Aplikasi Pesan dengan Algoritma *Twofish* pada *Platform Android*

Messaging Application with Twofish Algorithm on Android Platform

Midian Octaviano Gurning

Prodi S1 Sistem Komputer, Fakultas Teknik, Universitas Telkom

thunderhearted91@gmail.com

Abstrak

Perkembangan teknologi mobile sangat cepat dalam jangka waktu beberapa tahun terakhir ini. Dan Android menjadi salah satu platform yang paling populer digunakan. Diiringi dengan semakin banyaknya pengguna, sistem keamanannya juga harus semakin ditingkatkan. Dibutuhkan sistem enkripsi dan dekripsi yang cukup aman dan juga cepat agar tidak membebani perangkat pengguna. Salah satu alternatifnya adalah enkripsi dengan menggunakan algoritma Twofish. Pada tugas akhir ini, algoritma Twofish digunakan untuk enkripsi SMS (Short Message Service) dengan bahasa pemrograman Java. Aplikasi ini diharapkan bisa beroperasi tanpa menghambat kinerja dari perangkat dengan menggunakan algoritma Twofish.

Kata kunci : Kriptografi, enkripsi, dekripsi, Twofish, Android, SMS

Abstract

Technology surely improves so fast for these last few years. And Android becomes one of the most popular platform that people often use. As the more people use this platform, the security system must be improved as well. It needs a encryption and decryption that secure enough and also lightweight so it won't slow down the device. One of the alternative is using Twofish Algorithm. In this final project, Twofish algorithm is used as an encryption to SMS (Short Message Service) in Java programming language. This application is expected to run without slowing down the performance with Twofish Algorithm.

Keywords : Cryptography, encryption, decryption, Twofish, Android, SMS

1. Pendahuluan

Perkembangan teknologi membawa hal positif bagi perkembangangan telekomunikasi. Tapi seiring berkembangnya teknologi, timbul juga beberapa masalah, seperti masalah keamanan. Dibutuhkan suatu sistem yang bisa menjamin keamanan informasi yang dimiliki pengguna. Salah satu jenis komunikasi elektronik yang paling populer digunakan adalah komunikasi dengan menggunakan pesan teks. Salah satu cara untuk mengamankan komunikasi ini adalah dengan menggunakan enkripsi. Diperlukan algoritma enkripsi yang efektif tetapi jua simpel digunakan dan tidak membebani perangkat pengguna. Di dalam tugas akhir ini, algoritma Twofish dipilih sebagai algoritma enkripsi.

2. Dasar Teori

Algoritma Twofish mempunyai 2 elemen utama yang berfungsi sebaai pengacak pesan, yaitu matriks MDS (Maximum Distance Separable) dan PHT (Pseudo Hadamard Transform) [5]. Algoritma ini dikembangkan oleh Bruce Schneier. Kemampuan yang paling menonjol dari algoritma ini dilihat dari digunakannya S-box yang bersifat key dependent dan sudah dipre-komputasi, serta *key-scheduling* yang relatif kompleks.

Twofish didesain untuk memenuhi kriteria AES yang diberikan oleh NIST, seperti[5] :

1) Blok cipher simetri sebanyak 128-bit
2) Panjang kunci yaitu antara 128, 192 atau 256
3) Tidak memiliki kunci lemah
4) Efisien dari segi kecepatan, baik pada *processor* Intel Pentium Pro maupun perangkat keras dan lunak lainnya
5) Desain yang sederhana, meliputi :
   a) Bisa melakukan penambahan kunci
   b) Mudah untuk diimplementasikan pada banyak platform
   c) Bisa melakukan *stream-ciphering*
   d) Bisa dikombinasikan dengan fungsi *Hash* dan MAC (*Message Authentication Code*)

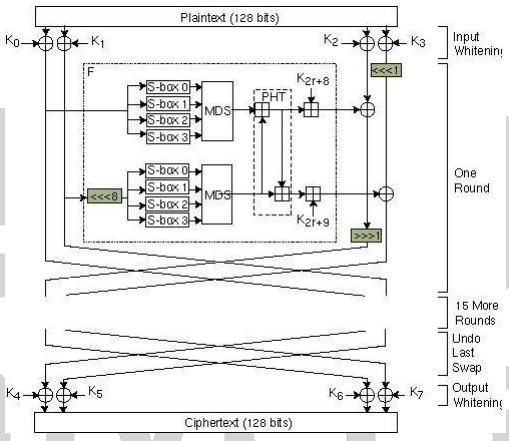
Twofish bekerja sangat baik pada CPU kecepatan tinggi, CPU yang mempunyai smart card kecil dan juga pada perangkat keras. Hampir semua algoritma enkripsi melakukan hal yang sama, yaitu mengambil kunci dan kemudian mengacaknya dengan round yang banyak, lalu membuat kunci untuk setiap round. *Twofish* juga melakukan hal tersebut. *Twofish* mengambil kunci, membuat S-Box yang *key-dependent*, kemudian membuat subkey untuk setiap round-nya.

Twofish juga mempunyai hal unik lainnya, yaitu melakukan *trade-off* antara waktu *set-up* kunci dan waktu enkripsi[8]. Ada 2 pilihan yang bisa dilakukan dalam trade-offnya:

1) *Set-up* kunci lama dan enkripsi lebih cepat. Hal ini sangat cocok dilakukan apabila sedang mengenkripsi pesan dalam jumlah yang banyak dalam kunci yang sama.
2) *Set-up* kunci cepat dan enkripsi lama. Hal ini cocok dilakukan apabila sedang mengenkripsi blok-blok kecil yang disusun secara seri dan kuncinya sering berubah-ubah.

2. 1. **Blok-blok pada Algoritma Twofish**

Gambar 2. 1. **Alur Round**



2. 2. **S-Box**

Merupakan tabel substitusi non-linear yang dipakai pada kebanyakan blok cipher. Ukuran masukan dan keluarannya bervariasi dan dapat dibuat random atau dirancang secara algoritmik. Setiap S box didefinisikan dengan 2, 3 atau 4 byte key material. Tergantung berapa ukuran kunci yang dipakai Twofish.

Penyebab mengapa S-box diberi key adalah:

1) S-box yang bersifat tetap (fixed) membuat hacker bisa mempelajari S-box nya dan menemukan titik lemah
2) Dengan menggunakan S-box yang bersifat key dependent, hacker tidak bisa mengetahui apa S-box nya
3) Pertahanan terhadap "serangan tidak terduga"
4) Kompleksitas dari S-box tergantung dari panjangnya key
5) Kekurangannya membutuhkan waktu yang lebih lama untuk setup key, karena tiap key harus mempunyai S-Box

2. 3. **MDS Matrix**

MDS (Maximum Distance Separable) merupakan pemetaan matriks yang dilakukan secara linear yang dilakukan pada 2 komponen, yaitu a dan b dengan cara melakukan penambahan a+b. Pemetaan dari MDS dapat direpresentasikan sebagai sebuah matriks yang terdiri dari komponen a x b[7]. MDS berfungsi sebagai pengacak pada algoritma Twofish. Serge Vaudenay menyarankan memakai MDS Matrix dalam kriptografi primitif untuk menghasilkan fungsi linear yang dia namakan sebagai multipermutasi. Menurut Serge, fungsi ini merupakan difusi sempurna dimana mengubah input sebanyak t akan mengubah output setidaknya sebanyak m-t+1. Serge menunjukkan bagaimana memanfaatkan fungsi tidak sempurna untuk fungsi kriptoanalisis yang bukan multipermutasi.

Pseudo-Hadamard Transform (PHT)

Pseudo-Hadamard Transform merupakan fungsi pengacak sederhana, transformasi dua arah pada 1 bit string yang menghasilkan difusi kriptografis. Twofish memakai PHT 32-bit untuk mengolah hasil dari fungsi g yang didefinisikan sebagai berikut [7]:

$$A0 = a + b \bmod 232$$

$$B0 = a + 2b \bmod 232$$

Dan untuk melakukan reverse terhadap fungsi tersebut dilakukan dengan cara:

$$b = B0-A0 \pmod{232}$$

$$a = 2A0-B0 \pmod{232}$$

2. 4. **Whitening**

Whitening merupakan teknik meng-XOR-kan key material sebelum ronde pertama dan sesudah ronde terakhir. Dalam serangan kriptoanalisis terhadap Twofish, terbukti bahwa whitening secara substansial meningkatkan kesulitan pada pesan yang sudah di cipher, dengan cara menggunakan fungsi f untuk menyembunyikan masukan spesifik pada awal dan akhir round dari fungsi F[7].

2. 5. **Fungsi F (F Function)**

Fungsi F adalah permutasi terhadap nilai 64-bit. Fungsi F bersifat key-dependent. Fungsi f memerlukan 3 buah variabel, yaitu R0 dan R1 (dalam satuan words) dan nomor round (r) yang dipakai nantinya untuk memilih subkey. R0 kemudian dimasukkan ke dalam fungsi g, dan akan didapatlah nilai T0. R1dirotasikan ke kiri sebanyak 8-bit dan dimasukkan ke dalam fungsi g untuk mendapatkan nilai dari T1. Nilai dari T0 dan T1 kemudian dimasukkan ke dalam PHT dan ditambahkan 2 words expanded key.

T0 = g(R0)

T1 = g(ROL(R1,8))

F0 = (T0 + T1 + K2r+8) mod 232

F1 = (T0 + T1 + K2r+9) mod 232

Berdasarkan fungsi di atas, F0 dan F1 merupakan hasil dari fungsi F. Sedangkan ROL adalah nilai R1 yang dirotasikan ke kiri sebanyak 8 bit.

2. 6. **Fungsi g**

Gambar 2. 2. **Fungsi g**

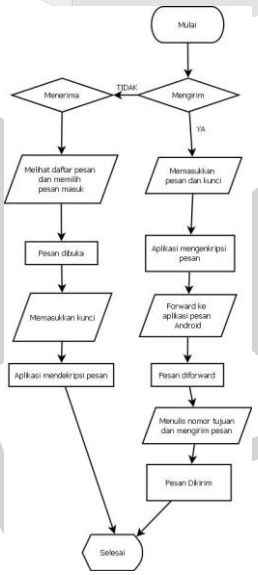
$$
\begin{aligned}
x_i &= \lfloor X/2^{8i} \rfloor \bmod 2^8 \qquad i = 0, \ldots, 3 \\
y_i &= s_i[x_i] \qquad i = 0, \ldots, 3 \\
\begin{pmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \end{pmatrix} &= \begin{pmatrix} \cdot & \cdots & \cdot \\ \vdots & \text{MDS} & \vdots \\ \cdot & \cdots & \cdot \end{pmatrix} \cdot \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} \\
Z &= \sum_{i=0}^{3} z_i \cdot 2^{8i}
\end{aligned}
$$

Berdasarkan gambar diatas, nilai masukan dari X dipisah menjadi 4 byte. Tiap byte kemudian dimasukkan ke dalam masing-masing S-Box, dimana tiap S-Box bersifat key-dependent. Tiap S-Box juga bersifat bijektif, yang artinya apabila masukannya 8-bit, maka keluarannya juga 8-bit. Hasil dari keempat S-box kemudian didefinisikan sebagai sebuah vektor matriks berukuran 4x4 dan kemudian dikalikan dengan matriks MDS. Hasil perkalian ini disusun dalam 32-bit word dan merupakan keluaran dari fungsi g.

3. **Pembahasan**

3. 1. **Diagram Alur (Flowchart)**

Gambar 3. 1. **Flow Chart**



Flowchart merupakan jenis diagram yang merepresentasikan algoritma dan juga alur atau proses yang terdapat di dalam sistem. Pesan awal (plain text) merupakan kondisi pesan pada saat sebelum dienkripsi. Kemudian pengguna meminta aplikasi untuk mengubah pesan awal menjadi chiper text. chiper text merupakan kondisi pesan setelah dienkripsi. Ketika pengguna yang dituju menerima pesan yang berbentuk chiper text, penerima pesan harus memasukkan kunci yang diperlukan untuk

melakukan mendekripsi pesan agar pesan bisa diolah kembali menjadi plain text. Kunci ini disebut juga dengan private key. Private key sudah disepakati terlebih dahulu antar pengguna yang terlibat dalam proses pendistribusian pesan.

3. 2. **Diagram Use-Case**

Gambar 3. 2. **Diagram Use-case**

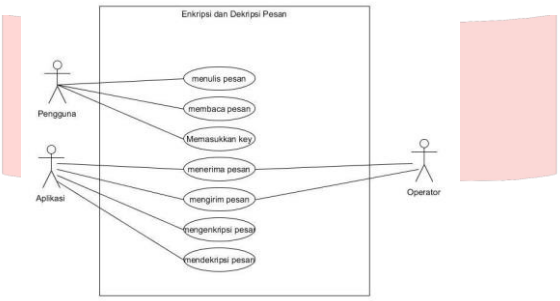


Diagram use-case merupakan gambaran interaksi dan peran pengguna di dalam sistem serta fungsi-fungsi yang tersedia pada sistem untuk pengguna.

Pada aplikasi ini, ketika pengguna berada dalam posisi "aktif", hal-hal yang bisa dilakukan oleh pengguna antara lain:

1) Menulis pesan
2) Menulis kontak
3) Mengenkripsi pesan
4) Mengirimkan pesan

Sedangkan ketika pengguna dalam posisi "pasif", hal yang dilakukan pengguna antara lain:

1) Menerima pesan
2) Melihat percakapan
3) Mendekripsi pesan

3. 3. **Parameter Pengujian**

Pengujian merupakan proses eksekusi aplikasi untuk menemukan ada tidaknya kesalahan atau *bug* pada aplikasi yang dibangun. Pengujian merupakan tahap yang penting dalam pengembangan aplikasi untuk mendapatkan hasil akhir yang sesuai dengan apa yang diharapkan oleh pengguna. Pengujian tidak mengarah pada perbaikan kualitas walaupun error atau kesalahan telah ditemukan dan diperbaiki. Ada beberapa parameter yang dijadikan acuan saat melakukan pengujian aplikasi ini. Parameter yan diuji mengacu pada spesifikasi, desain dan performansi. Parameter-parameter tersebut antara lain sebagai berikut:

1) Melakukan analisis terhadap *memory heap*
2) Mengetahui apakah aplikasi dapat berjalan dengan baik pada perangkat *Android*.
3) Mengetahui waktu respon aplikasi
4) Mengetahui tingkat *Avalanche Effect*

3. 4. **Implementasi Antarmuka**

Antarmuka aplikasi diimplementasikan pada perangkat Asus Zenfone dengan sistem operasi *Android* versi 4.4.2. Implementasi antarmuka diujicobakan untuk tiap fase memulai aplikasi, menulis pesan, dan membaca SMS.

Berikut ini adalah tampilan antarmuka dari aplikasi ini:

1) Memulai aplikasi

Gambar 4.1. **Memulai Aplikasi**

2) Menulis pesan

Tampilan dari implementasi saat pengguna memilih untuk menulis pesan. Pengguna menuliskan pesan yang akan dikirim, menuliskan *key* untuk enkripsi, dan aplikasi kemudian meneruskan pesan yang sudah dienkripsi ke aplikasi SMS *default* dari *Android*.

Gambar 4.2. **MenulisPesan**

3) Membaca SMS

Tampilan dari aplikasi saat membaca SMS yang masuk dan mendekripsi *chiper text* menjadi *plain text*.

Gambar 4.3. **Membaca SMS**

4. Kesimpulan dan Saran

4. 1. Kesimpulan

Berdasarkan pengujian yang dilakukan, aplikasi enkripsi ini mampu bekerja dengan baik pada perangkat *Android.* Algoritma yang dipakai juga bekerja dengan sangat cepat. Sedangkan untuk pemakaian memori, tidak mengganggu kinerja perangkat yang dipakai.dan berjalan normal. *Avalanche effect* dari algoritma *Twofish* juga sangat baik. Dari 1 bit yang diganti, keluaran *chiper text* berubah secara drastis.

4. 2. Saran

Saran yang dapat penulis berikan untuk pengembangan selanjutnya adalah sebagai berikut :

1) Dilakukan kriptoanalisis untuk menguji ketahanan algoritma enkripsi
2) Pengembangan pada platform *smartphone* selain *Android*
3) Melakukan perbandingan dengan algoritma enkripsi yang lain

5. Daftar Pustaka

[1] Cryptography, Define Cryptography at Dictionary.com , dilihat 29 Mei 2016 http://dictionary.reference.com/browse/cryptography

[2] http://www.archaeology.org/9903/newsbriefs/egpt.html , dilihat 29 Mei 2016

[3] http://www.simonsingh.com/Zimmermann_Telegram.html , dilihat tanggal 29 Mei 2016

[4] http://www.garykessler.net/library/crypto.html , dilihat tanggal 29 Mei 2016

[5] S. Aarti 1997, 'Study of MDX Matrix Used in *Twofish* AES (Advanced Encryption Standard) Algorithm and its VHDL Implementation'

[6] S. Bruce & W. Doug. 2000. ' A Performance Comparison of Five AES Finalists'.pdf.

[7] S. Bruce dkk. 2000. '*Twofish*-A 128 bit Block Cipher'.pdf

[8] The *Twofish* Encryption Algorithm, http://www.drdobbs.com/article/print?articleId=184410744&siteSectionName=security , dilihat tanggal 4 Juni 2016

[9] Tumanggor. S. F. 2009, 'Studi Enkripsi dan Dekripsi File dengan Menggunakan algoritma *Twofish*.pdf'

[10] https://en.wikipedia.org/wiki/History_of_cryptography, dilihat tanggal 16 Juli 2016.

[11] https://en.wikipedia.org/wiki/Data_Encryption_Standard, dilihat tanggal 16 Juli 2016.

[12] https://en.wikipedia.org/wiki/Symmetric-key_algorithm, dilihat tanggal 16 Juli 2016.

[13] https://en.wikipedia.org/wiki/Public-key_cryptography, dilihat tanggal 16 Juli 2016.