Penggunaan Digital Signature Standard (DSS) dalam Pengamanan Informasi

Wulandari – NIM : 13506001

Program Studi Teknik Informatika ITB, Jl Ganesha 10, Bandung, email: if16001@students.if.itb.ac.id

Abstract *– Makalah ini membahas tentang penggunaan dan penerapan salah satu cabang ilmu dari kriptografi, yaitu Digital Signature Standard(DSS), sebuah mekanisme yang didasari olehrancangan kunci public ElGamal.*

DSS merupakan sebuah bakuan (standard) untuk tanda tangan digital yang diresmikan pada bulan Agustus 1991oleh NIST (The National Institute of Standard and Technology).

Dalam pengaplikasiannya, DSS menggunakan dua komonen, yaitu Algoritma tanda-tangan digital, Digital Signature Algorithm (DSA), dan fungsi hash SHA (Secure Hash Algorithm).

DSA termasuk ke dalam system kriptografi kunci-publik. Pada umumnya , public-key-cryptosystems memiliki dua fungsi utama, enkripasi dan tanda tangan digital. Meskipun demikian, DSA tidak dapat digunakan untuk enkripsi. DSA mempunyai dua fungsi utama :

1. *Pembentukan sidik dijital (signature generation), dan*
2. *Pemeriksaan keabsahan sidik dijital (signature verivication).*

Sebagaimana halnya pada algoritma kriptografi kunci-publik, DSA menggunakan dua buah kunci, yaitu kunci publik dan kunci rahasia. Kunci publik dipublikasikan, sedangkan kunci rahasia disimpan. Pembentukan sidik dijital menggunakan kunci rahasia pengirim, sedangkan verifikasi sidik dijital menggunakan kunci public pengirim.

Disamping itu, DSA juga menggunakan fungsi hash SHA (Secure Hash Algorithm) untuk mengubah pesan menjadi message digest yang berukuran 160 bit

1. PENDAHULUAN

Tanda tangan merupakan salah satu identitas diri yang dimiliki setiap orang. Karena keunikannya, tanda tangan digunakan sebagai salah satu alat untuk memvalidasi suatu kesepakatan atau dokumen.

Namun, seiring dengan berkembangnya teknologi dan era digital, terjadi perpindahan proses legalisasi. Legalisasi yang sebelumnya didokumentasikan dalam media kertas, yang dalam banyak hal menggunakan tanda tangan, semakin lama semakin jarang digunakan. Sebaliknya, masyarakat cenderung lebih menyukai kesepakatan – kesepakatan melalui media digital. Hanya saja, melalui media digital kebutuhan akan adanya suatu jaminan bahwa informasi dapat terjaga keaslian dan keutuhannya belum begitu terpenuhi. Padahal, sangatlah penting untuk memastikan bahwa dokumen atau kesepakatan yang dibuat tidak mengalami perubahan oleh pihak – pihak yang tidak berhak.

Hal ini menunjukkan bahwa perlu adanya proses legalisasi digital yang dapat memenuhi aspek – aspek dan kriteria – kriteria di atas. Proses legalisasi ini dapat diterapkan dengan menggunakan tanda tangan dijital (digital signature).

2. ENKRIPSI KUNCI PUBLIK

Konsep kunci publik atau public-key cryptosystems diperkenalkan pertama kali oleh Whitfield Diffie dan Martin Hellman pada 1976. Public-key cryptosystems memiliki dua fungsi utama, enkripsi dan tanda tangan digital.

Pada sistemnya, terdapat sepasang kunci, kunci publik dan kunci rahasia. Kunci publik dipublikasikan, sedangkan kunci rahasia disimpan.

Semua komunikasi yang dikirim ataupun diterima hanya mencakup kunci publik, kunci rahasia tidak pernah ditransmisikan atau dipakai bersama. Pada sistem ini, prngirim dan penerima tidak memerlukan lagi segi keamanan alat-alat komunikasi. Yang perlu diperhatikan hanyalah bagaimana kunci publik dapat diasosiasikan dengan pngirim maupun penerima dengan cara yang dapat dipercaya.

Setiap orang dapat mengirimkan pesan rahasia hanya dengan menggunakan kunci publik, tetapi pesan hanya dapat didekripsi dengan kunci privat, yang merupakan milik penerima. Pada umumnya, kunci rahasia selalu dihubungkan secara matematis dengan kunci publik. Karena itu, dapat terdapat kebocoran dalam sistem kunci publik. Jalan terbaik yang dapat dilakukan adalah dengan menurunkan kunci rahasia sesulit

2.1 RSA

RSA cryptosystem adalah public-key cryptosystem yang menawarkan baik enkripsi maupun tanda tangan digital (otentikasi). System RSA dikembangkan oleh Ronald Rivest, Adi Shamir, dan Leonard pada tahun 1977.

Algoritma RSA bekerja seperti berikut:

1. ambil dua bilangan prima besar, *p* dan *q*
2. hitung hasil kalin kedua bilangan tersebut, *n* = *pq*; *n* disebut dengan modulus.
3. Pilih sebuah bilangan, *e*, yang lebih kecil dari *n* dan merupakan bilangan prima secara relatif dari (*p*-1)(*q*-1), yangartinya *e* dan (*p*-1)(*q*-1) tidak memiliki FPB lain kecuali 1.
4. temukan bilangan lain *d* sehingga (*ed* - 1) dapat dibagi dengan (*p*-1)(*q*-1).
5. Nilai-nilai *e* dan *d* masing-masing disebut eksponen publik dan privat. Kunci publik adalah pasangan (*n*, *e*); kunci privat adalah (*n*, *d*).

Faktor *p* dan *q* dapat dihancurkan atau disimpan dengan kunci privat. Sulit untuk mendapatkan kunci privat *d* dari kunci publik (*n*, *e*).
Jika seseorang dapat memfaktorkan *n* menjadi *p* dan *q*, maka ia bisa mendapatkan kunci privat *d*. Sehingga keamanan sistem RSA berdasar pada asumsi bahwa pemfaktoran sulit dilakukan.

2.2 Diffie Hellman key exchange

Diffie-Hellman key exchange adalah metode dimana subyek menukar kunci rahasia melalui media yang tidak aman tanpa mengekspos kunci. Metode ini diperlihatkan oleh Dr. W. Diffie dan Dr. M. E. Hellman pada tahun 1976 pada papernya "New Directions in Cryptography". Metode ini memungkinkan dua pengguna untuk bertukar kunci rahasia melalui media yang tidak aman tanpa kunci tambahan.

Metode ini memiliki dua parameter sistem, *p* dan *g*. Kedua parameter tersebut publik dan dapat digunakan oleh semua pengguna sistem. Parameter *p* adalah bilangan prima, dan paramater *g* (sering disebut generator) adalah integer yang lebih kecil dari *p* yang memiliki properti berikut ini:
Untuk setiap bilangan *n* antara 1 dan *p-1* inklusif, ada pemangkatan *k* pada *g* sehingga $g^k = n \bmod p$ mungkin.

2.1 Enkripsi Kunci Publik El-Gamal

Konsep El-Gamal berasal dari perluasan konsep Diffie-Hellman untuk diterapkan pada enkripsi dan tanda tangan digital. Sistem El Gamal adalah public-key cryptosystem yang berdasar pada masalah logaritma diskret. Enkripsi kunci publik El-Gamal inilah yang digunakan dalam pembuatan tanda tangan digital.

3. FUNGSI SATU ARAH / FUNGSI HASH

Fungsi satu arah atau one-way function adalah fungsi matematis yang secara signifikan lebih mudah dihitung pada satu arah (arah maju) ketimbang arah sebaliknya (inverse). Sebagai perbandingan, pada perhitungan fungsi dengan arah maju dapat diselesaikan dalam beberapa detik, sedangkan untuk arah inverse proses perhitungan dapat memakan waktu berbulan-bulan atau bertahun-tahun.

Trapdoor one-way function adalah fungsi satu arah dimana arah inversnya mudah diberikan sebuah informasi (trapdoor), tetapi sulit untuk melakukan hal sebaliknya. Public-key cryptosystems berdasarkan pada trapdoor one-way functions. Kunci publik memberikan informasi tentang instans tertentu dari fungsi, kunci privat memberikan informasi tentang trapdoor. Jika seseorang mengetahui trapdoor dapat dengan mudahnya menghitung fungsi dalam dua arah, Sebaliknya, jika trapdoor tidak diketahui, seseorang hanya dapat menjalankan fungsi dengan mudah pada arah maju. Arah maju digunakan untuk enkripsi dan verifikasi tandatangan, arah invers digunakan untuk dekripsi dan pembuatan tandatangan.

Fungsi hash adalah fungsi yang memproduksi output dengan panjang tetap dari input yang berukuran variabel. Output dari fungsi hash disebut dengan message digest. Fungsi hash memiliki karakteristik fungsi satu arah karena file asli tidak dapat dibuat dari message digest.

3.1 Message Digest 5 (MD5)

MD5 merupakan algoritma message digest yang dikembangkan oleh Ronald Rivest pada tahun 1991. MD5 mengambil pesan dengan panjang sembarang dan menghasilkan message digest 128 bit. Pada MD5 pesan diproses dalam blok 512 bit dengan empat round berbeda.

3.2 Secure Hash Algorithm (SHA)

Secure Hash Algorithm (SHA) adalah algoritma yang dispesifikasikan dalam Secure Hash Standard (SHS, FIPS 180), yang dikembangkan oleh NIST. SHA-1 adalah revisi terhadap SHA yang dibuat oleh pemerintah Amerika dan dipublikasikan pada tahun 1994, serta menjadi standar pada tahun 1995. Revisi ini mengkoreksi kekurangan – kekurangan yang terdapat pada SHA. Desainnya mirip dengan fungsi hash MD4 yang dikembangkan oleh Rivest. SHA-1 juga dideskripsikan dalam standar ANSI X9.30

Algoritma SHA mengambil pesan yang panjangnya kurang dari 2^64 bit dan menghasilkan message digest 160-bit atau 20 byte.
Sebagai contoh apabila terdapat sebuah pesan **M** dan hasil hash **H(M)** yang dikirimkan, maka di penerima bisa melakukan hashing dari pesan **M** untuk menghasilkan **H'(M)**. Apabila hasil **H(M)** dan **H'(M)**

sama maka bisa dipastikan bahwa penerima menerima pesan **M** yang sama. Satu-satunya cara untuk melindungi pesan dari perubahan adalah dengan mengamankan hasil hash (**H(M)**).

Pengamanan terhadap hasil hash dapat dilakukan dengan menambahkan informasi rahasia yang hanya diketahui oleh pihak pengirim dan penerima dan ditambahkan ke dalam pesan sebelum dilakukan hash. Misalnya **H(S+M)**, dengan **S** adalah informasi rahasia yang diketahui bersama, maka ketika pesan didapatkan dan dengan **S** yang dimiliki bisa dilakukan dengan membuat **H'(S+M)**. Digest yang baru yang dihasilkan disebut sebagai HMAC atau Hashed Messsage Authentication Code.

3.3 Hashed Messsage Authentication Code

HMAC adalah algoritma hash yang menggunakan sebuah kunci untuk menghasilkan Message Authentication Code (MAC). MAC adalah tag otentikasi (disebut juga checksum) yang diturunkan dengan bersama dengan sebuah kunci rahasia ke sebuah pesan. MAC dihasilkan sebelum pesan dikirim, ditambahkan pada pesan, keduanya kemudian ditransmisikan. Pada sisi penerima, MAC dihasilkan dari pesan sendiri dengan menggunakan algoritma yang sama seperti yang digunakan pengirim dan MAC ini dibandingkan dengan MAC yang dikirim bersama pesan. Jika mereka tidak sama, berarti pesan telah dimodifikasi dalam perjalanannya.

Algoritma hashing dapat digunakan untuk menghasilkan MAC dan algoritma MAC menggunakan kunci yang menyediakan perlindungan yang lebih kuat bila dibandingkan dengan tidak menggunakan algoritma ini.

Pada contoh yang sama seperti point 3.2, jika HMAC digunakan, maka kekuatan pengamanan terhadap data bergantung pada ketidakmampuan penyerang untuk mengetahui **S**. Oleh karena itu **S** harus sesuatu yang tidak mudah ditebak dan sesuatu yang harus sering berubah. Salah satu cara terbaik yang dapat digunakan adalah dengan menggunakan Kerberos.

Dalam Kerberos, suatu sentral memberikan suatu kunci sementara yang digunakan seperti tiket pada saat ada 2 pihak yang akan berkomunikasi. Pengirim akan mendapatkan tiket untuk mengirimkan informasi kepada penerima tertentu. Pada saat mengirimkan informasi, pengirim akan membuka tiketnya dan mengambil **S**, mengirimkan pesan **M**, HMAC nya dan tiket milik penerima. Penerima harus membuka tiket menggunakan password yang sudah didaftarkan dalam Kerberos dan mengambil **S** dan informasi mengenai identitas
pengirimnya. Kemudian penerima akan mengambil pesan **M** dan membuat **H'(S+M)** dan mengecek kecocokannya. Apabila cocok maka penerima akan mengetahui bahwa pengirim yang mengirimkan pesan/informasi dan identitas pengirim diberitahu oleh Kerberos.

Hanya saja, metode HMAC ini masih memiliki beberapa kekurangan, diantaranya :

(1) HMAC menggunakan kunci simetrik sehingga sulit untuk mempertukarkan kunci simetrik antara pengirim dan penerima karena memerlukan jaringan khusus dan aman untuk pertukaran.
(2) Kunci simetrik hanya bisa dilakukan untuk 2 pihak (pengirim dan penerima) sehingga untuk mengirimkan pesan lebih ke 1 penerima maka akan timbul masalah keamanan yang lain yaitu bahwa kunci diketahui lebih dari 2 orang dan dapat dengan mudah menyebar kepada pihak yang tidak diinginkan.

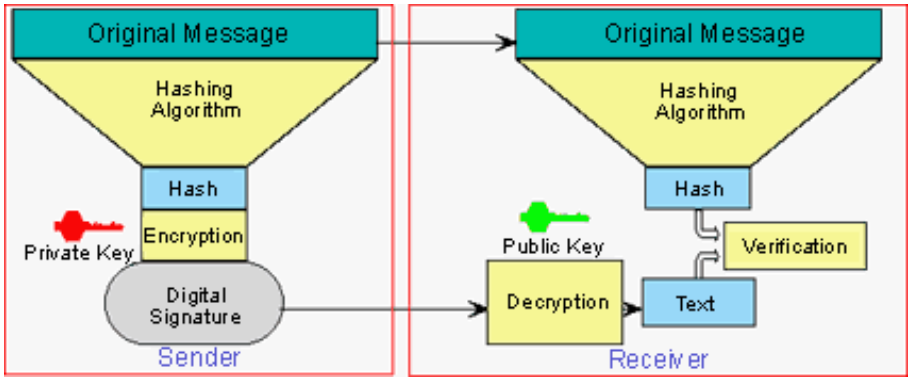
Metode lain untuk memproteksi digest adalah dengan menggunakan kriptografi kunci publik, seperti RSA yang telah dijelaskan pada bagian sebelumnya.

4. TANDA TANGAN DIGITAL

Prinsip yang digunakan dalam tanda tangan digital adalah data yang dikirimkan harus ditanda tangani oleh pengirim dan tanda tangan bisa diperiksa oleh penerima untuk memastikan keaslian data yang dikirimkan. Proses ini tidak jauh berbeda dengan proses penandatanganan dokumen kertas yang biaa kita lakukan. Dengan cara ini, pengirim bertanggung jawab terhadap isi dokumen dan dapat dicek keaslian dokumen oleh penerima.Tanda tangan digital dapat memberikan pengecekan integritas pesan. Jika salah satu byte dari pesan berubah maka tanda tangan digital juga akan berubah.

Proses yang dilakukan dalam pembuatan tanda tangan digital pertama kali akan dilakukan hashing terhadap pesan yaitu pengubahan dari pesan asli menjadi suatu pesan dengan ukuran tertentu yang disebut sebagai digest.

Setelah message digest dihitung, kemudian message digest dienkripsi dengan kunci privat pengirim. Penerima kemudian mendekripsi message digest dengan menggunakan kunci publik pengirim. Jika kunci publik ini membuka message digest dan benar bahwa ia merupakan kunci publik pengirim, verifikasi pengirim telah tercapai. Verifikasi terjadi karena hanya kunci publik pengirim yang dapat mendeskripsikan message digest yang dienkripsi dengan kunci privat pengirim.Setelah itu, penerima dapat menghitung message digest dari file yang diterima menggunakan fungsi hash yang identik dengan pengirim. Jika message digest identik dengan message digest yang dikirim sebagai bagian dari tanda tangan, maka pesan tidak dimodifikasi atau dapat dipastikan keasliannya.



Gambar 1. Pembuatan Tanda Tangan Digital

5. DIGITAL SIGNATURE STANDARD (DSS)

National Institute of Standard and Technology (NIST) menerbitkan standar untuk Digital Signature yang dikenal sebagai Digital Signature Standard (DSS) dan diterbitkan sebagai Federal Information Processing Standard (FIPS) PUB 186 tahun 1991, dan direvisi tahun 1993 dan 1996.

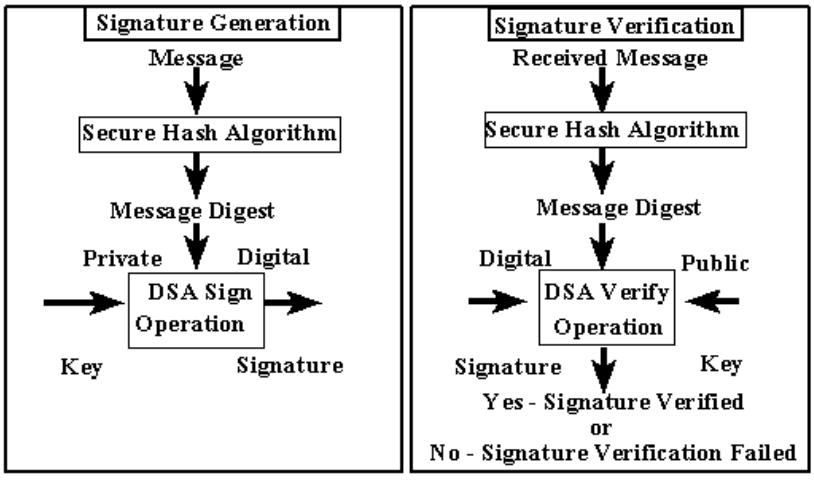
Standar ini memungkinkan penggunaan algoritma tanda tangan digital RSA atau Digital Signature Algorithm (DSA). DSA berdasar pada modifiksi metodologi tanda tangan digital El Gamal dan dikembangkan oleh ClausSchnorr. Kedua algoritma tanda tangan digital ini menggunakan Secure Hash Algorithm(SHA-1) seperti terdefinisi dalam FIPS 180. Standar menggunakan SHA-1 ini disebut SHS (Secure Hash Standard).

6. PROSES PERHITUNGAN DSA

Proses perhitungan DSA dapat diklasifikasikan menjadi 3 prosedur, antara lain : prosedur pembangkitan sepasang kunci, prosedur pembangkitan tanda tangan, dan prosedur verifikasi keabsahan tanda tangan.

Adapun secara rinci proses – proses tersebut antara lain :

a. Prosedur pembangkitan sepasang kunci
   1. Pilih bilangan prima p dan q yang dalam hal ini $(p\text{-}1) \bmod q = 0$
   2. Hitung $g = {}^{h\,(p-1)/q} \bmod p$, yang dalam hal ini $1 < h < (p\text{-}1)$ dan $h^{(p-1)/q} \bmod p > 1$
   3. Tentukan kunci rahasia x dimana $x < q$
   4. Hitung kunci publik $y = g^x \bmod p$
b. Prosedur pembangkitan tanda tangan
   1. Hitung nilai hash $(H(m))$ dari pesan
   2. Tentukan bilangan acak k dimana $k < q$
   3. Hitung r dan s, dimana :
      $r = (g^k \bmod p) \bmod q$
      $s = (k^{-1} (H(m) + x * r)) \bmod q$
   4. Kirim pesan m dan tanda-tangan r dan s
c. Prosedur verifikasi keabsahan tanda tangan
   1. Hitung :
      $w = s^{-1} \bmod q$
      $u_1 = (H(m) * w) \bmod q$
      $u_2 = (r * w) \bmod q$
      $v = ((g^{u1} * y^{u2}) \bmod p) \bmod q)$
   2. Jika $v = r$, maka tanda tangan sah



Gambar 2. Prosedur Pembangkitan Tanda Tangan (kiri) dan Prosedur Verifikasi Keabsahan Tanda Tangan (kanan)

7. KESIMPULAN

Kesimpulan yang dapat ditarik adalah berkembangnya Teknologi dan era digital bukan berarti masyarakat kehilangan cara untuk menentukan kelegalan / leabsahan dari suatu dokumen. Dengan adanya konsep tanda tangan digital merupakan salah satu acara yang paling efektif untuk menghilangkan segala kekhawatiran akan perubahan dokumen yang dapat dilakukan oleh pihak – pihak yang tidak berhak.

DAFTAR REFERENSI

[1] http://kur2003@if.itb.ac.id
[2] http://www.total.or.id
[3] http://www.kriptonesia.com
[4] http://www.wikipedia.com
[4] http://bebas.vlsm.org