

Keamanan Sistem lanjut

EC 7010

Analisa Kinerja Cryptography

Secure Hash Standard

(SHA1, SHA224, SHA256, SHA384, dan SHA512)

pada Digital Signature Standard

(DSA, RSA, dan ECDSA)

Proposal Tugas Akhir

Disusun oleh:

Halga Tamici

23206013



Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

2007

Daftar Isi

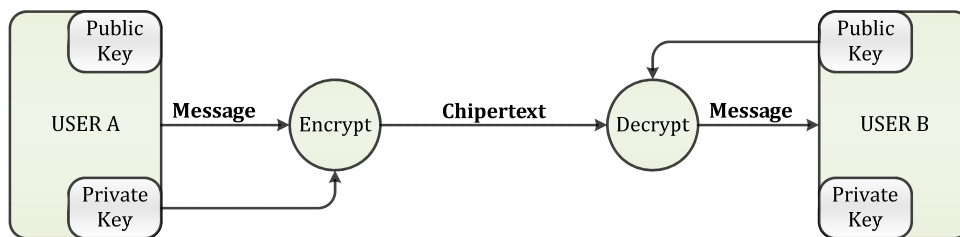
Latar Belakang	3
Tujuan	5
Rumusan Masalah	6
Batasan Masalah	6
Metodologi	6
Referensi	7

1. Latar Belakang

Keamanan pada proses transmisi data telah berkembang cukup pesat. Salah satunya yaitu pada *cryptography*. *Cryptography* merupakan ilmu teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data, *authentication* dan keaslian data. Algoritma *cryptography* merupakan blok penting yang digunakan untuk memberikan keamanan pada jaringan komunikasi umum, seperti internet. Bersamaan dengan peningkatan pada konektivitas wireless dan data rate, keamanan protokol telah dikembangkan beberapa tahun lalu termasuk algoritma *cryptography* yang lebih *resource-friendly*.

Terdapat berbagai macam jenis standarisasi *cryptography* pada FIPS (*Federal Information Processing Standards*) sesuai dengan fungsinya. Salah satu standar *cryptography* pada FIPS yang dapat membangkitkan tanda tangan digital adalah *Digital Signature Standard* (DSS).

Pada kebanyakan *file-file* penting selalu diutamakan informasi identitas *user* pengirim untuk membuat suatu prifasi. Proses ini dilakukan untuk mengubah *signature* menjadi dokumen *digital*. Sehingga pada saat pengaksesan dokumen *digital* tersebut maka terdapat verifikasi pada *signature* yang diberikan.

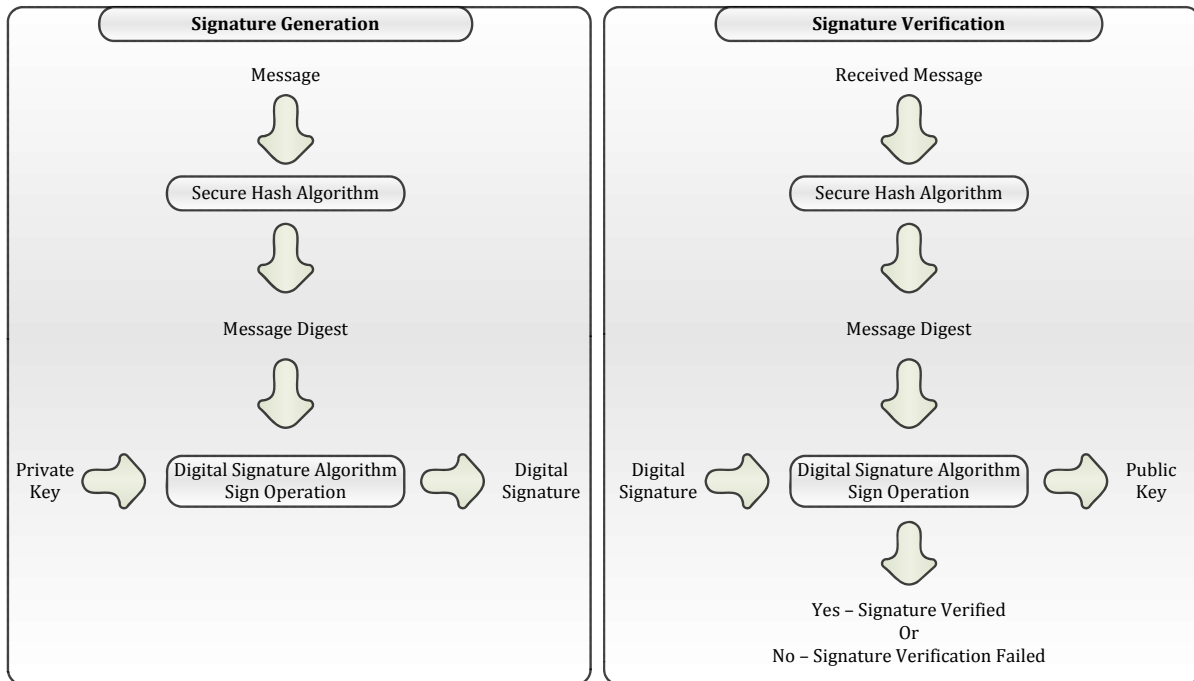


Gambar 1. Proses enkripsi dan dekripsi *public-key*

Digital signature memiliki *key* yang terdiri atas *public key* dan *private key*. *Cryptography public key* dapat mudah diserang dengan peniruan *public-key*, tetapi user bertanggung jawab untuk menjaga *private key* dengan aman. Dari *signature* yang diterima, harus memiliki *public key* yang dibangkitkan oleh *user* pengirim. *Public key* ini yang merupakan informasi *user* pengirim yang terdapat pada *user* penerima yang harus sesuai dengan *signature* yang dihasilkan oleh *user* pengirim.

Menurut FIPS 186 (*Federal Information Processing Standards*), pada bulan Mei 1994, standarisasi DSS yang pertama ditetapkan adalah DSA (*Digital Signature Standard*). Berikutnya pada bulan Mei 1997, NIST (*National Institute of Standards and Technology*) memberitahukan revisi bahwa memperbolehkan penggunaan RSA (Ron Rives, Adi Shamir, dan Len Adleman) dan ECDSA (*Elliptic Curve Digital Signature Algorithm*) sebagai alternative dari DSA. Sehingga terdapat revisi pertama pada FIPS (FIPS 186-1) yang diutarakan bulan Desember 1998, dengan menambahkan RSA seperti ditetapkan pada ANSI X9.31. Dan pada

bulan Januari 2000, revisi kedua (FIPS 186-2), dengan penambahan ECDSA (seperti ditetapkan pada ANSI X9.62) [3][5][6].



Gambar 2. Diagram Blok Sistem Cryptography

Pada implementasinya, suatu sistem pembuatan signature selalu memerlukan *hash function*. *Hash function* biasanya diperlukan bila kita menginginkan pengambilan sidik jari suatu pesan. Sebagaimana sidik jari manusia yang menunjukkan identitas pemilik sidik jari. Fungsi ini diharapkan pula mempunyai kemampuan yang serupa dengan sidik jari manusia, di mana sidik jari pesan diharapkan menunjuk ke satu pesan dan tidak menunjuk kepada pesan lainnya. Fungsi ini juga dinamakan fungsi kompresi karena biasanya, masukan fungsi satu arah ini selalu lebih besar dari pada keluarannya, sehingga seolah-olah mengalami kompresi. Namun kompresi hasil fungsi ini tidak dapat dikembalikan ke asalnya sehingga disebut sebagai fungsi satu arah (*one way function*). Output *hash function* dinamakan *message digest*, karena seolah-olah merupakan inti sari pesan. Padahal tidak demikian. Sebab inti sari pesan mestinya merupakan ringkasan pesan yang masih dapat dipahami maknanya, sedangkan di *hash function* terjadi perlakuan sebaliknya, orang tidak tahu pesan aslinya. Standarisasi *hash function* yang ditetapkan di dalam FIPS adalah SHS (*Secure Hash Standard*) [2][4].

Algorithm	Message Size (bits)	Block Size (bits)	Word Size (bits)	Message Digest Size (bits)	Security (bits)
SHA-1	$<2^{64}$	512	32	160	80
SHA-256	$<2^{64}$	512	32	256	128
SHA-384	$<2^{128}$	1024	64	384	192
SHA-512	$<2^{128}$	1024	64	512	256

Tabel 1. Spesifikasi SHA-1, SHA-256, SHA-384, dan SHA-512

Versi pertama dari SHS yaitu SHA-1 (*Secure Hash Algorithm*) sesuai dari FIPS 180-1 pada bulan April 1997. SHA-1 merupakan jenis *hash function* yang saat ini paling banyak diimplementasikan. Dengan adanya kemajuan teknologi, tentunya kebutuhan ukuran message juga semakin besar begitu juga dengan tingkat ketahanan dari *attacker*. Sehingga NSA (*National Security Agency*) mengembangkan standarisasi SHS sehingga pada bulan Agustus 2002, versi kedua dari SHS terdiri dari beberapa *hash function* yaitu SHA-224, SHA-256, SHA-384, dan SHA-512 [2].

Algorithm	Message Size (bits)	Block Size (bits)	Word Size (bits)	Message Digest Size (bits)	Security (bits)
SHA-224	$<2^{64}$	512	32	224	112

Tabel 2. Spesifikasi SHA-224

2. Tujuan

Tujuan dari tugas akhir ini adalah sebagai berikut:

- ❖ Merancang penggabungan tiap-tiap SHS (SHA-1, SHA-224, SHA-256, SHA-384, dan SHA-512) dan DSS (DSA, RSA, dan ECDSA).
- ❖ Merancang suatu sistem dengan masukan image yang akan dienkripsi danembalikan ke image setelah didekripsi.
- ❖ Membandingkan kinerja tiap-tiap sistem penggabungan SHS dan DSS hasil perancangan.
- ❖ Melakukan analisis performansi dengan masukan jumlah bit yang besar untuk tiap-tiap penggabungan SHS dan DSS.
- ❖ Melakukan analisis statistik terhadap keluaran *cryptography*.

3. Rumusan Masalah

Beberapa permasalahan pada tugas akhir dapat didefinisikan sebagai berikut :

- ❖ Proses pembuatan *message digest* dan parameter *message size*, *block size*, *word size*, *message digest size*, dan *security bits* pada *Secure Hash Standards*.
- ❖ Perbandingan jenis-jenis SHS (SHA-1, SHA-224, SHA-256, SHA-384, dan SHA-512) dan memilih jenis SHS yang terbaik dilihat dari beberapa parameter dan kondisi.
- ❖ Proses pembuatan *signature* (*public key* dan *private key*), *signature verification*, dan parameter yang diperlukan pada *Digital Signature Standards*.
- ❖ Perbandingan jenis-jenis DSS (DSA, RSA, dan ECDSA) dan memilih jenis DSS yang terbaik dilihat dari beberapa parameter dan kondisi.
- ❖ Teknik penggabungan ketiga jenis *Digital Signature* dengan kelima jenis *hash function* dan memilih penggabungan yang terbaik dilihat dari beberapa parameter dan kondisi.
- ❖ Keuntungan dan kerugian pada proses penggabungan tersebut beserta parameter yang mempengaruhinya.
- ❖ Teknik membuat simulasi untuk proses enkripsi dan dekripsi dengan inputan *image*.

4. Batasan masalah

Batasan masalah yang digunakan dalam Tugas Akhir ini adalah:

- ❖ Informasi yang akan dienkripsi dan dekripsi adalah *character* (huruf, angka, atau simbol) dan proses inputan *image*.
- ❖ Proses *Digital Signature Standards* yang digunakan adalah DSA, RSA, dan ECDSA.
- ❖ *Cryptography* RSA yang digunakan adalah *RSA signature*.
- ❖ Proses *hash function* pada *Secure Hash Standard* yang digunakan adalah SHA-1, SHA-224, SHA-256, SHA-384, dan SHA-512.
- ❖ Simulasi dilakukan dengan menggunakan software Matlab 7.0.3.

5. Metodologi

Metodologi penelitian yang dilakukan dalam penyusunan tugas akhir ini meliputi:

- ❖ **Eksplorasi awal**, dilakukan dengan cara melakukan eksplorasi mengenai pengenalan sistem signature cryptography dan hash function yang dapat digunakan untuk security.
- ❖ **Studi literatur**, dilakukan dengan cara mempelajari literatur-literatur baik yang berupa buku (*textbook*), jurnal dan artikel ilmiah, maupun website untuk memahami teknik hash function dan signature cryptography.
- ❖ **Analisis penyelesaian masalah**, dilakukan dengan menganalisis cara kalkulasi data pada algoritma cryptography, beberapa algoritma yang mendukung pembentukan cryptography dan teknik pengacakan pada cryptography.
- ❖ **Perancangan perangkat lunak**, dilakukan dengan cara membuat desain prototipe perangkat lunak yang dapat mengimplementasikan hasil analisis penyelesaian masalah di atas.
- ❖ **Implementasi perangkat lunak**, dilakukan berdasarkan hasil perancangan prototipe perangkat lunak. Penelitian dilakukan dengan bentuk simulasi program dengan

menggunakan software Matlab 7.0.3 yang memungkinkan peneliti memanipulasi variable–variabel input dan meneliti akibatnya terhadap performansi cryptography.

- ❖ **Pengujian perangkat lunak**, pengujian yang dilakukan dengan melihat performansinya dari efisiensi jumlah bit masukan yang sangat besar sampai masukan berupa *image* dengan tidak melupakan keakuratan penggabungan sistemnya.
- ❖ **Analisis hasil dan penarikan kesimpulan**, dari pengujian diatas dapat dilihat efisiensi dari jumlah blok *cryptography* dan berapa lama proses *cryptography* ini dilakukan. Selain dari performansinya sendiri, dilakukan analisis statistik terhadap keluaran *cryptography*. Hal ini dilakukan untuk mengetahui performansi dari *cryptography* tersebut.

6. Referensi

- [1] Coron, J.S. 2002. “*Security Proof for Partial-Domain Hash Signature Schemes*”. Gemplus Card International.
- [2] Federal Information Processing Standards Publication 180–2. 2002. “*Secure Hash Standard*”. National Institute of Standards and Technology.
- [3] Federal Information Processing Standards Publication 186–2. 2000. “*Digital Signature Standard (DSS)*”. National Institute of Standards and Technology.
- [4] Kurniawan, Y. 2004. “*Kriptografi: Keamanan Internet dan Jaringan Komunikasi*”. Bandung: Penerbit Informatika.
- [5] Menezes, A. 2002. “*Evaluation of Security Level of Cryptography: RSA Signature Schemes*”. University of Waterloo.
- [6] Menezes, A.J. and D.B. Johnson. “*Elliptic Curve DSA (ECDSA): An Enhanced DSA*”. University of Waterloo.
- [7] Menezes, A. et al. 2001. “*Evaluation of Security Level of Cryptography: ECDSA Signature Scheme*”. Certicom Research.
- [8] Menezes, A.J., P.C.V. Oorschot, and S.A. Vanstone. 2001. “*Handbook of Applied Cryptography*”. CRC Press.
- [9] Vaudenay, S. “*Digital Signature Schemes with Domain Parameters*”. Ecole Polytechnique Federale de Lausanne.