ANALISIS PERBANDINGAN ANTARA ALGORITMA KRIPTOGRAFI SERPENT DAN AES PADA IMPLEMENTASI ENKRIPSI SMS DI PERANGKAT ANDROID

ANALISYS OF COMPARATION BETWEEN CRYPTOGRAPHIC ALGORITHM SERPENT AND AES IN SMS ENCRYPTION ON ANDROID DEVICE IMPLEMENTATION

Bayu Rizki R, R. Rumani M, Muhammad Nasrun

Prodi S1 Teknik Komputer, Fakultas Teknik, Universitas Telkom

Prodi S1 Teknik Komputer, Fakultas Teknik, Universitas Telkom

Prodi S1 Teknik Komputer, Fakultas Teknik, Universitas Telkom

kunciitem@students.telkomuniversity.ac.id, rumani@telkomuniversity.ac.id, nasrun@telkomuniversity.ac.id

Abstrak

Perangkat mobile saat ini telah berkembang dengan sangat pesat. Salah satu perkembangannya adalah sistem operasi Android. Teknologi enkripsi merupakan proses yang memiliki komputasi cukup kompleks, sehingga proses enkripsi dan dekripsi dapat membebani perangkat mobile. Agar suatu aplikasi enkripsi dapat berjalan dengan baik di perangkat Android maka dibutuhkan penggunaan algoritma kriptografi yang paling optimal. Maka dari itu dilakukan implementasi enkripsi SMS pada perangkat Android. Implementasi tersebut dilakukan dengan menggunakan Android SDK dan Android Studio untuk membuat aplikasinya. Algoritma kriptografi yang digunakan adalah Serpent dan AES. Hasil implementasi kedua algoritma tersebut akan diuji dengan parameter waktu enkripsi, waktu dekripsi, heap memory, dan avalanche effect. Dari hasil pengujian tersebut didapat sebuah kesimpulan kalau AES adalah algoritma kriptografi paling optimal untuk diterapkan di perangkat Android.

Kata Kunci : Kriptografi, Serpent, AES, Android.

Abstract

Development of mobile devices today is really fast. One of the notable development is Android operating system. Encryption technology are a process that have complex computation, therefore can slowed down mobile devices. For an encryption application can work well in Android device, criptographic algorithm that can run most optimum on Android devices need. Therefore an implementation made for SMS encryption on Android devices. That implementation done with Android SDK and Android Studio for make the application. Cryptographic algorithm used in this implementation are Serpent and AES. That implementation will tested with parameter like encryption time, decryption time, heap memory, and avalanche effect. Result of the test concluded AES are most optimum cryptographic encryption to implement on Android device.

Keyword : Cryptography,Serpent, AES, Android.

1. Pendahuluan

Perangkat mobile hingga sekarang ini telah berkembang dengan sangat pesat. Dimulai dari telepon selular yang hanya dapat mengirimkan pesat teks dan suara, hingga sekarang menjadi lebih kompleks dan mampu menjalankan file multimedia. Salah satu perkembangan dari perangkat mobile adalah sistem operasi Android. Android yang merupakan sistem operasi open source yang dikembangkan dari kernel linux menjadi sangat populer dan digunakan di berbagai perangkat mobile saat ini.

Teknologi enkripsi merupakan proses yang memiliki komputasi cukup kompleks, sehingga proses enkripsi dan dekripsi dapat membebani perangkat Android. Selain itu perangkat Android memiliki spesifikasi yang berbeda-beda sehingga suatu aplikasi enkripsi harus memiliki kompatibilitas yang tinggi. Agar suatu aplikasi enkripsi dapat berjalan dengan baik di perangkat Android maka dibutuhkan penggunaan algoritma kriptografi yang paling optimal. Untuk menentukan algoritma mana yang paling optimal berjalan di perangkat Android dibutuhkan pengujian – pengujian terhadap performansi algoritma tersebut di perangkat Android.

Dari algoritma kriptografi yang ada, dipilih algoritma kriptografi AES dan Serpent. AES (Advanced

Encryption Standart) merupakan standar algoritma enkripsi yang telah disahkan oleh NIST (National Institute of Standart and Technology) pada tahun 2001. AES dipilih karena, dalam jurnal yang diajukan (Daemen, Rijmen 2002) menyebutkan kalau AES mudah untuk diimplementasikan di berbagai perangkat. Sedangkan Serpent dipilih karena (Anderson, Bilham dan Knudsen 2001) menyebutkan di jurnalnya kalau Serpent merupakan salah satu finalis AES yang memiliki tingkat keamanan yang tinggi. Selain itu menurut jurnal yang diajukan (Benfano 2008) menyatakan kalau Serpent memiliki delay saat enkripsi dan dekripsi yang rendah sehingga layak untuk diimplementasikan di perangkat Android. Kedua algoritma tersebut akan diuji implementasinya di perangkat Android untuk menentukan mana algoritma yang paling optimal untuk diterapkan.

2. Dasar Teori

2.1 Android SDK (Software Development Kit)

Android SDK adalah tools API (Aplication Programming Interface) yang digunakan untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, middleware dan aplikasi kunci yang di release oleh Google. Saat ini disediakan Android SDK (Software Development Kit) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java. Sebagai platform aplikasi netral, Android memberi anda kesempatan untuk membuat aplikasi yang kita butuhkan yang bukan merupakan aplikasi bawaan hanphone/smartphone.

2.2 Android Studio

Android Studio adalah official IDE untuk mengembangkan aplikasi Android berdasarkan IntelliJ IDEA. Dengan kapabilitas dari IntelliJ, Android Studio menawarkan:

- Sistem gradle-based yang fleksibel
- Membangun varian dan apk file generation ganda
- Template kode untuk membantu membangun aplikasi tertentu
- Layout yang kaya dengan bantuan untuk drag and drop editing
- Built-in support untuk Google Cloud Platform, memudahkan untuk integrasi dengan Google Cloud Messaging dan App Engine

2.3 Pengertian Kriptografi

Kriptografi adalah suatu metode keamanan untuk melindungi suatu informasi dengan menggunakan kata-kata sandi yang hanya bisa dimengerti oleh orang yang berhak mengakses informasi tersebut. Kriptografi merupakan satu-satunya metode yang digunakan untuk melindungi informasi yang melalui jaringan komunikasi yang menggunakan landline (kabel di bawah tanah), satelit komunikasi, dan fasilitas microwave (gelombang mikro) menururt (Ariyus 2008, p. 23). Prosedur-prosedur kriptografi juga bisa digunakan untuk authentifikasi pesan, digital signature, dan identifikasi pribadi untuk mengotorisasi transfer uang secara digital melalui ATM, kartu kredit, dan melalui suatu jaringan.

Kriptografi sebenarnya adalah suatu metode yang sering sekali digunakan untuk melindungi berbagai macam data yang prosesnya disebut dengan enkripsi, yaitu adalah suatu proses yang mengkonversi sebuah pesan plaintext menjadi sebuah ciphertext yang bisa dibalik ke bentuk asli seperti semula, yang juga bisa disebut sebagai proses decoding atau dekripsi.

2.3.1 Kriptografi Kunci Simetri

Terdapat dua jenis algoritma kriptografi berbasis kunci yaitu kunci simetris dan kunci publik. Menurut (Scheiner 1996, p. 53) algoritma kunci simetris yang terkadang disebut algoritma konvensional, adalah algoritma dimana kunci untuk enkripsi dapat dikalkulasi dari kunci untuk dekripsi begitu juga sebaliknya. Kebanyakan algoritma kunci simetris menggunakan kunci yang sama dengan kunci untuk dekripsi. Algoritma ini juga disebut algoritma kunci rahasia atau algoritma satu kunci dimana sebelum berkomunikasi kedua pihak harus menyetujui kunci yang digunakan terlebih dahulu.

2.3.2 Block Cipher

Algoritma kriptografi beroperasi pada plainteks atau cipherteks dalam bentuk blok bit, biasanya berukuran 64 bit atau lebih. Dengan block cipher, plaintext yang sama akan selalu mengenkripsi blok dengan hasil ciphertext yang sama dengan kunci yang sama.

2.4 Algoritma Kriptografi

2.4.1 Serpent

Serpent merupakan salah satu finalis dalam kompetisi menentukan AES yang menjadi standar algoritma enkripsi. Serpent bekerja dengan masukan berupa block dengan ukuran 128 bit yang terdiri dari empat buah 32 bit word yang direpresentasikan dalam little endian. Pada jurnal (Anderson, Bilham, dan Knudsen 2001) menyatakan, Serpent mengenkripsi plaintext menjadi chipertext melalui proses sebanyak 32 putaran. Dalam proses ini digunakan sebuah kunci sepanjang 256 bit.

2.4.2 Advanced Encryption Standard (AES)

Algoritma AES merupakan algoritma block chiper yang dibuat berdasarkan rijndael chiper dan diciptakan oleh dua kriptografer belgia yang bernama Joan Daemen dan Vincent Rijmen. Block chiper berarti algoritma mengkonversi plaintext yang sudah dibagi dalam suatu kelompok yang disebut block dengan ukuran tertentu. AES menggunakan kunci simetris yang artinya kunci untuk enkripsi dan dekripsi sama. Algoritma ini diresmikan oleh NIST sebagai standar algoritma enkripsi pada tahun 2001. Pada jurnal (Daemen dan Rijmen 2002) menyebutkan kalau AES menggunakan ukuran block sebesar 128 bit dan dapat menggunakan kunci sepanjang 128, 192, dan 256 bit. Panjang kunci mempengaruhi jumlah ronde pada proses enkripsi. Di tugas akhir ini akan digunakan kunci sepanjang 256 bit yang berarti akan ada 14 putaran proses enkripsi.

2.5 Enkripsi Di Perangkat Mobile

Perangkat mobile yang bertukar informasi sensitif telah digunakan secara luas. Menjaga keamanan tanpa memberatkan perangkat mobile menjadi isu yang penting. Dari hasil yang pernah diujikan, setiap algoritma memiliki keunggulan tertentu untuk perangkat tertentu dan konfigurasi tertentu pula. Contohnya, jurnal (Benfano 2008) menyebutkan kalau RC6 memiliki keunggulan yang baik antara konsumsi daya, kecepatan dan kebutuhan storage, akan tetapi algoritma Serpent dapat dieksekusi lebih cepat.

2.6 Pengiriman SMS

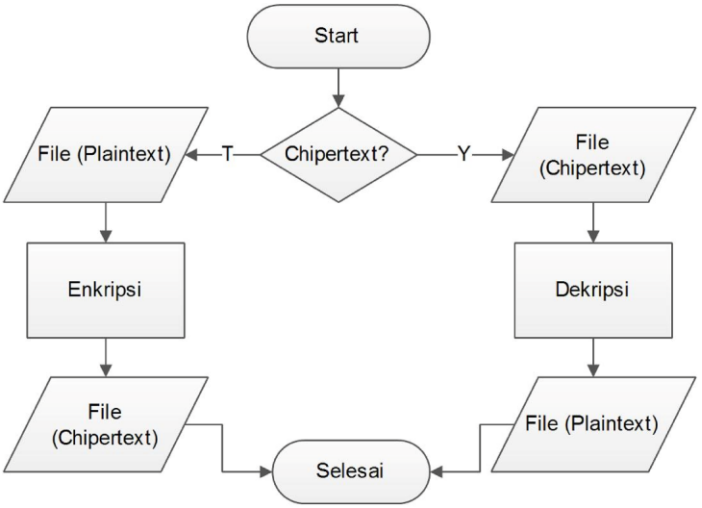
Menurut (Enck et al. 2005) ada dua metode untuk mengirim sebuah pesan text ke suatu perangkat bergerak, yaitu melalui suatu perangkat bergerak ataupun melalui suatu External Messaging Entities (ESMEs). ESMEs terdiri dari sejumlah besar perangkat berbeda dan memiliki berbagai antarmuka seperti email, portal messaging berbasis web yang terkoneksi pada jaringan telepon bergerak melalui internet maupun kanal dedicated tertentu.

Pesan awalnya dikirim ke suatu server yang menangani trafik SMS yang dikenal sebagai Short Messaging Service Center (SMSC). Suatu provider yang mendukung pesan text harus memiliki paling sedikit satu SMSC pada jaringan mereka. SMSC perlu untuk menentukan bagaimana pesan disampaikan ke perangkat target. SMSC menanyakan kepada suatu basis data Home Location Register (HLR) yang menyimpan data pemakai dan informasi lokasi. Melalui interaksi dengan elemen lainnya, HRL menentukan routing informasi ke tujuan. Jika SMSC menerima balasan bahwa target tidak dapat dicapai, maka pesan akan disimpan untuk dikirim nantinya, jika sebaliknya maka akan dibalas dengan alamat Mobile Switching Center (MSC) yang tersedia untuk melayani. Ketika suatu pesan tiba dari SMSC ke MSC, MSC menanyakan kepada suatu basis data Visitor Location Register (VLR) yang akan mengembalikan suatu duplikat informasi dari perangkat target ketika dia tidak berada pada HLR-nya. MSC kemudian mengirim pesan kepada Base Station (BS) untuk disampaikan ke target.

3. Pembahasan

3.1 Perancangan Sistem

Berikut merupakan Flowchart aplikasi yang dirancang. Flowchart ini menggambarkan alur kerja dari aplikasi yang dirancang. Aplikasi mempunyai kemampuan untuk mengenkripsi dan mendekripsi pesan. Plaintext merupakan file sebelum dienkripsi. Pesan tersebut dapat diakses secara normal. Sedangkan chipertext merupakan pesan yang telah dienkripsi. Pesan ini tidak dapat diakses sebelum didekripsi.



Gambar 3.1 Flowchart Aplikasi

3.2 Cara pengujian

Setelah aplikasi selesai dibuat maka akan dilakukan pengujian - pengujian terhadap aplikasi tersebut untuk melihat keunggulan dari tiap algoritma kriptografi yang diimplementasikan. Ada beberapa parameter uji yang digunakan pada tugas akhir ini.

Waktu Enkripsi dan Dekripsi

Waktu enkripsi merupakan waktu yang dibutuhkan untuk mengubah plaintext ke chipertext, sebaliknya untuk waktu dekripsi. Pada tiap algoritma akan diuji enkripsi terhadap pesan yang bervariasi ukurannya.

Heap Memory

Heap memory merupakan porsi dari memory yang dialokasikan secara dinamis. Untuk menyediakan pengalaman user yang stabil, penting untuk aplikasi tidak begitu banyak mengambil porsi memory.

Avalanche Effect

Avalanche effect merupakan suatu output yang diinginkan dari algoritma enkripsi. Output tersebut berupa chipertext. Hasil dari enkripsi dikatakan baik apabila perubahan 1 bit input menghasilkan perubahan besar pada output. Avalanche efferct merupakan parameter uji yang biasa digunakan untuk menggambarkan tingkat keamanan pada kriptografi kunci simetris dan fungsi hash. Pengujian ini dilakukan untuk melihat tingkat keamanan dari algoritma Serpent dan AES. Avalanche effect dihitung dengan rumus

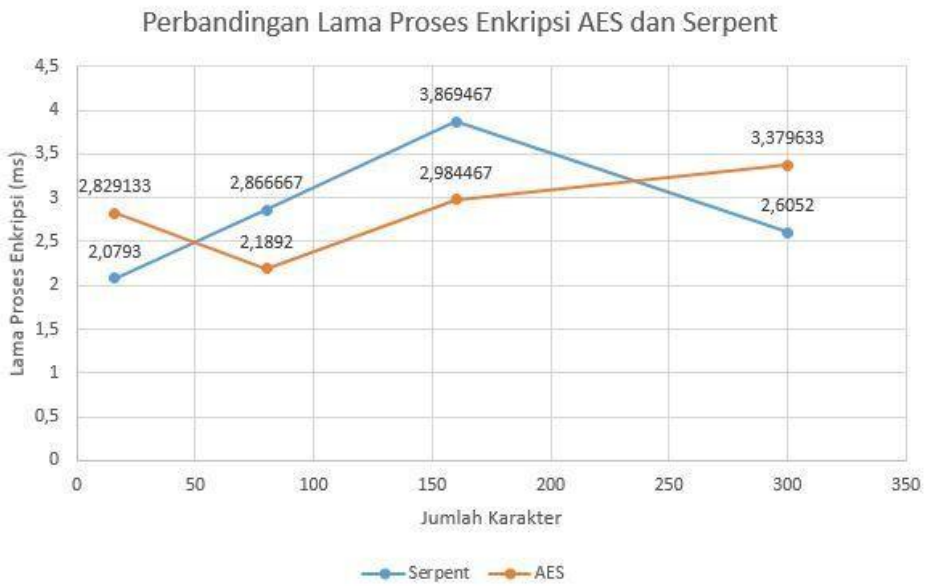
$$h \qquad = \frac{h \qquad h\,(\,h \qquad)}{h \quad (\,h \qquad)}$$

Rumus 3.1 Menghitung Avalanche Effect

3.3 Hasil Pengujian

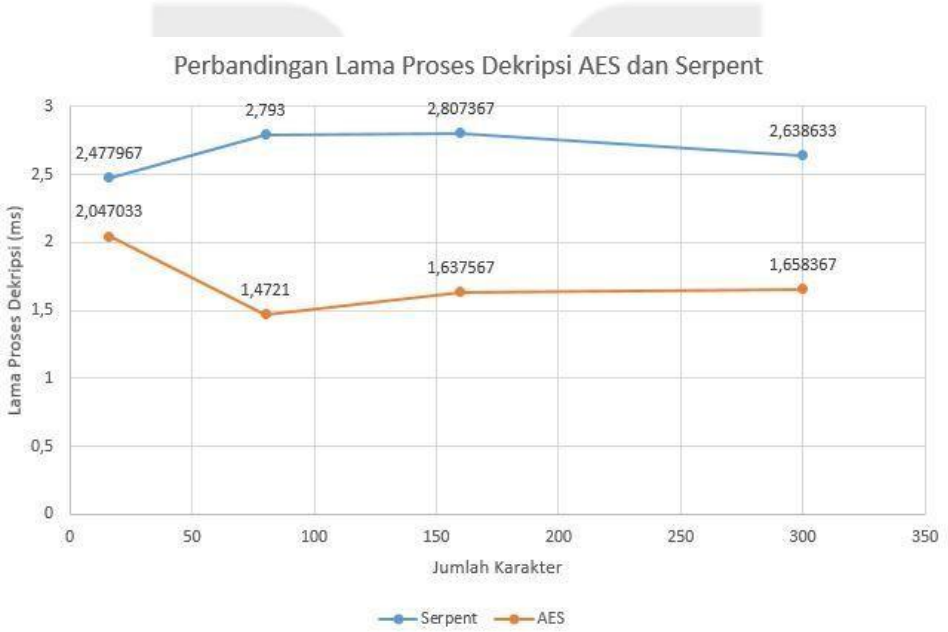
Pengujian Waktu Enkripsi dan Dekripsi

Pengujian ini dilakukan dengan mengukur lama proses enkripsi dan dekripsi pesan pada masing-masing algoritma dengan menggunakan systrace. Pengujian dilakukan dengan panjang pesan berbeda – beda yaitu 16,80,160,300 karakter. Pengujian dilakukan 30 kali untuk masing – masing panjang karakter dengan hasil uji berupa millisecond.



Gambar 3.2 Perbandingan Waktu Enkripsi AES dan Serpent

Gambar 3.2 menunjukkan dengan panjang 16 dan 300 karakter, Serpent lebih unggul dari AES dalam waktu enkripsi.



Gambar 3.3 Perbandingan Waktu Dekripsi AES dan Serpent

Gambar 3.3 menunjukkan AES lebih memakan waktu sedikit untuk enkripsi.

Pengujian memory

Pengujian memory dilakukan dengan Android Device Monitor (ADV). Untuk skenario pengamatan dengan ADV, aplikasi dijalankan kemudian mengakses menu enkripsi dan dekripsi pada aplikasi dan mengukur total heap size dan allocated heap size yang terlihat.

Rata – rata total dan allocation heap dari algoritma Serpent adalah 15,928 MB dan 9,5568 MB. Sedangkan rata – rata total dan allocation heap dari algoritma AES adalah 15,6995 MB dan 9,4197 MB. Dari sini dapat dilihat AES lebih sedikit dalam penggunaan memory dibanding Serpent.

Avalanche Effect

Berikut merupakan hasil pengujian avalanche effect:

1. Serpent

Tabel 3.1 Avalanche effect Serpent

Plaintext	Chipertext	Avalanche effect
73656E696E206B616D6973207261627 5	2B6A77645369535462364D7A79317147	0,4375(56)
53656E696E206B616D6973207261627 5	58437032424A6653357834514270314 9	

2. AES

Tabel 3.2 Avalanche effect AES

Plaintext	Chipertext	Avalanche effect
73656E696E206B616D6973207261627 5	424F5450697766664872304649776665	0,3515625(45)
53656E696E206B616D6973207261627 5	76486C665861687664747035536563 0	

Dari hasil di atas terlihat bahwa Serpent memiliki avalanche effect yang lebih tinggi dari AES.

4. Kesimpulan dan Saran

4.1 Kesimpulan

Dari implementasi dan pengujian yang telah dilakukan dapat dilihat jika kedua algoritma mampu bekerja dengan baik pada perangkan Android. Kedua aplikasi memiliki waktu enkripsi dan dekripsi yang begitu cepat. Sedangkan untuk penggunaan memory tidak begitu ada perbedaan yang signifikan. Dengan hasil pengujian sebagai berikut :

1. Enkripsi dengan panjang pesan 16 dan 300 karakter, Serpent lebih cepat dari AES
2. Enkripsi dengan panjang pesan 80 dan 160 karakter, AES lebih cepat dari Serpent
3. Dekripsi dengan panjang pesan 16, 80, 160, 300 karakter, AES lebih cepat dari Serpent
4. Avalanche effect untuk Serpent dan AES adalah: 0,4375 dan 0,3515625

Meskipun avalanche effectnya lebih baik, namun dari segi waktu dekripsi AES lebih unggul. Dari pengujian tersebut dapat diambil kesimpulan kalau AES lebih efisien untuk diimplementasikan ke perangkat Android dibandingkan Serpent.

4.2 Saran

Saran yang dapat penulis berikan untuk melakukan pengembangan selanjutnya antara lain sebagai berikut:

1. Pengimplementasian dan pengujian pada perangkat dengan platform selain Android, contohnya Windows Phone dan IOS.
2. Dilakukan pengujian cryptoanalisis untuk menguji ketahanan algoritma enkripsi.

Daftar Pustaka

[1] Anderson, R, Biham, & E, Knudsen, L 2001, 'Serpent: A Proposal for the Advanced Encryption Standard', Encyclopedia of Cryptography and Security, pp. 563-564.

[2] Android Developer 2012, 'Android Studio Overview', dilihat 3 Mei 2015, https://developer.Android.com/tools/Studio/index.html

[3] Ariyus, D 2008, 'Pengantar Ilmu Kriptografi', Andi, Yogyakarta.

[4] Daemen, J & Rijmen, V 2002, 'The Design of Rijndael: AES – The Advanced Encryption Standard', FIPS-197: Federal Information Processing Standard Publication 197, dilihat 3 Mei 2015, http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[5] Safaat, N 2012, 'Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android', Informatika, Bandung.

[6] Scheiner, B 1996, 'Applied Cryptography', Wiley, New York.

[7] Soewito, B 2008, 'Characterizing Power and Resource Consumption of Encryption/Decryption in Portable Devices', Region 5 Conference, IEEE, pp. 17-20.

[8] Enck, W, Traynor, P, McDaniel, P & La Porta, T 2005, 'Exploting Open Functionality in SMS-Capable Cellular Networks', Pennsylvania State Univerity, Virgina, dilihat 3 Mei 2015, www.cse.psu.edu/~tfl12/paper/f16-traynor.pdf