Pengamanan Pengiriman SMS dengan kombinasi partisi, enkapsulasi, dan enkripsi menggunakan teknik ECB

Arief Suharsono - 13510087
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
arief.suharsono@comlabs.itb.ac.id

***Abstrak*—Pada zaman modern, layanan pesan singkat atau biasa disebut SMS (Short Message Service) yang disediakan oleh service provider GSM/CDMA dipakai sebagai salah satu alternatif untuk berkomunikasi. SMS sangat banyak dipakai, namun belum banyak teknik pengamanan yang diaplikasikan untuk mengamanakan komunikasi melalui SMS. Dalam makalah ini, penulis mencoba mengamankan komunikasi melalui SMS, dengan menggunakan metode kriptografi ECB (Electronic Code Book) yang merupakan salah satu teknik enkripsi block Cipher, dimana ECB sendiri juga memiliki kelemahan. Dalam makalah ini juga, penulis juga mencoba memberikan solusi untuk mengatasi kelemahan ECB, yaitu dengan partisi pesan, dan enkapsulasi pesan untuk menandai masing-masing partisi.**

***Kata Kunci*—ECB, SMS, partisi, enkapsulasi, enkripsi, dekripsi.**

I. PENDAHULUAN

Pada zaman modern, layanan pesan singkat/SMS (Short Message Service) yang disediakan oleh service provider dipakai sebagai salah satu alternatif untuk berkomunikasi. Data SMS dari masing-masing pengguna layanan selular ditransmisikan melalui udara dengan gelombang GSM/CDMA dengan frekuensi tertentu bergantung pada layanan yang digunakan. Meskipun mulai ditinggalkan dan digantikan oleh perpesanan singkat lainnya yang berbasis internet (BBM, Whatssap, YM, dll) karena kurang praktis, namun layanan SMS ini relatif lebih aman daripada penggunaan perpesanan singkat yang berbasis internet tersebut. Hal ini dikarenakan pesan SMS lebih sulit di-"intercept" di tengah jalan, karena jalur data nya bukan melalui jalur data internet, melainkan jalur data internal pada service provider.

Teknik pengamanan transmisi data melalui jalur internet telah banyak dikembangkan, karena serangan-serangan terhadap metode tersebut juga telah banyak dilakukan. Namun pengamanan terhadap SMS masih sedikit dikembangkan. Sehingga, dalam makalah ini, saya akan mencoba membuat rancangan dan implementasi pengamanan pengiriman SMS dengan kombinasi partisi, enkapsulasi, dan enkripsi. Implementasi pengamanannya hanya sebatas prototype, dimana implementasinya akan dilakukan dengan cara membuat aplikasi untuk PC yang terhubung dengan ponsel/modem (sebagai transmitter), bukan membuat aplikasi untuk mobile.

Teknik pengamanannya menggunakan 3 kombinasi, yaitu partisi, enkapsulasi, dan enkripsi. Pada tahap pertama, pesan SMS dikonversi menjadi rangkaian bilangan biner (konversi dari byte ke bit), kemudian dienkripsi dengan metode Block Cipher-ECB (Electronic Code Book), yang selanjutnya menghasilkan rangkaian bit baru. Selanjutnya rangkaian bit baru tersebut direpresentasikan dalam bentuk hexadecimal, representasi inilah yang akan dikirim. Kelemahan dari metode ini adalah sepotong rangkaian plainteks yang sama akan menghasilkan sepotong rangkaian cipherteks yang sama juga, sehingga dilakukan partisi secara acak yang kunci partisinya dibangkitkan dari kunci ECB. Partisi ini dilakukan agar serangkaian cipherteks menjadi saling terpisah karakter-karakternya selama masa transmisi.

Selanjutnya setiap hasil partisi akan dienkapsulasi untuk menandai nomor pesan. Nomor pesan diberikan di awal pesan, untuk menandai partisi tersebut adalah partisi untuk pesan yang mana. Selanjutnya, diberikan nomor partisi, pemberian nomor partisi juga akan dilakukan dengan tersembunyi, yaitu menyisipkan angka nomor partisi pada karakter nomor tertentu sesuai dengan kunci ECB. Pesan yang dienkpsulasi selanjutnya dikirim melalui SMS.

Penterjemahan pesan dilakukan dengan cara yang sama tetapi dengan langkah yang terbalik. Yaitu de-kapsulasi, rekonstruksi hasil partisi, kemudian dekripsi dengan menggunakan kunci ECB.

Dalam pengerjaan makalah ini, penulis mencoba membuat kakas sendiri yang berupa aplikasi desktop. Kakas tersebut digunakan untuk mencoba implementasi dari teknik yang dirancang tersebut. Kakas untuk implementasi dibuat dengan menggunakan bahasa Java, yang dihubungkan dengan sebuah modem GSM melalui Software Gammu v1.28.92.

II. LANDASAN TEORI

A. Block Cipher

Block Cipher adalah salah satu jenis algoritma kriptografi modern. Dimana algoritma kriptografi modern beroperasi dalam mode bit (tidak seperti algoritma kriptografi klasi yang beroperasi dalam mode karakter).

Kunci, plainteks, cipherteks, semua diproses dalam rangkaian bit, dimana dalam pemrosesannya paling banyak menggunakan operasi bit xor.

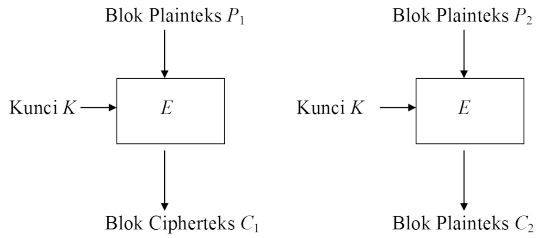
Algoritma kriptografi modern masih menggunakan ide algoritma klasik, yaitu substitusi dan transposisi dengan tingkat kerumitan yang ditingkatkan, sehingga lebih sulit dipecahkan. Hal ini didorong oleh penggunaan komputer digital untuk komunikasi pesan dimana komputer digital dapat merepresentasikan data dalam bit (dan bisa juga dalam byte).

Pada algoritma block cipher, pesan yang akan dienkripsi dipecah menjadi beberapa blok yang terdiri dari beberapa bit yang berurutan, lalu setiap blok dapat direpresentasikan menjadi sesuatu yang berbeda-beda, misal : decimal integer, character, hexadecimal integer. atau tetap dalam bentuk biner.

Operasi yang paling sering digunakan adalah operasi XOR bitwise, yaitu dengan meng-XOR-kan setiap bit yang berkoresponden pada ciphertext/plaintext dengan bit yang berkoresponden pada key. Selain operasi XOR, digunakan juga operasi lain seperti padding, left/right shift, replace, dll.

B. Electronic Code Book (ECB)

Electronic Code Book (ECB) merupakan salah satu algoritma block cipher. Skema ECB secara umum adalah sebagai berikut :



Gambar 1. Langkah-langkah umum algoritma ECB

Langkah pertama, setiap blok plainteks dienkripsi secara individual dan independen menjadi blok cipherteks C_i. Enkripsi C_i adalah $E_k(P_i)$, dan Dekripsi P_i adalah $D_k(C_i)$, dimana dalam hal ini, P dan C adalah rangkaian blok plainteks dan cipherteks. Operasi Enkripsi/dekripsi yang paling sederhana adalah operasi XOR antara plainteks/cipherteks dan kunci, dimana dalam makalah ini penulis akan mencoba menggunakan operasi tersebut.

Salah satu karakteristik mode ECB, blok plainteks yang sama selalu dienkripsi menjadi blok cipherteks yang sama. Karena setiap blok plainteks yang sama akan menghasilkan blok cipherteks yang sama, maka secara teoritis dapat dibuat buku kode yang menterjemahkan antara plainteks dan cipherteks yang berkoresponden. Namun semakin besar ukuran blok, maka ukuran buku kode juga semakin besar (untuk 1 buah kunci). Sehingga, metode ECB ini masih cukup sulit untuk di-kriptanalisis.

Metode ECB ini memiliki beberapa keuntungan. Keuntungan yang pertama, blok plainteks dienkripsi secara independen, sehingga jika ingin mengenkripsi, kita tidak perlu mengenkripsi seluruh plainteks secara linear. Kita dapat mengenkripsi sebagian blok saja, dimanapun tempatnya dalam plainteks. Mode ECB cocok untuk mengenkripsi file yang diakses secara acak, misalnya arsip basis data. Keuntungan berikutnya adalah toleransi terhadap kesalahan, dimana kesalahan 1 atau lebih bit pada blok cipherteks hanya mempengaruhi cipherteks yang bersangkutan pada waktu dekripsi.

Namun metode ECB ini juga memiliki kelemahan, yaitu karena banyak bagian blok plainteks yang berulang, akan menghasilkan banyak bagian blok cipherteks yang berulang juga. Hal ini dapat diserang secara statistik. Cipherteks juga dapat dengan mudah dimodifikasi untuk membodohi/mengelabui penerima pesan. Hal ini disebabkan karena ketika cipherteks dipotong/dibuang sebagian, hasil potongan tersebut masih dapat didekripsi dan menghasilkan kata-kata yang berarti.

III. IMPLEMENTASI

A. Overview

Secara umum, langkah-langkah dalam implementasi algoritma yang diusulkan oleh penulis adalah sebagai berikut :

Proses Enkripsi

1. Representasi karakter menjadi hexadecimal (plainteks dan kunci)
2. Proses Enkripsi dengan menggunakan metode ECB, dengan operasi enkripsi hanya 1x xOR saja.
3. Partisi pesan menjadi 2 bagian, dengan menggunakan kunci yang diberikan user.
4. Enkapsulasi pesan, dengan memberikan :
    a. Penanda bahwa pesan tersebut adalah pesan terenkripsi
    b. Id pesan, untuk menandai pesan-pesan dengan Id yang sama adalah bagian cipherteks untuk plainteks yang sama.
    c. Nomor pesan, untuk menandai pesan tersebut adalah potongan cipherteks nomor 1 atau 2.
5. Pengiriman pesan

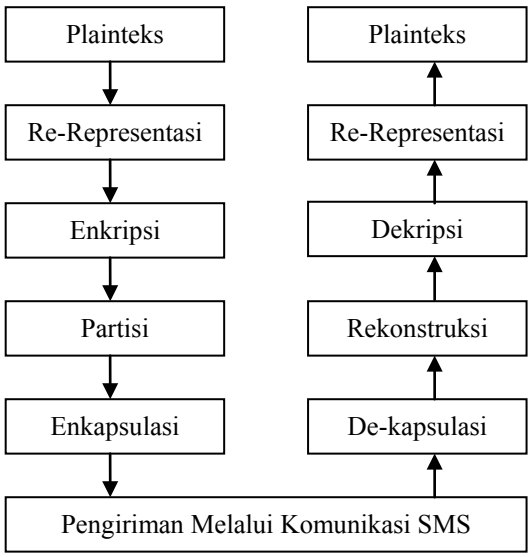
Proses Dekripsi

1. Penerimaan pesan
2. Representasi karakter menjadi hexadecimal (kunci)
3. De-kapsulasi pesan, dengan menghilangkan :
    a. Penanda bahwa pesan adalah pesan terenkripsi
    b. Id pesan, untuk menandai pesan-pesan dengan Id yang sama adalah bagian cipherteks untuk plainteks yang sama.
    c. Nomor pesan, untuk menandai pesan

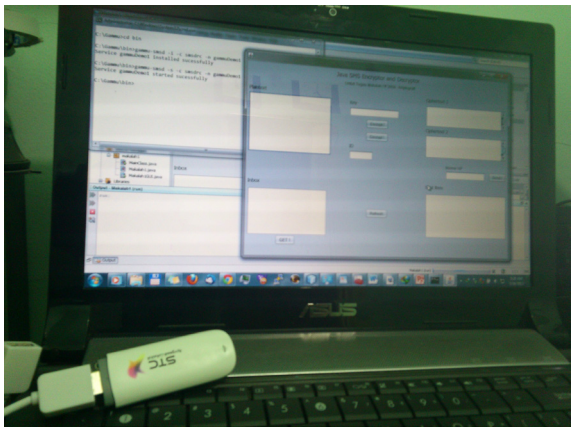
tersebut adalah potongan cipherteks nomor 1 atau 2.

4. Rekonstruksi pesan, dengan menggunakan key dan nomor pesan yang telah didapat sebelumnya, didapatkan cipherteks secara utuh.
5. Proses dekripsi dengan menggunakan metode ECB, dengan operasi hanya 1x xOR saja.
6. Pengubahan representasi dari integer-hexadesimal menjadi karakter.

Komunikasi pesan SMS dilakukan dengan menggunakan kakas buatan sendiri yang telah dihubungkan dengan modem GSM, dimana SMS tersebut dikirimkan melalui jaringan GSM yang tersedia. Secara umum, diagramnya adalah sebagai berikut :



Gambar 2. Langkah-langkah proses Enkripsi dan Dekripsi



Gambar 3. Foto Implementasi

B. Proses Enkripsi

Langkah pertama, rangkaian karakter yang diberikan oleh user, diubah representasinya menjadi bilangan integer-heksadesimal. Dalam pemrograman, 1 karakter (char) direpresentasikan dalam 8 bit, dan 1 digit integer-heksadesimal dalam 4 bit. Seingga, dalam langkah ini hanya perlu membagi representasi karakter dalam bit menjadi 2 bagian, sehingga dihasilkan 2 bilangan heksadesimal.

Contoh :

- Karakter ‘o’, representasi bit : 01101111
  Diubah menjadi 2 buah blok 4 bit : 0110 dan 1111. Sehingga menghasilkan 2 digit heksadesimal : 6f.
- Karakter ‘+’ representasi bit : 00101011
  Diubah menjadi 2 buah blok 4 bit : 0010 dan 1011. Sehingga menghasilkan 2 digit heksadesimal : 2b.

Pengubahan representasi tersebut berlaku untuk setiap karakter yang tersedia dalam SMS (alphabet, angka, dan karakter alphanumeric).

Langkah kedua, plainteks dan kunci yang telah diubah ke dalam representasi heksadesimal, dimasukkan ke dalam operasi enkripsi metode ECB, dimana operasi yang digunakan adalah operasi 1x xOR saja. Bilangan plainteks pertama akan di-xOR dengan bilangan kunci pertama, bilangan plainteks kedua akan di-XOR dengan bilangan kunci kedua, ketika bilangan kunci sudah habis, akan kembali bilangan kunci pertama, dan seterusnya hingga plainteks selesai dienkripsi.

Contoh :

- Plainteks : 6f, representasi bit : 0110 1111
- Kunci : 2b, representasi bit : 0010 1011
- Hasil : (0110 1111) xOR (0010 1011) = 0100 0100. Dalam representasi heksadesimal = 44

Jika panjang kunci lebih kecil dari panjang plainteks, maka kunci akan terus diulang hingga semua bilangan dalam plainteks selesai dienkripsi. Namun jika panjang kunci lebih besar dari panjang plainteks, maka plainteks akan tidak sepenuhnya digunakan (digunakan hanya sepanjang plainteks saja).

Langkah ketiga, hasil enkripsi tersebut dipartisi menjadi 2 bagian dengan menggunakan kunci yang diberikan oleh user. Hal ini dilakukan untuk meningkatkan keamanan pesan, dimana ketika si-interceptor mendapatkan pesan, dia tidak tahu bagaimana urutan pesan dan bagaimana cara mengkonstruksinya menjadi sebuah ciphertext utuh jika dia tidak memiliki kunci. Idenya, setiap bit dari kunci yang diberikan, ditelusuri satu persatu. Jika pada bit pertama dari kunci bernilai 0, maka karakter ke-0 dari cipherteks akan dimasukkan ke dalam potongan pertama, namun jika nilainya 1, maka karakter tersebut akan dimasukkan ke dalam potongan kedua.

Contoh :

- Representasi kunci dalam bit : 0110010101100
- Hasil cipherteks : 2f6a77ee1234da

- Maka hasilnya adalah :
  - Bit pertama kunci = 0, karakter pertama cipherteks (2) dimasukkan ke potongan pertama.
  - Bit kedua kunci = 1, karakter kedua (f) dimasukkan ke potongan kedua.
  - dan seterusnya.
  - Hasil : potongan pertama : 2a7e2da. Potongan kedua : f6e134

Langkah keempat, masing-masing potongan di-enkapsulasi, dengan memberikan identitas-identitas yang dibutuhkan ketika nanti proses dekripsi.

- Menambahkan 2 karakter '#' untuk menandai bahwa pesan tersebut adalah cipherteks
- Menambahkan angka id pesan, sehingga 2 potongan pesan dapat diketahui bahwa potongan-potongan tersebut adalah potongan untuk cipherteks yang sama. id pesan diakhiri dengan 1 digit angka 0.
- Menyisipkan nomor pesan (1 atau 2), pada setiap potongan pesan, pada karakter ke-x, dimana x adalah panjang kunci (dalam representasi heksadesimal) jika panjang kedua potongan pesan lebih panjang dari kunci, atau x adalah panjang pesan yang paling pendek jika salah satu atau kedua pesan lebih pendek dari kunci.

Langkah kelima, kedua potongan pesan dikirim melalui jaringan GSM, dalam pembuatan makalah ini, penulis mengaplikasikannya dengan membuat kakas pengenkripsi-pendekripsi yang terhubung dengan modem GSM sebagai pengirim pesan.

B. Proses Dekripsi

Pada dasarnya, proses dekripsi hanya mengikuti proses enkripsi dengan langkah terbalik. Dengan kunci terlebih dahulu diubah menjadi representasi heksadesimal.

Langkah pertama, program meminta seluruh pesan yang diterima (inbox) dari modem GSM yang telah terhubung, dengan komputer, lalu program akan meminta user untuk memilih pesan mana yang ingin di-dekripsi.

Langkah kedua, kedua potongan pesan yang didapat dari SMS dikumpulkan dan di-dekapsulasi. Proses dekapsulasi meliputi :

- Menghilangkan 2 karakter '#' di-awal sebagai penanda pesan terenkripsi
- Menghilangkan id pesan
- Melihat nomor urutan pesan, dengan melihat panjang kunci dan panjang kedua pesan, jika :
  - Panjang kunci < Panjang kedua pesan, lihat nomor urutan pesan pada karakter ke-x pada setiap potongan pesan, dengan x adalah panjang kunci dalam representasi heksadesimal.
  - Panjang kunci > Panjang kedua pesan, lihat nomor urutan pesan pada karakter ke-x pada setiap potongan pesan, dengan x adalah panjang terkecil dari kedua pesan.
- Nomor pesan tersebut kemudian diubah ke-dalam representasi internal program untuk menandai urutan pesan, lalu dihilangkan dari potongan pesan tersebut.

Langkah ketiga, kedua potongan pesan di-rekonstruksi dengan menggunakan kunci yang diberikan oleh pengguna. Idenya, setiap bit dari kunci yang diberikan, ditelusuri satu persatu. Jika pada bit pertama dari kunci bernilai 0, maka karakter ke-0 dari cipherteks akan diambil dari potongan pertama, namun jika nilainya 1, maka karakter tersebut akan diambil dari potongan kedua

Langkah keempat, cipherteks yang sudah tersusun penuh, di-dekripsi dengan metode ECB dengan kunci yang diberikan pengguna, dengan operasi hanya 1x xOR saja. Dari langkah ketiga ini akan dihasilkan plainteks dalam bentuk representasi heksadesimal.

Langkah kelima, plainteks yang terrepresentasi dalam bentuk heksadesimal tersebut dikonversi menjadi representasi karakter yang dibaca oleh user, menjadi plainteks seperti semula.

D. Contoh Kasus

- Plainteks : "Mari Bertemu di depan perpustakaan setelah kuliah pukul 12.00"
- Kunci enkripsi : "//;*&KodeEnkripsi"
- Id Pesan = 999
- Pemrosesan

1. Re-representasi :
   Plainteks :
   4d617269204265274656d75206469206465706
   16e207065727075737461 6b61616e2070757361
   7420736574656c6168206b756c6961682070756
   b756c2031322e3030
   Kunci :
   2f2f3b2a264b6f6465456e6b7269707369
2. Enkripsi :
   Hasil Cipherteks :
   624e494306090a161120031e520d19530d4a5f5a
   44063b0a1615301d1f13021112070f5f4e59473f
   4f1700310b07130150181c43465a42063b1a0f10
   294e5a40474043
3. Partisi :
   Kunci (dalam bit) :
   0010 1111 0010 1111 0011 1011 0010 1010
   0010 0110 0100 1011 0110 1111 0110 0100
   0110 0101 0100 0101 0110 1110 0110 1011
   0111 0010 0110 1001 0111 0000 0111 0011
   0110 1001
   Hasil potongan 1 :
   62e06911352d930da5a406b1513011170545943
   47107351c34a2030f1025a403
   Hasil potongan 2 :

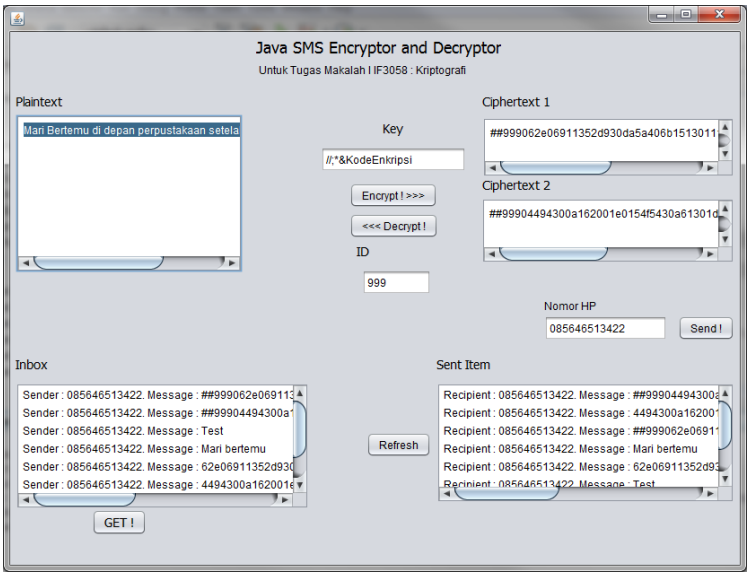
4494300a162001e0154f5430a61301df1220ffe7f
f1003b010101846546b1a94e40744

4. Enkapsulasi
   Hasil potongan 1 :
   ##999062e06911352d930da5a406b1513011170
   5**1**4594347107351c34a2030f1025a403
   Hasil potongan 2 :
   ##99904494300a162001e0154f5430a61301df12
   2**2**0ffe7ff1003b010101846546b1a94e40744

IV. PENGUJIAN

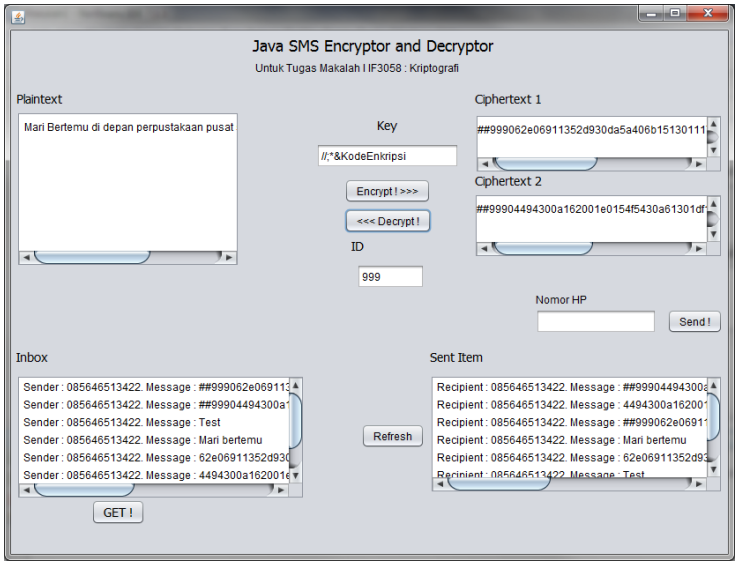
Pengujian dilakukan dengan menggunakan kakas java buatan sendiri, dengan menggunakan Modem Huawei E173, dengan menggunakan jaringan IM3 GSM-WCDMA.

Pengujian dilakukan dengan menggunakan contoh kasus pada bagian implementasi. Hasilnya adalah sebagai berikut :



Gambar 4. Pengujian program untuk proses Enkripsi

Pengujian kedua, dilakukan dengan mencoba proses dekripsi, dan dapat dilihat bahwa kedua bagian cipherteks menghasilkan plainteks yang sama seperti semula.



Gambar 5. Pengujian program untuk proses Dekripsi

V. ANALISIS HASIL

Pada bagian ini, penulis mencoba melakukan analisis hasil, untuk mengukur tingkat keamanan algoritma, dan untuk menguji apakah modifikasi yang dilakukan penulis dapat mengatasi metode ECB.

A. Percobaan Plainteks Berulang

Plainteks : "halotemanhalo"
Kunci : "S"
Hasil Cipherteks 1 :
"##99903313f233e33323"
Hasil Ciphterteks 2 :
"####9990b223c7632db3fc"

Dari percobaan tersebut, dilakukan pengulangan kata "halo", namun pada hasil cipherteks 1 dan cipherteks 2 tidak ditemukan bagian yang berulang. Sehingga, dapat disimpulkan teknik ini dapat mengatasi kelemahan ECB pada plainteks berulang.

B. Percobaan Pemotongan/Pembuangan Pesan

Plainteks : "Silahkan kirim uang lima puluh juta rupiah"
Kunci : "+-/<>"
Hasil Cipherteks 1 :
"##9990784356041c15465e4ce4742**1eb44965**a0b5f74a5"
Hasil Ciphterteks 2 :
"##9990445d44c154225f511e4**15b1445**d558350b47485f5a4c54"
Cipherteks 1 dipotong :
##9990784356041c15465e4ce4742**1eb965**a0b5f74a5
Cipherteks 2 dipotong :
##9990445d44c154225f511e4**15b45d**558350b47485f5a4c54
Hasil Dekripsi : Gagal

Dari percobaan tersebut, dilakukan pemotongan pada Cipherteks1 dan Cipherteks2, masing-masing 2 karakter, dan hasilnya, pesan tidak dapat didekripsi. Sehingga, dapat disimpulkan bahwa teknik ini dapat mengatasi kelemahan ECB pada bagian pemotongan pesan.

C. Pengubahan 1 bit plainteks

Plainteks : "Silahkan kirim uang lima puluh juta rupiah"
Kunci : "+-/<>"
Hasil Cipherteks 1 :
"##9990784356041c15465e4ce47421eb44965a0b5f74a5"
Hasil Ciphterteks 2 :
"##9990445d44c154225f511e415b1445d558350b47485f5a4c54"
Plainteks : "Silahkan kirim uang **lma** puluh juta rupiah"
Kunci : "+-/<>"
Hasil Cipherteks 1 :
"##9990784356041c15465e4ce470e4ee554e5b5958f43"
Hasil Ciphterteks 2 :
"##9990445d44c154225f511e415b1441c541a141585d1e5f555"

Dari percobaan tersebut, dilakukan pengurangan 1 karakter plainteks, hasilnya memberikan perubahan pada beberapa (lebih dari 1) karakter pada Cipherteks. Sehingga dapat disimpulkan bahwa dengan teknik ini, pengubahan 1 karakter plainteks dapat memberikan perubahan pada beberapa karakter cipherteks

D. Penambahan 1 karakter cipherteks

Plainteks : "Silahkan kirim uang lima puluh juta rupiah"
Kunci : "+-/<>"
Cipherteks 1 :
"##9990784356041**c1**5465e4ce47421eb44965a0b5f74a5"
Ciphterteks 2 :
"##9990445d44c154225f511e415b1445d558350b47485f5a4c54"
Hasil Plainteks : "Silahkan kirim uang lima puluh juta rupiah"

Kunci : "+-/<>"
Cipherteks 1 :
"##9990784356041**ac1**5465e4ce47421eb44965a0b5f74a5"
Ciphterteks 2 :
"##9990445d44c154225f511e415b1445d558350b47485f5a4c54"
Hasil Plainteks : Gagal

Dari percobaan tersebut, dilakukan penambahan 1 karakter cipherteks, menyebabkan cipherteks tidak dapat didekripsi (gagal). Dari percobaan ini dapat disimpulkan bahwa penambahan 1 karakter cipherteks menyebabkan cipherteks tidak dapat didekripsi (gagal).

VI. KESIMPULAN

Dalam implementasi, SMS dapat dikirim dan diterima dengan baik, seluruh proses dekripsi dan enkripsi dapat berjalan dengan baik, sehingga dapat disimpulkan bahwa metode ECB dapat diaplikasikan dalam pengamanan pesan SMS.

Dalam analisis hasil, didapat kesimpulan bahwa algoritma yang dibuat dapat menangani kelemahan algoritma ECB, baik dalam hal plainteks berulang dan pemotongan cipherteks.

Dari analisis hasil, pengubahan sedikit plainteks dapat mengubah banyak bagian cipherteks, dan pengubahan sedikit cipherteks dapat mengubah banyak bagian plainteks atau bahkan dapat membuat cipherteks tidak

dapat didekripsi.

Dari keseluruhan implementasi dan analisis, dapat dilakukan pengembangan makalah ini lebih lanjut. Misalnya menggunakan algoritma yang lain yang lebih kuat dari ECB. Pengembangan juga dapat dilakukan dengan membuat aplikasi mobile untuk enkripsi SMS ini mengingat penggunaan SMS pada dasarnya dilakukan pada perangkat mobile, sedangkan sekarang yang dibuat hanya versi desktop.

REFERENCES

[1] Munir, Rinaldi. 2011. "Bahan Kuliah IF3054 Kriptografi". Departemen Teknik Informatika, Institut Teknologi Bandung

[2] http://wammu.eu/gammu : Tutorial penggunaan perangkat modem/ponsel sebagai sms gateway untuk PC
waktu akses : 25 Maret 2013, 18:00 WIB

[3] http://www.unicode.org/charts/PDF/U0000.pdf : Daftar karakter ASCII
waktu akses : 25 Maret 2013, 18:00 WIB

[4] http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf : Teknik ECB dan block cipher lainnya
waktu akses : 25 Maret 2013, 18:00 WIB

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, 26 Maret 2013

Arief Suharsono - 13510087