkaitan dengan penerima pesan. Jaringan

lebih memilih menggunakan TMSI dibandingkan dengan nomor telepon dari perangkat *mobile*, adalah suatu upaya untuk menghindari dari *eavesdroppring* yang berusaha mengetahu dari identitas si penerima telepon. Ketika perangkat mendengarkan bahwa ada TMSI, maka perangkat tersebut akan mengontak base station melalui RACH dan menginformasikan jaringan bahwa perangkat tersebut dapat menerima pesan teks.

Setelah respon diterima, maka base station menginstruksikan perangkat *mobile* tersebut untuk mendengar jalur *Standalone Dedicated Control Channel* (SDCCH) yang spesifik. Dengan menggunakan SDDCH, *base station* dapat memfasilitasi autentifikasi dari perangkat yang dituju (menggunakan informasi subscriber yang terdapat pada MSC), melakukan enkripsi, mengrimkan TMSI, dan kemudian mengirimkan pesan teksnya.

3. Kelemahan pada Jaringan SMS atau Selular

Sebagian besar dari penggunaan SMS tidak terlalu mementingkan kerahasiaan dari pesannya. Tetapi seiring perkembangan waktu, SMS menjadi pilihan utama untuk mengirimkan informasi-informasi yang harus cepat disampaikan, sehingga isi pesan SMS kini semakin personal dan rahasia. Tetapi ternyata SMS yang kini dapat diakses melalui internet memiliki beberapa celah yang dapat dimanfaatkan oleh pihak-pihak yang tidak bertanggung jawab.

3.1 Menentukan Bottleneck pada Jaringan

Pada teknisnya ada suatu ketimpangan pada kemampuan sistem untuk menangani pengiriman pesan SMS ke jaringan telepon dan menyampaikan pesan ke perangkat *mobile*. Sehingga ketimpangan tersebut dapat menyebabkan terjadinya *bottleneck* pada jaringan. Tetapi untuk memahami bagaimana *bottleneck* itu terjadi, tidak hanya satu faktor saja yang mengakibatkannya, sehingga pembahasannya akan dibagi menjadi tiga bagian pembahasan, pertama tentang konsep pengiriman, kecepatan pengiriman, dan antarmuka yang digunakan.

3.1.1 Konsep Pengiriman

Konsep pengiriman akan menjelaskan bagaimana jaringan mengatur jalannya pesan ketika masuk ke dalam sistem. Dengan mempelajari aliran datanya, kita dapat menentukan bagaimana sistem harus melakukan respon terhadap pesan teks. Secara keseluruhan sistem adalah kumpulan dari banyak titik-titik antrian pesan. Pada dokumentasi ada dua titik utama dalam konsep pengiriman, SMSC dan perangkat yang dituju.

SMSC adalah pusat dari aliran pesan SMS, semua pesan harus melalui SMSC. Tetapi secara teknis SMSC memiliki keterbatasan dalam menangani jumlah pesan. SMSC hanya dapat melakukan antrian pesan dalam jumlah yang terbatas untuk setiap pengguna. Sebelumnya telah dijelaskan bahwa SMSC melakukan mekanisme penyimpanan dan kemudian meneruskan pesan dalam menyampaikan pesan SMS. Pesan akan terus disimpan sampai perangkat *mobile* yang dituju menerima pesan, atau terhapus karena batasan waktu yang ditentukan. Kapasitas penyimpanan dan kebijakan penyedia layanan menentukan apakah pesan diterima atau tidak oleh pengguna layanan SMS.

Setiap penyedia layanan SMS memiliki kapasitas penyimpanan dan kebijakan yang bermacam-macam dalam menangani antrian pesan SMS di SMSC. Dengan mengetahui hal ini, kita dapat mengetahui bagaimana sebuah pesan dapat hilang jika ada seseorang yang melakukan DoS Attack, seperti pada email yang menerima banyak spam, tetapi berbeda dengan email, pengguna SMS hanya memiliki sedikit sekali kapasitas untuk menampung SMS.

3.1.2 Kecepatan Pengiriman

Terjadinya bottleneck bukan hanya terjadi karena keterbatasan media penyimpanan, bottleneck dapat juga terjadi karena kecepatan menyampaikan pesan dari SMSC ke perangkat *mobile* lebih lambat dibandingkan dengan pengiriman pesan dari perangkat *mobile* ke SMSC.

Menentukan seberapa kecepatan pengiriman pesan ke SMSC pada suatu penyedia layanan SMS sangatlah sulit untuk dilakukan, hal ini karena kita tidak tahu ada berapa banyak SMSC yang berada pada

jaringan yang digunakan. Tetapi dilihat dari banyaknya cara untuk mengirimkan pesan ke jaringan selular, seperti lewat website, *email*, *instant messaging*, dan koneksi melalui *Short Messaging Peer Protocol* (SMPP), kita dapat mengestimasi bahwa pengiriman pesan ke dalam jaringan selular dapat berjumlah ratusan pesan per detik.

Pada sebuah percobaan informal didapatkan bahwa kira-kira diperlukan waktu sebesar 0,71 detik untuk pengiriman pesan dan waktu sebesar 7-8 detik untuk menerima pesan. Dari hal tersebut dapat kita lihat ketidakseimbangan waktu untuk mengirim dan menerima pesan.

Sementara SMS memiliki besar data maksimum sebesar 160 byte, tetapi ditambah dengan ukuran data untuk *header* pada protokol yang digunakan. Jika SMS dikirim melalui internet maka dibutuhkan besar data sekitar 1500 byte untuk mentransmisikannya.

Dari hal diatas kita dapat melihat satu celah lagi dalam pengiriman SMS via internet, dengan kemampuan pengiriman banyak SMS dalam satu waktu menggunakan antarmuka website, kemungkinan untuk membuat penuh inbox dan kehilangan SMS yang penting menjadi besar.

3.1.3 Antarmuka

Sebelumnya telah dibahas keterbatasan SMSC dalam menampung antrian pesan SMS yang akan dilanjutkan untuk dikirim ke perangkat *mobile* yang dituju. Tetapi keterbatasan tersebut seolah bertolak belakang dengan kemampuan suatu antarmuka pengiriman SMS via internet seperti website atau instant messaging yang dapat mengirimkan 50 pesan dalam satu waktu, dan ternyata hal tersebut menghasilkan respon kegagalan dalam pengiriman pesan.

Oleh karena hal tersebut dibutuhkan suatu batasan jumlah pengiriman pesan melalui antarmuka menggunakan internet, disesuaikan dengan kemampuan jaringan selular dalam menerima dan mengirimkan pesan.

Service	URL
Instant Messaging	
AOL IM	mymobile.aol.com/portal/index.html
ICQ	www.icq.com/sms/
MSN Messenger	mobile.msn.com
Yahoo Messenger	messenger.yahoo.com/messenger/wireless/
Information Services	
CNN	www.cnn.com/togo/
Google	sms.google.com
MSNBC	net.msnbc.com/tools/alert/sub.aspx
Bulk SMS	
Clickatell	www.clickatell.com
SimpleWire	www.simplewire.com/services/smpp/
START Corp.	www.startcorp.com/StartcorpX/Mobile_Developer.aspx

Gambar 6 Contoh beberapa antarmuka yang dapat digunakan untuk mengirim pesan SMS

3.2 Pembuatan Hit-List

Selain bottleneck, dalam jaringan selular juga dimungkinkan seseorang membuat daftar target yang berupa nomor telepon. Dengan daftar tersebut seseorang dapat menginisiasi sebuah worm yang dapat menyebar dengan cepat.

Langkah yang terpikirkan untuk membuat daftar tersebut mungkin dengan berusaha mendapatkan nomor telepon yang didapat dari sinyal yang tersebar di udara. Tetapi karena penggunaan TMSI maka mendapatkan nomor telepon dari sinyal di udara tidak dimungkinan, hal yang mungkin untuk dilakukan adalah melalui media antarmuka internet.

Ada beberapa metode untuk mendapatkan nomor telepon, salah satunya adalah dengan cara *web scrapping*.

Web scrapping adalah suatu metode yang digunakan oleh spammers untuk mengumpulkan informasi dari target. Melalui *search engines* atau *script* yang dibuat, para spammers mampu mendapatkan alamat email yang diakses melalui halaman web.

Metode yang sama juga dapat diterapkan untuk mendapatkan nomor telepon yang diakses melalui halaman web. Tetapi kelemahan dari metode ini adalah, para spammers tidak pernah tahu apakah nomor yang mereka dapatkan masih aktif ataukah tidak. Biasanya sebuah website pribadi tidak menjamin apakah isi dari halaman webnya *up to date*.

4. Serangan pada Jaringan Selular

Dengan kelemahan jaringan selular seperti bottleneck dan hit-list, sekarang kita akan mendiskusikan kemungkinan-kemungkinan serangan terhadap celah-celah tersebut. Dilihat dari celah-celah tersebut, kemungkinan besar yang dilakukan untuk melakukan serangan adalah mengirimkan sejumlah pesan yang berjumlah sangat banyak untuk membanjiri jaringan dan membuat banyak pengguna tidak mendapatkan layanan SMS seperti yang diharapkan. Tetapi serangan tidak hanya sebatas itu, dengan mengetahui sistem jaringan selular dengan baik seseorang dapat melumpuhkan jaringan dengan antrian yang sangat banyak kemudian berupaya menggunakan identitas orang lain atau identitas penyedia layanan SMS untuk mengirimkan pesan ke target yang dituju.

Gambar 7 Spoofing notifikasi dari penyedia layanan SMS

Kini SMS sangat mirip dengan email, jika digunakan secara baik SMS akan sangat berguna dalam menyediakan layanan komunikasi real time. Tetapi disisi yang lain dengan kemiripan tersebut kita dapat menduga bahwa serangan-serangan yang terjadi pada email dapat juga terjadi pada SMS, seperti spam, phising, dan virus.

5. Solusi untuk Celah pada Jaringan Selular

Ada beberapa mekanisme untuk menjaga jaringan selular yang kini dapat diakses menggunakan internet, dari serangan-serangan, sehingga jaringan selular dapat terus berfungsi secara normal.

5.1. Memisahkan suara dan data

Sebenarnya banyaknya koneksi yang terjadi antara internet dan jaringan selular dapat dibatasai oleh penyedia layanan telekomunikasi. Salah satu cara untuk mengurangi kerusakan jika terjadi serangan adalah memisahkan secara penuh layanan suara dan data. Dengan cara tersebut banyaknya data yang masuk ke jaringan selular melalui internet, tidak akan mengurangi kualitas layanan suara.

Pemisahan tersebut seharusnya diimplementasikan baik pada jaringan kabel maupun pada jaringan wirelessnya. Dengan membuat jalur yang berbeda untuk pengiriman data lewat medium udara, dapat menghilangkan kemungkinan seseorang untuk merusak komunikasi lewat suara. Walalupun dengan hal tersebut terjadi inefisiensi dalam penggunaan sumber daya gelombang. Pemisahan secara parsial telah diimplementasikan dengan memperkenalkan teknologi GPRS dan EDGE.

5.2. Pengaturan Resource

Banyak penyedia layanan komunikasi telah mengetahui statistik penggunaan layanan, sehingga tahu kapan dan dimana tingkat permintaan layanan menjadi sangat tinggi. Contohnya adalah COSMOTE, penyedia layanan komunikasi di yunani. Pada saat penyelenggaraan Olimpiade 2004, mereka menambahkan base station dan MSC di sekitar daerah pelaksanaan Olimpiade. Hasilnya mereka bisa mengirimkan sebanyak 100 juta pesan teks dalam 17 hari. Mirip dengan di Indonesia ketika hari raya Idul Fitri datang.

Begitu juga untuk menanggulangi efek dari serangan SMS mengggunakan internet. Akibat serangan dapat dikurangi dengan menyediakan sumber daya tambahan untuk lokasi-lokasi tertentu yang mempunyai kemungkinan besar memenuhi SMSC dengan pesan teks. Walaupun untuk mengimplementasikan hal ini membutuhkan biaya yang besar, sehingga penambahan peralatan menjadi terlalu mahal.

5.3. Pembatasan Jumlah Pengiriman Pesan

Jika pada keadaan khusus dibutuhkan suatu mekanisme untuk mengembalikan keadaan jaringan selular menjadi normal, adalah dengan cara pembatasan jalur yang digunakan untuk pengiriman SMS.

Pada medium udara yang digunakan untuk pengiriman pesan SMS, jumlah jalur SDDCH yang dapat digunakan untuk menyampaikan pesan dapat dibatasi penggunaannya. Tetapi hal tersebut tidak mencegah seseorang untuk melakukan flooding terhadap jalur-jalur lainnya.

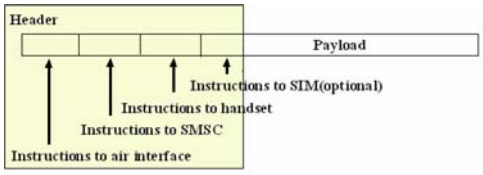
Selain itu kemampuan pengiriman SMS melalui antarmuka web pun harus dibatasi. Kemampuan untuk mengirimkan sepuluh SMS dalam satu pengiriman akan sangat berbahaya, hal itu dapat digunakan untuk melakukan flooding.

6. Solusi Keamanan untuk SMS

Selain masalah menjaga jaringan agar berfungsi dengan normal, salah satu permasalahan lainnya adalah bagaiman agar pesan SMS yang dikirimkan aman dari orang-orang yang tidak berhak. Ada dua hal yang dapat dilakukan, pertama memperbaiki keamanan dari protokol yang digunakan dalam pengiriman SMS dan kedua dengan melakukan enkripsi pada SMS yang biasanya dilakukan oleh *base station* ketika menggunakan TMSI.

6.1. Perbaikan Desain Protokol pada SMS

Salah satu cara untuk membuat peningkatan keamanan pada pengiriman SMS adalah perencanaan penentuan protokol yang tepat pada penggunaan SMS. Dengan protokol yang baik, selain performa yang dijaga juga diberi ruang pada struktur data di SMS untuk dilakukan enkripsi yang akan menjaga autentifikasi dan integritas dari data yang dikirim.



Gambar 8 Struktur Pesan SMS

6.1.1. Kebutuhan Desain

Ada dua pertimbangan dalam mendesain sebuah protokol untuk mengirimkan pesan SMS. Pertama adalah keterbatasan kemampuan CPU dan besar memoy yang dimiliki oleh perangkat *mobile*. Pada protokol yang diterapkan kita menggunakan kriptografi untuk menjaga kerahasiaan dan integritas dari pesan. Kriptografi secara komputasi akan sangat mahal untuk dilakukan ketika kriptografi kunci publik digunakan. Kedua adalah struktur dan panjang dari SMS. Bagian yang dapat dimanipulasi dari pesan SMS hanyalah bagian payload, yang artinya dengan dilakukan enkripsi akan terjadi penambahan byte yang mengurangi kapasitas payload untuk pesan yang belum terenkripsi (plainteks).

Dalam membuat desain protokol ada beberapa kebutuhan yang harus dipenuhi :

- Aman
- Mudah untuk diimplementasikan
- Membutuhkan komputasi yang sedikit
- Tidak diperlukan media penyimpanan untuk kunci kriptografi (karena keterbatasan ruang dalam struktur data pada SMS)
- Melakukan autentifikasi isi pesan
- Melakukan pertukaran kunci

6.1.2. Spesifikasi Desain

Pada sepesifikasi ini kita merencanakan pembuatan desain yang menerapkan mekanisma kunci publik dengan kunci simetri dan strategi autentifikasi menggunakan *password*. Beberapa penjelasan notasi yang akan digunakan

- S : Menyimbolkan Server
- C : Menyimbolkan *Mobile* client
- PK_{pub} : Kunci publik dari server
- PK_{pri} : Kunci privat
- Rc : Angka acak 64 bit yang dibuat oleh C
- Rs : Angka acak 64 bit yang dibuat oleh S
- SK : Kunci rahasia simetri yang saling dipertukarkan antara C dan S
- SQ : 32 bit angka berurut yang dibuat oleh C dimulai dari angka 1
- Slt : 128 bit *salt value* yang dibuat oleh C, yang digunakan C dan S untuk membentuk SK
- DTc : Pesan teks yang dikirim dari C ke S
- DTs : Pesan teks yang dikirim dari S ke C
- $E_{PKpub}[X]$: Enkripsi data X menggunakan kunci publik S
- $E_{SK}[X]$: Enkripsi menggunakan SK
- Username : Nama pengguna dari C yang telah teregistrasi di S
- PIN : Personal Identification Number yang diketahui oleh C dan S
- || : Penggabungan
- *n* : Nomor dari pesan

Protokol yang akan dibentuk terdiri dari dua bagian , yaitu handshake (HS) dan transaction (TS) yang memenuhi persamaan dibawah ini :

$$P = HS + TS \qquad (EQ1)$$

HS adalah bagian dari protokol yang melakukan pertukaran kunci dan autentifikasi sedangkan TS adalah semua transaksi yang dilakukan dengan menggunakan enkripsi. Salah satu yang terjadi pada transaksi tersebut adalah pertukaran pesan SMS antara C dan S.

Selanjutnya kita dapat mendefinisikan HS dengan :

M1 : C → S : E_{PKpub}[Username || Slt || SQ || Rc]
M2 : S → C : E_{SK}[Rc || Rs || SQ], SQ_n > (SQ_{n-1} + 1)
M3 : C → S : E_{SK}[Rc || Rs || SQ], SQ_n > (SQ_{n-1} + 1)

Sedangkan TS didefinisikan dengan :

M4 : S → C : E_{SK}[DTc || SQ_n], SQ_n > (SQ_{n-1} + 1)
M5 : C → S : E_{SK}[DTc || SQ_n], SQ_n > (SQ_{n-1} + 1)
.
.
.
Mn dimana n menunjukan nomor dari pesan

Dari persamaan diatas kita dapat membuat persamaan umum :

HS = M1 + M2 (EQ2)
TS = M_{n-1} dimana n diluar HS (EQ3)

Untuk algoritma enkripsi, akan dibahas dibagian selanjutnya. Kini akan kita deskripsikan apa yang terjadi pada saat *handshake* dan transaksi.

M1 *:* Ketika inisialisasi handshake dilakukan, C akan mengenkapsulasi dan mengenkripsi bagian data-data dari S yang merupakan kunci publik, yaitu username, salt value Slt, nomor urut SQ, dan angka acak Rc. Tujuan dari dibentuknya Rc adalah untuk memastikan bahwa penggunaan protokol yang sedang dieksekusi adalah penggunaan baru.

Kemudian SK diketahui oleh C dengan me-hash Slt, PIN dan username untuk membentuk kunci sepanjang 128 bit. Kemudian M1 yang telah terenkripsi dikirim dalam bentuk binary dari C ke S. Setelah dikirim nilai dari SQ pada C ditambah satu.

M2 : Ketika S menerima M1, S akan mendekripsi M1 dengan kunci privat. Jika proses deskripsi berhasil dilakukan, S akan mengecek apakah username sedang dipakai ataukah tidak. Jika sedang dipakai, S akan mengabaikan M1 dan menghentikan proses handshake. Tetapi jika tidak S akan membentuk SK (caranya sama seperti pada C) dan Rs. Nilai SQ yang diterima dari M1 kemudian ditambah satu. S akan membentuk M2 dengan cara mengenkripsi Rc, Rs, dan SQ yang baru. M2 kemudian dikirim dalam bentuk binary dari S ke C. Setelah dikirim niai dari SQ pada S ditambah satu.

M3 : C akan mendekripsi M2 dengan menggunakan SK. Jika proses dekripsi berhasil dilakukan, C akan memeriksa Rc kemudian membandingkan nilai SQ yang terakhir dikirim pada M1 dengan SQ yang diterima dari M2. Jika nilai SQ pada M2 lebih besar satu dari nilai SQ terakhir yang dikirim pada M1 dan nilai Rc pada M2 cocok dengan nilai Rc yang dibentuk pada saat M1, maka membuktikan kepada C bahwa S adalah asli. Tetapi jika kondisi yang diharapkan tidak tercapai, C akan mengabaikan M2 dan mengakhiri proses handshake. Jika S terbukti asli, C akan mengirimkan M3 yang dibentuk dari hasil enkripsi Rc, Rs, dan nilai SQ yang telah ditambah satu ke S dengan menggunakan SK sebagai kunci enkripsinya, dan mengirimkannya dalam bentuk binary.

Ketika S menerima M3 kemudian mendekripsikannya. Jika proses dekripsi berhasil dilakukan, S akan memeriksa Rs dan nilai SQ. jika nilai SQ yang diterima pada M3 lebih besar satu dengan nilai SQ pada S, dan nilai Rs sama dengan nilai Rs yang dibentuk pada saat pengiriman M2 maka S mengetahui bahwa C adalah asli. Jika kondisi tersebut tidak tercapai, maka S akan mengabaikan M3 dan mengakhiri proses handshake. Sedangkan jika C terbukti asli, maka proses *handshake* telah berhasil dilakukan secara sempurna.

M4, M5, dan Mn : Pada tahap ini pesan akan dikirimkan antara C dan S. pada tahap ini baik C maupun S akan saling mempertukarkan SK. Baik C dan S harus memastikan bahwa SQ pada pesan yang baru diterima nilainya lebih besar satu dibandingkan dengan nilai SQ pada pesan sebelumnya. Hal ini untuk memastikan tidak

terjadi *replay attack* pada saat komunikasi berlangsung.

Timeout akan terjadi jika S atau C tidak menerima pesan yang telah melewati proses HS atau TS. Jangka waktu terjadi timeout ditentukan sesuai kebijakan masing-masing penyedia layanan.

Dibandingkan dengan protokol lainnya seperti HTTPS (*Hypertext Transfer Protocol Secure*), desain protokol diatas lebih kecil dalam ukuran data, dan tidak terlalu mengurangi jumlah ukuran *payload* yang digunakan untuk menyimpan pesan dari SMS.

6.2. Enkripsi SMS

Pada jaringan selular lebar *bandwith* sangat terbatas, sehingga algoritma enkripsi selain harus memenuhi standar keamanan, juga harus menjaga agar hasil file enkripsi tetap kecil (lihat 6.1.1).

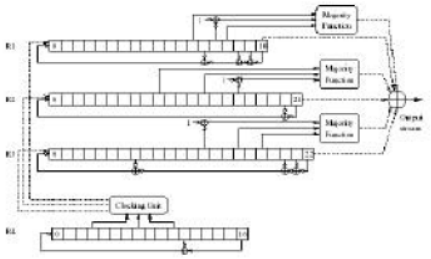
Oleh karena keterbatasan tersebut, solusi yang ditawarkan untuk enkripsi SMS adalah kerahasiaan algoritmanya, yang ditawarkan oleh penyedia layanannya atau juga yang disediakan oleh perangkat *mobile* tertentu.

6.2.1. Algoritma A5

Algoritma A5 adalah algoritma yang digunakan pada GSM. Ada dua versi algoritma A5 yang sering digunakan dalam GSM yaitu A5/1 dan A5/2 yang merupakan algoritma enkripsi stream chipper. Selain algoritma diatas ada juga algoritma A5/3.

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 x^{19} (+) x^5 (+) x^2 (+) x (+)1, x^{22} (+) x (+)1, x^{23} (+) x^{15} (+) x^2 (+) x (+)1, dan x^{17} (+) x^5 (+)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[0].



Gambar 9 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

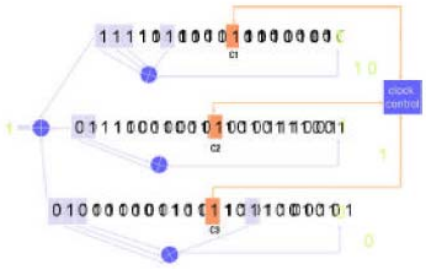
1. kunci seluruh LFSR
2. R1[0]¬R1[0] Å f[i]
3. R2[0]¬R2[0] Å f[i]
4. R3[0]¬R3[0] Å f[i]
5. R4[0]¬R4[0] Å f[i]

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 publik f[i].

Sedangkan proses pembangkitan key-stream adalah:

1. inisialisasi status internal dengan nilai 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 outputnya sebagai key-stream

Pada dasarnya algoritma A5/2 dibangun dengan kerangka yang sama dengan A5/1. Fungsi-fungsi feedback untuk register R1, R2, dan R3 pada A5/2 sama dengan fungsi feedback pada A5/1, begitu pula halnya dengan proses inisialisasi yang dilakukan A5/1 dan A5/2 serupa. Yang membedakan algoritma A5/1 dan A5/2 adalah A5/1 hanya terdiri dari tiga LFSR dengan panjang maksimum masing-masing R1, R2, R3 adalah 19 bit, 22 bit, dan 23 bit sehingga tidak ada pendefinisian untuk register R4 sehingga A5/2 juga harus melakukan inisialisasi R4 dan nilai satu bit pada tiap register harus diisikan dengan nilai 1 setelah dilakukan inisialisasi. Selain itu A5/2 membuang 99 bit output sementara A5/1 membuang 100 bit output.

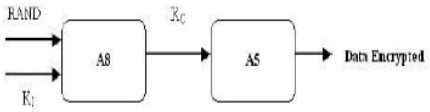


Gambar 10 Algoritma A5/1

Namun, baik A5/1 maupun A5/2 telah berhasil dipecahkan oleh beberapa kriptanalis di dunia dengan menggunakan serangan yang dikenal sebagai plaintext attack.

Mekanisme enkripsi data adalah sebagai berikut :

i. Memproses RAND, yang diterima pada saat akan melakukan otentikasi pengguna, dengan algoritma A8 dan Ki untuk menghasilkan kunci enkripsi Kc (ciphering key).
ii. Mengenkripsi plaintext dengan algoritma A5 dan kunci Kc untuk menghasilkan ciphertext, yang akan ditransmisikan melalui jaringan.



Gambar 11 Skema Enkripsi Pada GSM

Kc dibangkitkan pada saat dilakukan otentikasi pengguna. Untuk setiap anggilan, Kc yang dibangkitkan akan berbeda nilainya.

Kc hasil proses algoritma A8 disimpan ke dalam SIM dan terbaca oleh ponsel. Jaringan juga membangkitkan Kc dan mendistribusikannya kepada base station(BTS) yang menangani koneksi.

6.2.2. Alternatif Algoritma

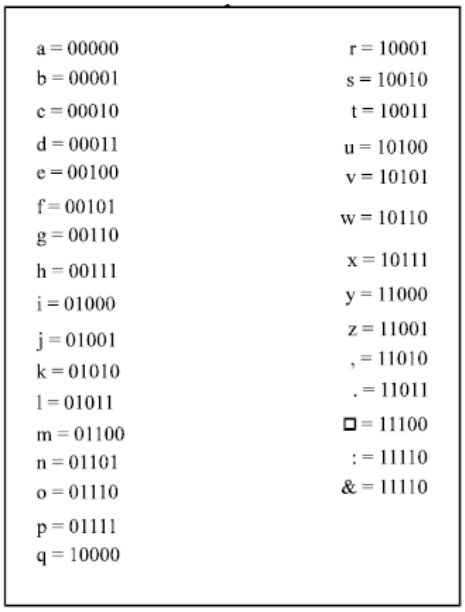
Selanjutnya akan dibahas sebuah algoritma yang berusaha untuk mengurangi kebutuhan memori untuk melakukan enkripsi tersebut. Sehingga server dalam jaringa selular tidak terkurangi bebannya. Walaupun pengurangan komputasinya tidak signifikan untuk sebuah enkripsi, tetapi jika dilihat dengan jumlah pesan SMS yang dikirimkan dalam satu hari, optimasi kecil tersebut menjadi sangat berarti.

Aturan standar ketika kita merepresentasikan satu karakter, maka dibutuhkan satu byte data (8 bit), sekarang bagaimana jika kita bisa merepresentasikan

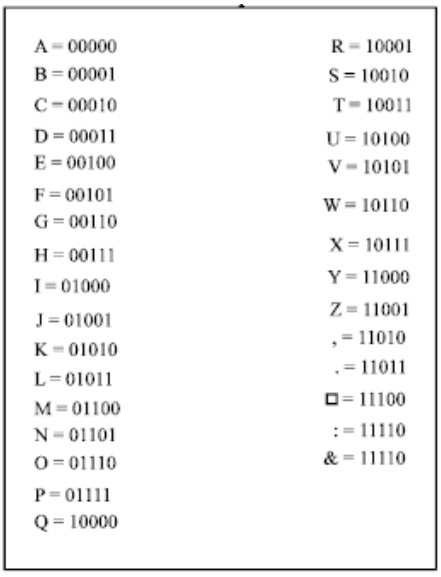
datanya hanya dengan lima bit, hal tersebut akan menghemat alokasi memori di server.

Proses : Secara umum pada SMS kita memiliki karakter-karakter (a, b, c, d, e,….., x, y, z) kemudian karakter (A, B, C, D, E,…., X, Y, Z) dan juga karakter-karakter (?, !, @, #,….,+, -). Biasanya ketika membuat pesan SMS, kita menggunakan karakter lowercase atau uppercase secara tersendiri, dan subscriber tentunya tidak ingin mengubah teks tersebut. Misalnya kita menulis SMS, selamat hari raya idul fitri, atau SELAMAT HARI RAYA IDUL FITRI, walaupun jikalau semuanya berbentuk uppercase akan menimbulkan kesan yang berbeda.

Pada metode ini kita dapat memanfaatkan karakteristik SMS tersebut. Kita dapat mengkategorikan karakter-karakter dalam SMS menjadi dua grup, setiap grup mengandung 32 karakter. Grup 1 mengandung 26 huruf lowercase dan 6 karakter khusus. Kemudian pada grup kedua mengandung 26 huruf uppercase dan 6 karakter khusus.

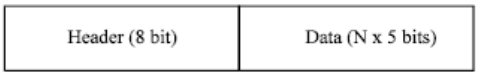


Gambar 12 Grup 1, dilengkapi dengan representasi binary



Gambar 13 Grup 2, dilengkapi dengan representasi binary

Dari penggrupan diatas, sekarang kita mengusahakan untuk membuat frame dari SMSnya.



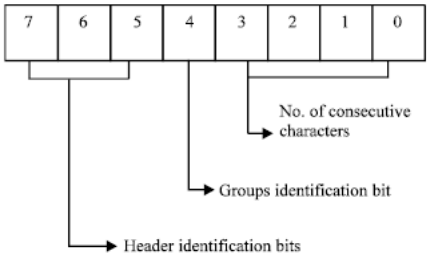
Gambar 14 Usulan Struktur Frame

Karena dalam setiap grup hanya mengandung 32 karakter, membuat representasi karakter-karakter tersebut hanya membutuhkan 5 bit.

Sekarang kita perhatikan 160 byte sebagai stream dari 160 x 8bit. Jika empat atau lebih kit menuliskan karakter berurutan yang berasal dari grup yang sama, kita dapat mengurangi jumlah bit representasi dari (Nx 8) menjadi (Nx5+8) bit. Jumlah huruf yang sama dari tiap grup akan dianggap sebagai dari sebuah frame, yang terdiri dari dua bagian, pertama bagian header dan kedua adalah bagian dari data. Bagian dari header mengandung 8 bit dan data mengandun (Nx5) bit. Dimana N adalah jumlah karakter berurutan yang berasal dari grup yang sama berasal .

Pada header, 3 bit pertama adalah Most Significant Bit (MSB), bit ke-5, ke-6, dan ke-7 digunakan untuk identifikasi header. Bit ke-4 digunakan untuk identifikasi grup, dan 4 bit terakhir digunakan untuk

menyimpan informasi jumlah karakter yang berurutan yang berasal dari grup yang sama.



Gambar 15 Struktur Header

Berikut adalah aturan dalam membaca header.

- Jika 3 bit pertama dari MSB, dari setiap bit yang dibaca adalah 000, maka bit tersebut akan diidentifikasi sebagai header. Tetapi jika tidak akan diperlakukan sebagai karakter biasa yang menggunakan 8 bit. Dari tabel ASCII kita dapat melihat, bahwa setiap karakter pasti memiliki nilai 1 untuk 3 bit pertamanya.
- Bit berikutnya, bit nomor 4 menunjukan dari grup mana segmen tersebut berasal. Jika nilai bitnya 1, maka berasal dari grup 1 dan jika nilainya 0 maka berasal dari grup 2.
- Bit nomor 0 sampai 3, 4 bit tersebut menunjukan jumlah karakter berurutan yang berasal dari grup yang sama. Misalnya jika bit-bit tersebut bernilai 0111, maka 7 kareakter berikutnya dianggap sebagai karakter yang telah terkompresi (dibaca tiap 5 karakter).

Dengan menggunakan 4 bit, maka jumlah karakter maksimum dalam suatu frame adalah 15 karakter. Sehingga jika ada 27 karakter yang berurutan maka dibutuhkan 2 frame yang berari ada 2 header. Header pertama akan mengandung 15 karkakter, sedangkan header kedua akan mengandung 12 karakter.

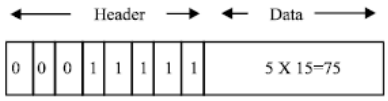
Algoritma : Algoritma dari metode ini dapat dilihat pada gambar 18.

Implementasi : Di bawah adalah contoh implementasi dari metode ini.

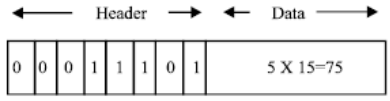
Contoh dari pesan SMS-nya adalah sebagai berikut :

‘congratulation, how are you?’

Diatas terdapat 28 karakter. Pada kata pertama yaitu ‘congratulation’, kata tersebut terdiri dari 15 huruf berurutan yang berasal dari grup yang sama, sehingga kita bentuk satu frame, dimana 8 bit header dan 5 bit untuk tiap karakter sehingg dibutuhkan data sebesar 8+(5x15) = 83 bit. Sedangkan karakter ‘ how are you?’ kita akan membentuk lagi satu frame dengan 8 bit heder, sehingga dibutuhkan 8+(5x13) = 73 bit. Total besar memori yang dibutuhkan adalah (83+73) yaitu 156 bit atau 19,5 bytee ~ 20 byte, yang artinya menghemat ukuran memori sebesar 8 byte.



Gambar 16 Struktur frame untuk ‘congratulation’



Gambar 17 Struktur frame untuk ‘_how_are_you?’

Analisi Performa :

Best case : kasus terbaik terjadi jika terdapat 15 karakter berurutan yang berasal dari grup yang sama. Seperti contoh sebelumnya yaitu kata ‘congratulation’.

Pada umumnya satu karakter membutuhkan 8 bit, sehingga untuk 15 karakter akan dibutuhkan memori sebesar 15x8= 120 bit. Sedangkan dengan metode diatas hanya dibutuhkan 83 bit saja.

Worst case : kasus terburuk terjadi jika tidak ada empat atau lebih karakter berurutan yang berasal dari grup yang sama. Sehingga tidak terjadi kompresi data.

Kompresi pesan tersebut dilakukan pada perangkat *mobile*, tentunya metode ini dapat diimplementasikan jikalau telah distandardisasi, karena berkaitan dengan

penyedia layanan dan produsen perangkat mobile.

Setelah kompresi dilakukan, proses enkripsi dapat dilakukan. Dengan ukuran data yang lebih kecil, ada sisa memori dari ukuran data standar (1 byte untuk tiap karakter) yang dapat dijadikan untuk ruang tempat menyimpan chiperteks.

Memang ide ini akan sulit diimplementasikan, karena membutuhkan standardisasi yang membutuhkan persetujuan dari banyak pihak. Walaupun ide ini cukup menarik, tetapi akan banyak kesulitan untuk menghadapi tantangan di masa depan.

Dengan begitu kita tidak terbatas lagi pada algoritma enkripsi yang menghasilkan chiperteks yang memiliki ukuran yang sama dengan plainteks, tetapi bisa lebih besar dari plainteks. Hal tersebut akan mempersulit seseorang untuk mendapatkan data yang tidak berhak dimilikinya. Selain itu kita tetap dapat membuhi batasan maksimal 160 byte (besar SMS pada umumnya).

Selain itu orang-orang akan beranggapan optimasi sebesar itu (misal menghemat 8 byte) tidak akan berarti banyak jika dibandingkan dengan kemampuan server saat ini.

```
Define NON_Group 2
Define Group1  1
Define Group2  0

InputBuffer      [Stores sms]
OutputBuffer     [Stores Compressed sms]

SameGroupCounter = 0
RunningGroup = NON_Group
I = 0
While not end of data
      X = InputBuffer[I]
     G = FindGroup(X)
     If RunningGroup  = = G then
         SameGroupCounter = SameGroupCounter + 1
         If SameGroupCounter ==15 and RunningGroup<2 then
               Add header to OutputBuffer  [1 byte/8 bit]
               Add Last SameGroupCounter characters from I of InputBuffer to OutputBuffer
in compressed form [5 bit each]
         End if
     Else
         If SameGroupCounter >= 5 and RunningGroup < 2  then
               Add header to OutputBuffer  [1 byte/8 bit]
               Add Last SameGroupCounter characters from I of InputBuffer to OutputBuffer
in compressed form [5 bit each]
         Else
              Add Last SameGroupCounter Characters from I of InputBuffer to OutputBuffer
in ASCII form [8 bit each]
         End if
         SameGroupCounter = 1
         RunningGroup = G
     End if
     I = I + 1
End While
```

Gambar 18 Algoritma enkripsi untuk kompresi

Selain metode-metode enkripsi diatas, saat ini juga diusulkan suatu metode enkripsi yang menggunakan media J2ME.

Perangkat-perangkat mobile saat ini telah banyak diimplementasikan JRE didalamnya, sehingga banyak orang beranggapan, untuk membuat sebuah aplikasi enkripsi berbasis J2ME untuk mengirimkan pesan.

Tentunya si penerima pesan pun harus mengimplementasikan hal yang sama pada perangkat mobilenya. Tentunya dengan hal tersebut tanggung jawab keamanan data sebagian telah diambil oleh perusahaan-perusahaan keamanan yang menyediakan aplikasi enkripsi tersebut.

Ditambah dengan kemampuan penyedia layanan saat ini yang dapat memfasilitasi pesan SMS yang melebihi ukuran 160 byte.

7. Kesimpulan

Sulit untuk mengusulkan solusi yang dapat diimplementasikan untuk saat ini dan akan cukup bertahan lama di masa yang akan datang, karena perkembangan jaringan selular sangatlah cepat.

Kita dapat melihat contoh dari implementasi teknologi GSM pada penyedia-penyedia layanan telekomunikasi selular di Indonesia, walaupun ada teknologi yang lebih baik dari GSM seperti CDMA yang dapat melakukan *multiple access*, tetapi tidak semudah itu untuk diubah karena banyak penyedia layanan telah menginvestasikan dana yang sangat besar untuk teknologi GSM. Sehingga usaha-usaha yang dilakukan adalah dengan mengupgrade semaksimal mungkin teknologi GSM tersebut, walaupun suatu waktu harus diputuskan untuk mengganti infrastruktur demi menerapkan teknologi baru.

Perkembangan teknologi jaringan selular sangatlah sulit untuk diprediksi kearah mana, karena banyak sekali pihak yang terlibat didalamnya. Terlebih lagi masuknya internet didalamnya, sehingga cara pandang jaringan selular sebagai bagian dari internet lama kelaman akan semakin umum.

Dimasa depan ketika teknologi 4G akan diterapkan, ukuran data yang dapat dikirimkan melalui jaringan selular semakin besar dan semua perangkat *mobile* akan memiliki IP, layanan SMS akan berevolusi menjadi layaknya email pada saat kini, tetapi dengan struktur datanya yang kecil, teknologi SMS akan terus menjadi pilihan untuk komunikasi real time. Tetapi dengan begitu permasalahan justru dimulai karena hal tersebut, karena kini penggunaan SMS bukan hanya untuk hal-hal yang tidak penting, oleh karena hal tersebut perlu dibuat mekanisme pengamanan yang lebih aman lagi.

Usulan Solusi :

Saat ini kekuatan algoritma enkripsi terletak dari kerahasiaan algoritmanya padahal hal tersebut mudah dipecahkan dengan melakukan *reverse engineering.* Penekanan pada kekompleksan komputasi bukan usulan yang baik, karena bagaimanapun juga kemampuan komputasi perangkat *mobile* akan jauh lebih rendah dibandingkan dengan komputer sehingga harus dipertimbangkan kembali mengenai performa komunikasi yang harus dibayar dengan peningkatan komputasi yang dilakukan.

Solusi yang ditawarkan penulis adalah menekankan pada perbaikan protokol, seperti pada bagian 6.1 dan mekanisme autentifikasi pada penggunaan SMS seperti perbaikan mekanisme TMSI, kemudian terus mengurangi kebutuhan komputasi untuk melakukan enkripsi hal ini untuk mengurangi beban dari mekanisme yang dilakukan oleh protokol. Penggunaan kunci publik dapat dilakukan dengan bekerja sama dengan pihak yang telah terpercaya secara global.

Selain itu alternatif pihak ketiga, yang disini adalah perusahaan-perusahaan keamanan penyedia aplikasi enkripsi berbasiskan J2ME dapat menjadi solusi praktis yang cukup efektif, walaupun tidak efisien, karena membutuhkan perangkat mobile yang memiliki spesifikasi cukup tinggi.

8. Daftar Pustaka

[1] Abu Yousuf, Mohammad. Hasan, Mustafa. Shams, S.F Shaif. “A New Encryption Method for Short Message Service (SMS)“. Universitas Sains dan Teknologi Mawlana Bhashani Bangladesh, 2006.

[2] Enck, William. Traynor, Patrick. McDaniel, Patrick. Dan La Porta, Thomas. “Exploiting Open Functionality in SMS-Capable Cellular Network”. Universitas Pennsylvania Amerika Serikat, 2005.

[3] Hardiantina, Ratih. Awaliyah, Siti. Syafwin, Sandra. “GSM Security”. Teknik Informatika, Institut Teknologi Bandung.

[4] Rathsinanga, Hulisani. Lo, Johnny. Bishop, Judith. “A Security Mechanism for Secure SMS Communication”. Universitas Pretoria Afrika Selatan, 2004.

[5] Sheriff, Ray E. Fun Hu, Y. “*Mobile* Satelite Communication Networks”. Universitas Bradford Inggris. John Wiley & Sons, 2001.