STUDI MENGENAI GROUP BLIND DIGITAL SIGNATURE

Ridwan – NIM : 13503072

Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail : if13072@students.if.itb.ac.id

Abstrak

Group digital signature scheme adalah metode yang membuat seorang anggota sebuah grup untuk secara *anonymous* menanda-tangani sebuah pesan sebagai wakil dari grup. Konsep ini pertama kali diperkenalkan oleh David Chaum dan Eugene van Heyst pada 1991. Sebagai contoh, skema tandatangan kelompok digital dapat digunakan oleh seorang karyawan dari sebuah perusahaan yang besar yang mana cukup bagi pihak yang akan memastikan keaslian pesan dengan hanya mengetahui pesan tersebut telah ditandatangani oleh seorang karyawan, tetapi tidak secara khusus tahu karyawan mana yang menandatangani pesan tersebut

Blind signature, seperti yang telah diperkenalkan juga oleh David Chaum, adalah salah satu bentuk dari *digital signature* yang mana pesan dalam berkas yang akan diberikan tandatangan digital akan disamarkan sebelum berkas tersebut ditandatangani. Hasil dari *blind signature* dapat di verifikasi di publik untuk dibandingkan dengan berkas aslinya, pesan yang tidak tersamarkan seperti yang digunakan dalam digital signature yang umum. *Blind signatures* umumnya digunakan untuk protokol yang berhubungan dengan prifasi dimana pemberi tandatangan dan pembuat pesan adalah pihak yang berbeda.

Studi yang akan dilakukan pada makalah ini adalah mengenai *Group Blind Digital Signature*. Group *Blind Digital Signature* menggabungkan *Blind Signature* dan *Group Digital Signature*. *Group Blind Digital Signature* memungkinkan anggota dari sebuah grup yang besar untuk memberi tandatangan digital mewakili kelompok tersebut. *Group Blind Digital Signature* sangat berguna di berbagai aspek dalam *e-commerce*.

Kata kunci: Group digital signature, Blind signature, Group Blind Digital Signature

1. Pendahuluan

(Keamanan oleh Kriptografi)

Selain dengan merahasiakan isi pesan dengan suatu teknik kriptografi, kriptografi juga digunakan untuk menangani masalah keamanan yang mencakup dua hal berikut [2]:

- Keabsahan pengirim (*user authentication*).
  Hal ini berkaitan dengan kebenaran identitas pengirim. Dengan kata lain, masalah ini dapat diungkapkan sebagai pertanyaan: "Apakah pesan yang diterima benar-benar berasal dari pengirim yang sesungguhnya?"
- Keaslian pesan (*message authentication*).
  Hal ini berkaitan dengan keutuhan pesan (*data integrity*). Dengan kata lain, masalah ini dapat diungkapkan sebagai pertanyaan: "Apakah pesan yang diterima tidak mengalami perubahan (modifikasi)?"
- Anti-penyanggahan (*nonrepudiation*).
  Pengirim tidak dapat menyangkal (berbohong) tentang isi pesan atau ia yang mengirimkan pesan. Masalah ini masih berkaitan dengan dengan masalah pertama dan kedua. Jika keabsahan pengirim dan keaslian pesan dapat diverifikasi, maka pengirim tidak dapat melakukan sanggahan terhadap pesan yang dikirim.

Ketiga masalah ini dapat diselesaikan dengan teknik otentikasi (*authentication*).

Teknik otentikasi (dalam komunikasi data) adalah prosedur yang digunakan untuk membuktikan keaslian pesan atau identitas pemakai.

Sebenarnya, algoritma kriptografi simetri sudah memberikan solusi untuk masalah keamanan pertama dan kedua, karena kunci simetri hanya

diketahui oleh pengirim dan penerima. Jadi, jika *B* menerima pesan dari *A*, maka ia percaya pesan itu dari *A* dan isinya tidak mengalami perubahan, karena tidak ada orang lain yang mengetahui kunci selali mereka berdua.

Namun, algoritma kriptografi simetri tidak dapat menyediakan cara untuk mengatasi masalah keamanan yang ketiga, yaitu jika salah satu dari dua pihak, *A* dan *B*, membantah isi pesan atau telah mengirim pesan.

2. Tanda Tangan Digital

Tanda tangan digital adalah istilah yang sering bermakna ambigu. Kadang, istilah ini diartikan sebagai bagian dari tanda tangan elektronik. Tetapi sebagian orang menggunakan istilah ini sebagai sesuatau yang sejajar dengan tanda tangan elektronik. U.S. Electronic Signatures in Global and National Commerce Act yang diselenggarakan pada tahun 2000 menggunakan tanda tangan elektronik saat tanda tangan digital didiskusikan, mengilustrasikan kebingungan secara legal. Tanda tangan elektronik berarti sebuah suara elktronik, symbol, atau proses, secara logic dihubungkan dengan kontrak atau lainnya dan dieksekusi atau diadopsi oleh seseorang dengan maksud menandai rekaman tersebut.

Tanda tangan digital berarti sebuah tanda tangan yang berbasiskan kepada skema kriptografi. Banyak yang sudah diajukan, beberapa sudah ditemukan dan sudah tidak digunakan lagi. Beberapa sudah dipatenkan, beberapa paten sudah tidak berlaku lagi, dan ada beberapa perbedaan pendapat dalam hal ini dengan tujuan komersil.

Tanda tangan digital ini menggunakan algoritma kunci nirsimetri dan biasanya menggunakan skema *public key infrastructur* (PKI) dimana kunci publik yang digunakan dalam skema tanda tangan terikat dengan pengguna oleh sebuah setifikat identitas digital yang dikeluarkan oleh sebuah organisasi yang memiliki otoritas untuk mengeluarkan sertifikat, biasanya dikelola oleh perusahaan komersil pihak ketiga. Sistem PKI memiliki tujuan untuk menyamarkan informasi pengguna (nama, alamat, no telepon, ...) menjadi sebuah kunci publik, ide dasarnya cukup dekat dengan tugas notaris. Ada beberapa skema tanda tangan, yang umumnya menggunakan dua algoritma, satu untuk memberi tanda tangan dan yang lainnya untuk melakukan pengecekan keaslian tanda tangan tersebut. Keluaran dari proses tanda tangan disebut juga dengan tanda tangan digital.

2.1 Keuntungan Tanda Tangan Digital

Ada tiga alasan utama untuk memasukan tanda tangan digital dalam komunikasi.

2.1.1 Authentication

Sistem kriptografi kunci public memperbolehkan pengenkripsian pesan dengan kunci privat pengguna. Pesan itu sendiri butuh untuk dikirimkan dalam bentuk cipherteks. Bila hash dari sebuah dokumen di bangun dan dienkripsi, dokumen tersebut tidak dapat di deteksi kecuali dengan mengubah nilai hashnya, yang mana, bila algoritma yang berkualitas digunakan, secara komputasi tidak dapat dilakukan. Dengan mendekripsi fungsi hash menggunakan kunci public yang dimiliki pengirim, dan mengecek hasilnya dengan hash yang baru dibangun, pihak penerima dapat menkonfirmasi bahwa enkripsi tersebut dibuat dengan kunci privat pengirim, dan pesan tersebut tidak diubah atau ditambahkan sejak pesan tersebut diberikan tanda tangan. Penerima tidak akan bisa yakin bahwa yang memberikan tanda tangan adalah benar-benar orang yang seharusnya mengirim pesan tersebut.

Hal yang penting dari keyakinan akan athentifikasi adalah tentunya mengenai hal-hal yang berhubungan dengan financial. Sebagai contoh, andaikan kantor cabang dari sebuah bank mengirim perintah kepada bank pusat dalam bentuk (a,b) dimana a adalah nomer account dan b adalah jumlah yang akan diberikan pada account tersebut. Pelanggan yang penuh tipu daya mungkin akan mendepositkan Rp100.000, mengobservasi hasil transmisi dan pengulangan transmit (a,b), mendapatkan deposit setiap saat dan menjadi kaya dengan cepat. Ini adalah contoh dari *replay attack*.

2.1.2 Integrity

Kedua belah pihak akan selalu menginginkan untuk yakin bahwa pesan yang dikirim tidak diubah atau ditambah-tambahkan dalam pengiriman. Enkripsi dari pesan membuatnya menjadi sulit untuk membacanya, namun pihak ketiga masih dapat menambahkan pesan tersebut tanpa membacanya. Sebagai contoh adalah serangan *homomorphism*

2.1.3 Non-repudiation

Dalam konyeks kriptografi, kata repudisi bermaksud pada pernyataan tidak memiliki tanggung jawab terhadap pesan tersebut. Penerima pesan harus memaksa sang pengirim untuk membeikan tanda tangannya dalam pesan yang akan dikirimkan untuk mempersulit *repudition*, selama sang penerima dapat menunjukkan tenda tangan tersebut pada pihak ketiga. Bagaimanapun, kehilangan kunci privat berarti semua tanda tangan digital yang menggunakan kunci tersebut adalah mencurigakan. Menyadari bahwa kemungkinan kehilangan kunci tersebut dapat terjadi bukanlah bagian dari permasalahan kriptografi, melainkan *human error*, dan tidak ada solusinya.

2.2 Implementasi dari Tanda Tangan Digital kunci publik

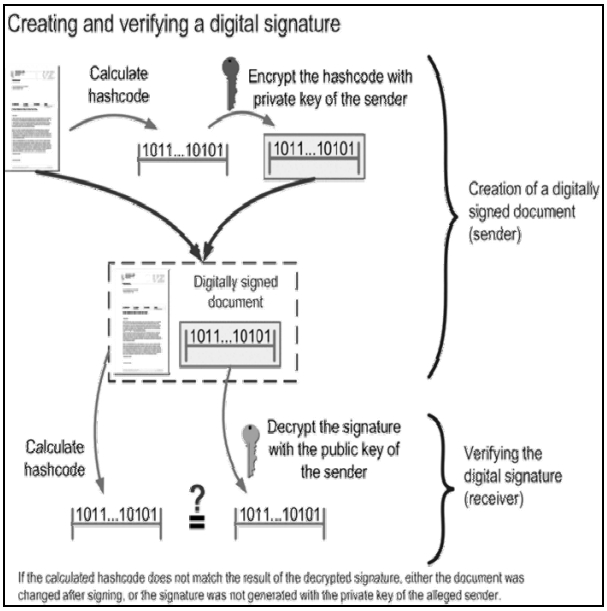
Skema tanda tangan digital kunci publik bergantung pada kriptografi kunci publik. Dalam kriptografi kunci publik, setiap pengguna memiliki sepasang kunci: satu bersifat public dan yang satunya bersifat private. Kunci yang bersifat public didistribusikan, akan tetapi kunci yang bersifat private dirahasiakan oleh pengguna, kebutuhan lainnya adalah kunci ini tidak boleh dapat ditemukan secara komputasi apabila diketahui kunci publiknya.

Umumnya, tanda tangan digital mencakupi tiga algoritma:

- Algoritma pembangkit kunci
- Algoritma pemberi tanda tangan
- Algoritma varifkasi tanda tangan

Sebagai contoh, bayangkan sebuah situasi dimana Bob mengirimkan pesan kepada Alice dan Alice ingin yakin bahwa pesan tersebut benar-benar dari Bob. Bob mengirimkan pesan ke Alice, memasukkan tanda tangan digital. Tanda tangan digital tersebut dibangun dari kunci privat Bob, dan berbentuk *Sring of bits*. Pada pihak penerima, Alice dapat memeriksa apakah benar pesan tersebut dating dari Bob atau bukan dengan menjalankan algoritma verifikasi pada pesan tersebut sekaligus dengan tanda tangan digitalnya, dengan menggunakan kunci publik Bob. Apabila cocok, maka Alice dapat yakin bahwa pesan tersebut benar-benar dari Bob.

2.3 Hubungan Tanda Tangan Digital dengan Enkripsi



Tanda tangan digital menggunakan tahnik enkripsi namun algoritma yang digunakan belum tentu langsung cocok dengan keperluan untuk membuat tanda tangan digital, banyak metode-metode yang lebih efesien yang dapat digunakan. Tentu saja sebuah dokumen dapat saja dikirimkan dalam bentuk terenkripsi melalui jalur komunikasi public seperti pesan-pesan lainnya.

Yang lebih umum terjadi, Bob pertama-tama menyodorkan fungsi hash kriptografi yang berkualitas pada pesan tersebut, lalu secara digital memberikan tanda pada hasil fungsi hash. Sebuah hash yang tidak aman dapat menurunkan keamanan tanda tangan digital. Sebagai contoh, apabila dimungkinkan untuk membangun kekacauan hash, dapat dimungkinkan untuk memalsukan tanda tangan digital.

Ada beberapa alasan untuk memberikan tanda pada hash yang baik (atau *message digest*) daripada ke seluruh dokumen.

- **Untuk efisiensi :** Tanda tangan akan menjadi jauh lebih pendek dan menyimpan banyak waktu karena hash dibangun dengan jauh lebih cepat daripada memberikan tanda tangan.
- **Dokumen ditujukan untuk dibaca oleh orang lain :** sebagai contoh, diploma, sertifikat identitas, sertifikat

kelahiran, SIM, dll. Dokumen ini harus berupa plainteks, namun mengandung tanda tangan digital. Tanda tangan digital tersebut dapat digunakan untuk memeriksa bahwa dokumen tersebut asli atau tidak diubah-ubah.

- **Untuk integritas**

2.3.1 Menggunakan kunci yang terpisah untuk memberi tanda dan enkripsi

Di beberapa negara, tanda tangan digital memiliki status seperti tanda tangan biasa. Pada umumnya, ini berarti bahwa di tempat tersebut tanda tangan digital secara legal menyamarkan pemberi tanda tangan pada dokumen. Untuk alasan tersebut, seringkali adalah lebih baik untuk menggunakan kunci yang terpisah untuk melakukan enkripsi dan pemberian tanda tangan. Menggunakan sepasang kunci enkripsi, seseorang dapat melakukan percakapan tetapi tidak usah secara legal memberi tanda pada setiap pesan. Hanya saat kedua pihak sepakat mereka memberikan tanda tangan dengan kunci tanda tangan mereka. Setelah memberi tanda tangan, dokumen dapat dikirimkan melalui link yang terenkripsi.

2.4 Hubungan Tanda Tangan Digital dengan *Trusted Time Stamping*

Algoritma dan protocol tanda tangan digital tidak secara menurun menyediakan kepastian mengenai tanggal dan waktu saat pemberian tanda tangan. Pemberi tanda tangan mungkin, atau mungkin tidak, sudah mengikut sertakan waktu pemberian tanda tangan dalam tanda tangan, atau pada dokumen itu sendiri, tetapi penerima dokumen tersebut tidak dapat yakin apabila pemberi tanda tangan tidak melakukannya. Isu ini dapat diselesaikan dengan *trusted time stamping* untuk menambahkan tanda tangan digital.

2.5 Penambahan keamanan

2.5.1 Memasukkan kunci privat dalam sebuah *smart card*

Semua system kriptografi kunci public bergatung sepenuhnya pada penjagaan kerahasiaan kunci private. Ebuah kunci prifat dapat disimpan di komputer dan dijaga oleh sebuah password. Namun hal ini memiliki dua kelemahan :

- Pengguna hanya dapat memberikan tanda tangan di komputer tersebut.
- Keamanan dari kunci privat bergantung pada keamanan dari komputer tersebut, yang mana tidak dapat diandalkan untuk sebagian besar komputer dan Sistem Operasi.

Alternatif yang lebih aman adalah menyimpan kunci prifat tersebut dalam sebuah *smart card*. Banyak *smart card* yang telah didesain untuk memiliki keamanan lebih. Dalam implementasinya, hash di kalkulasi dari dokumen dan dikirim ke *smart card*, yang mana CPU mengenkripsi hash dengan kunci prifat yang disimpan dan mengembalikanya. Biasanya, pengguna harus mengaktifasi *smart card* miliknya dengan memasukkan PIN. Patut dicatat bahwa kunci prifat dapat saja tidak pernah meninggalkan *smart card* tersebut. Bila *smart card* tersebut dicuri, pencurinya akan memerlukan PIN untuk membangun tanda tangan digital.

2.5.2 Menggunakan pembaca *smart card* dengan keyboard yang terpisah

Memasukkan PIN untuk mengaktifasi *smart card*, biasanya membutuhkan *keypad* nomorik. Beberapa pembaca *smart card* memiliki *keypad* miliknya sendiri. Ini lebih aman daripada menggunakan pembaca yang menyatu dengan PC lalu memasukkan PIN melalui keyboard komputer. Komputer dapat saja menjalankan *keystroke logger* (sebuah perangkat lunak untuk menyimpan ketikan pada keyboard) sehingga PIN menjadi menyebar. Pembaca yang khusus lebih aman dalam melawan serangan melalui perangkat lunak seperti yang telah disebutkan sebelumnya.

2.5.3 Menggunakan Tanda Tangan Digital hanya pada aplikasi yang dapat dipercaya

Satu perbedaan utama dari tanda tangan digital dengan tanda tangan yang biasa adalah bahwa pemberi tanda tangan tidak dapat “melihat” benda yang ia beri tanda tangan. Aplikasinya lah yang merepresentasikan kode hash untuk di enkripsi dengan kunci prifat, tapi untuk kasus aplikasi yang buruk, bias saja pemberi melihat tanda tangan digitalnya telah dimasukkan pada dokumen yang ia inginkan, namun sebenarnya ia memberi tanda tangan pada dokumen lain.

2.6 Beberapa algoritma tanda tangan digital

- **Full Domain Hash, berbasis pada RSA**

  Dalam kriptografi, **Full Domain Hash (FDH)** adalah skema tanda tangan yang berbasiskan pada RSA yang mengikuti paradigma *hash-and-decrypt*. Ini memungkinkan keamanan dalam model acak oracle. FDH juga melakukan hash pada pesan menggunakan fungsi yang mana ukuran image memiliki ukuran yang sama dengan ukuran modulus RSA, lalu meningkatkannya ke eksponen rahasia RSA. Penambahan dan hash tersebut dikombinasikan dalam satu langkah. Ini secara kasar menganalogikan tanda tangan digital RSA-OAEP.

- DSA

  Digital Signature Algorithm adalah standar pemerintah Amerika Serikat untuk tanda tangan digital. Ini diajukan oleh National Institute of Standards and Tecnology (NIST) pada Agustus 1991 utnuk digunakan dalam standar tanda tangan digital mereka.

  Pembangkitan kunci
  - Pilih sebuah 160-bit bilangan prima q.
  - Pilih sebuah L-bit bilangan prima p.
  - Pilih h, Dimana $1 < h < p - 1$ dimana $g = h^z \bmod p > 1$. (ingat bahwa $z = (p\text{-}1) / q$.)
  - Pilih x secara acak , dimana $0 < x < q$.
  - Hitung $y = g^x \bmod p$.
  - Kunci public adalah (p, q, g, y). Sedangkan kunci prifat adalah x.

  Catatlah bahwa (p, q, g) dapat disebarkan diantara pengguna dari system bila dibutuhkan.
  FIPS 186-3 yang akan datang menggunakan SHA-224/256/384/512 sebagai fungsi hash , q dengan ukuran 224, 256, 384, dan 512 bit, dan L setaa dengan 2048, 3072, 7680, dan 15360.

  Ada algoritma yang efesien untuk melakukan komputasi terhadap eksponensial modeluar $h^z \bmod p$ dan $g^x \bmod p$. Sebagian besar angka dalam h sesuai dengan kebutuhan, sehingga nilai 2 sering digunakan.

  Pemberian tanda tangan
  - Bangan secara acak per pesan nilai k dimana $0 < k < q$
  - Hitung $r = (g^k \bmod p) \bmod q$
  - Hitung $s = (k^{-1}(\text{SHA-1}(m) + x*r)) \bmod q$, dimana SHA-1($m$) adalah fungsi hash SHA-1 diberikan pada pesan m
  - Hitung ulang tanda tangan digitalnya untuk kasus tertentu seperti r=0 atau s=0
  - Tanda tangan digitalnya adalah (r,s)

  Algoritma Euclidian tambahan dapat digunakan untuk melakukan komputasi mod invers dari $k^{-1} \bmod q$.

  Memeriksa keabsahan tanda tangan
  - Tolak tanda tangannya apabila $0< r <q$ atau $0< s <q$ tidak terpenuhi.
  - Hitung $w = (s)^{-1} \bmod q$
  - Hitung $u1 = (\text{SHA-1}(m)*w) \bmod q$
  - Hitung $u2 = (r*w) \bmod q$
  - Hitung $v = ((g^{u1}*y^{u2}) \bmod p) \bmod q$
  - Tanda tangan tersebut valid jika $v = r$

  DSA mirip dengan skema tanda tangan digital El Gamal.

  Keabsahan dari algoritma
  Skema tanda tangan adalah benar apabila sang pemeriksa akan selalu menerima keaslian tanda tangan. Ini dapat dilihat sebagai berikut:
  Dengan $g = h^z \bmod p$ diikuti $g^q \equiv h^{qz} \equiv h^{p-1} \equiv 1 \pmod p$ dengan teorema Fermat. Karena $g>1$ dan q adalah bilangan prima maka g memiliki faktor q.

  Komputasi tanda tangan

$$s = k^{-1}(\text{SHA-1}(m) + xr) \mod q.$$

  Dimana

$$k \equiv \text{SHA-1}(m)s^{-1} + xrs^{-1} \equiv \text{SHA-1}(m)w + xrw \pmod{q}.$$

Karena *g* memiliki faktor *q* maka

$$g^k \equiv g^{\text{SHA}-1(m)w} g^{xrw} \equiv g^{\text{SHA}-1(m)w} y^{rw} \equiv g^{u1} y^{u2} \pmod{p}.$$

Akhirnya, keabsahab dari DSA dilihat dari

$$r = (g^k \mod p) \mod q = (g^{u1} y^{u2} \mod p) \mod q = v.$$

- ECDSA

  Elliptic Curve DSA (ECDSA) adalah variasi dari Digital Signature Algorithm (DSA) yang beroperasi dalam grup kurva elips. Variasi EC memungkinkan ukuran kunci yang lebih kecil untuk level keamanan yang sama. Di lain pihak, waktu eksekusi secara kasar sama dan ukuran tanda tangan digitalnya benar-benar sama: 4*t*, dimana *t* adalah parameter keamanan. Sebagai contoh, DSA dengan 1024-bit *p* dan 160-bit *q* dan ECDSA dengan 160-bit bilangan prima sama-sama menghasilkan 320-bits tanda tangan hanya memerlukan waktu beberapa milidetik untuk eksekusi dalam sebuah 2 GHz Pentium.

- ElGamal signature scheme

  ElGamal signature scheme adalah sebuah skema tanda tangan digital yang berbasiskan pada kerumitan dari komputasi logaritma diskrit. Skema ini dideskripsikan oleh Taher ElGamal tahun 1984.

  Skema tanda tangan ElGamal memungkinkan seorang pemeriksa dapat menkonformasi keabsahan dari pesan *m* yang dikirim oleh pemberi tanda tangan melalui jalur yang tidak aman.

  Parameter-parameter sistem
  - *H* adalah *collision-resistant hash function*.
  - *p* adalah bilangan prima besar yang mana untuk melakukan komputasi logaritmik diskrit modulus *p* adalah sulit.
  - *g* adalah bilangan acak yang dibangun oleh *multiplicative group* Z_p^*.

  Parameter-parameter system ini dapat dibagikan diantara sesame pengguna.

  Pembangkitan kunci
  - pilih secara acak kunci rahasia *x* dengan $1 < x < p - 1$.
  - Hitung $y = g^x \pmod{p}$.
  - Kunci public adalah (p,g,y).
  - Kunci privat adalah *x*.

  Langkah ini dilakukan sekali oleh pemberi tanda tangan.

  Pembangkitan tanda tangan
  - Pilih bilangan acak *k* dimana $0 < k < p - 1$ dan $gcd(k,p - 1) = 1$.
  - Hitung $r \equiv g^k \pmod{p}$.
  - Hitung $s \equiv (H(m) - xr)k^{-1} \pmod{p-1}$
  - Bila $s = 0$ mulai lagi dari awal.

  Maka pasangan *(r,s)* adalah tanda tangan digital dari *m*. Pemberi tanda tangan mengulangi langkah ini setiap memberikan tanda tangan.

  Verifikasi
  Sebuah tanda tangan *(r,s)* dari pesan *m* diverifikasi dengan:
  - $0 < r < p$ dan $0 < s < p - 1$.
  - $g^{H(m)} \equiv y^r r^s \pmod{p}$.

  Pemeriksa menerima keabsahan tanda tangan apabila semua kondisi terpenuhi dan menolaknya apabila tidak.

  Keabsahan algoritma
  Algoritma ini sah apabila tanda tangan yan gdibangun selalu diterima oleh pemeriksa keabsahan tanda tangan.

  Pembangunan tanda tangan menggunakan

  $$H(m) \equiv xr + sk \pmod{p-1}.$$

Sedangkan teorema Fermat menggunakan:

$$\begin{aligned} g^{H(m)} &\equiv g^{xr} g^{ks} \\ &\equiv (g^x)^r (g^k)^s \\ &\equiv (y)^r (r)^s \pmod{p}. \end{aligned}$$

Keamanan
Pihak ketiga dapat membuat tanda tangan digital baik dengan menemukan kunci rahasia pemberi tanda tangan atau dengan menemukan kekurangan dari fungsi Hash
$H(m) \equiv H(M) \pmod{p-1}$.
Kedua permasalahan tersebut sulit untuk diselesaikan.

Pemberi tanda harus dapat berhati-hati dalam memilih *k* yang berbeda-beda dalam bilangan acak untuk setiap tanda tangan dan yakin bahwa *k* tersebut informasinya benar-benar tidak tersebar. Jika tidak, seorang penyerang mungkin akan bisa mendapatkan kunci prifat *x* dari bilangan tersebut. Lebih khususnya apabila dua pesan dikirim dengan nilai *k* yang sama dan kunci yang sama, maka penyerang dapat langsung menghitung *x*.

- Undeniable signature

Undeniable signatures adalah bentuk dari digital signature ditemukan oleh David Chaum dan Hans van Antwerpen tahun 1989. Skema ini memiliki 2 fitur yang berbeda,

1. proses verifikasi bersifat interaktif, sehingga pemberi tanda tangan dapat membatasi siapa saja yang dapat memeriksa keabsahan dari tanda tangannya.
2. Protocol untuk menolak, ini adalah protocol kriptografi yang mana memperbolehkan untuk pembuktian tanda tangan adalah buatan.

Yang pertama memiliki arti bahwa pemberi tanda tangan dapat memperbolehkan hanya pada orang yang diberi otorisasi untuk mengakses dokumen tersebut untuk melakukan verifikasi terhadap tanda tangan di dokumen tersebut. Bila dokumen tersebut bocor ke pihak ketiga, pihak ketiga tersebut tidak dapat melakukan verifikasi bahwa tanda tangan yang ada pada dokumen tersebut adalah asli.

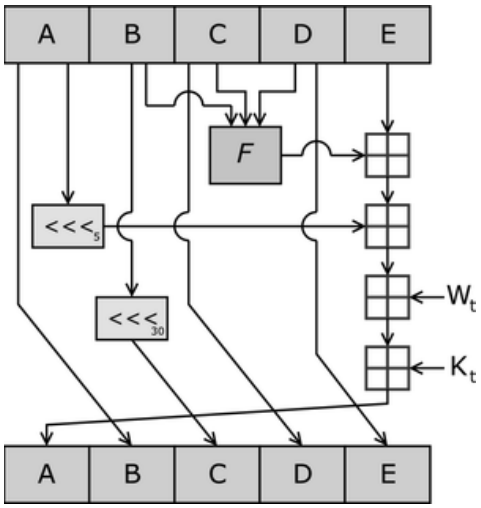
Bagaimanapun juga, karena adanya property ini, ini berarti bahwa pemberi tanda tangan mungkin dapat menolak keabsahan tanda tangan walaupun tanda tangan tersebut sebenarnya adalah valid. Untuk mencegah hal ini, kita memiliki prperti kedua, sebuah metode untuk membuktikan bahwa tanda tangan yang dibuat adalah buatan.

- SHA dengan RSA

Fungsi Hash **SHA** (Secure Hash Algorithm), dinamai SHA-1, SHA-224, SHA-256, SHA-384, and SHA-512, lima fungsi hash kriptografi yang saling berelasi yang didesain oleh NSA dan diumumkan sebagai standar pemerintahan Amerika Serikat. SHA-1 digunakan dalam banyak variasi dari berbagai aplikasi keamanan yang popular dan protocol termasuk TLS, SSL, PGP, SSH, S/MIME, dan IPSec. Ini dipertimbangkan sebagai penerus dari MD5, fungsi Hash yang sebelumnya banyak dipakai.

Empat variasi lainnya (SHA-224, SHA-256, SHA-384, dan SHA-512) adalah kadang disebut sebagai fungsi *SHA-2* atau *SHA-2*. Belum ada serangan yang dilaporkan pada variasi SHA-2, Namun karena variasi-variasi tersebut mirip dengan SHA-1, peneliti menjadi khawatir, dan membuat kandidat-kandidat yang baru dari standar hash yang lebih baik.

SHA-0 and SHA-1



Satu iterasi dalam fungsi kompresi SHA-. A, B, C, D dan E adalah kata dari status 32-bit; *F* adalah fungsi non linear yang bervariasi; $\lll_n$ mendenotasikan rotasi kiri bit sebanyak n; *n* bervariasi untuk setiap operasi. ⊞ mendenotasikan penambahan modulus 2^{32}. K_t adalah konstanta.

Spesifikasi asli dari algoritma tersebut diumumkan tahun 1993 sebagai standar keamanan hash, , FIPS PUB 180, oleh pemerintah agen standarisasi Amerika Serikat NIST (National Institute of Standards and Technology). Versi ini sekarang sering diacukan sebagai "SHA-0". Ini di ralat oleh NSA sesaat setelah pengumuman dan direvisi dengan yang baru, diumumkan tahun 1995 dalam FIPS PUB 180-1 dan sering diacu sebagai "SHA-1". SHA-1 berbeda dari SHA-0 hanya dari sebuah rotasi bitwise pada pesan dalam fungsi kompresi. Ini dilakukan untuk mengoreksi kekurangan pada algoritma sebelumnya yang mengurangi keamanan kriptografi. Bagaimanapun, NSA tidak memberikan penjelasan lebih lanjut kekurangan apa yang diperbaiki. Kelemahan sudah sering dilaporkan pada SHA-0 dan SHA-1. SHA-1 muncul untuk memberikan keamanan yang lebih baik dalam melawan serangan,

SHA-1 (seperti juga dengan SHA-0) menghasilkan degst 160-bit dari sebuah pesan dengan penjang maksimum 2^{64} - 1 bits, dan berdasarkan pada prinsip yang sama dengan yang digunakan oleh Professor Ronald L. Rivest dari MIT dalam pembuatan desain dari MD4 and MD5 algoritma message digest.

Cryptanalysis pada SHA-0

Pada CRYPTO 98, dua peneliti dari Perancis mendemokan serangan pada SHA-0 (Chabaud dan Joux, 1998): kerusakan dapat ditemukan dengan kompleksitas 2^{61}, lebih sedikit dari 2^{80} untuk fungsi hash yang ideal dengan ukuran yang sama.

Tahun 2004, Biham dan Chen menemukan *near-collisions* untuk SHA-0 — dua pesan yang memiliki nilai hash yang hamper sama, pada kasus ini, 142 dari 160 bit adalah sama. Mereka juga menemukan *collisions* penuh dari SHA-0 dikreduksi ke 62 dari awalnya 80 putaran.

Pada waktu yang berdekatan, pada 12 Augustus 2004, sebuah kekurangan pada SHA-0 diumumkan oleh Joux, Carribault, Lemuet, dan Jalby. Ini dikerjakan dengan membuat generalisasi dari serangan Chabaud dan Joux. Menemukan bahwa collision memiliki kompleksitas 2^{51} dan memakan waktu 80,000 CPU hours oleh sebuah superkomputer dengan 256 Itanium 2 processors.

Pada 17 Agustus 2004, pada sesi Rump dari CRYPTO 2004, hasil awal diumumkan oleh Wang, Feng, Lai, danYu, mengenai serangan pada MD5, SHA-0 dan fungsi Hash lainnya. Kompleksitas dari serangan mereka adalah 2^{40}, jauh lebih baik daripada serangan oleh Joux dkk.

Pada bulan Februari 2005, sebuah serangan oleh Xiaoyun Wang, Yiqun Lisa Yin, dan Hongbo Yu diumumkan yang mana dapat menemukan *collisions* pada SHA-0 dalam 2^{39} operasi.

Cryptanalysis of SHA-1

Dari hasil yang didapat dengan SHA-0, beberapa ahli menyarankan bahwa rencana untuk menggunakan SHA-1 dalam sistem kriptografi yang baru

sebaiknya dipikirkan ulang. Setelah hasil dari CRYPTO 2004 diumumkan, NIST mengumumkan bahwa mereka berencana untuk mengganti penggunaan SHA-1 pada 2010 dengan variasi dari SHA-2.

Pada awal 2005, Rijmen and Oswald mengumumkan serangan pada versi reduksi dari SHA-1 — 53 dari 80 putaran — dimana menemukan collisions dengan kompleksitas kurang dari 2^{80} operasi.

Pada Februari 2005, sebuah serangan dari Xiaoyun Wang, Yiqun Lisa Yin, dan Hongbo Yu diumumkan. Serangan tersebut dapat menemukan kolisi dari versi penuh SHA-1 membutuhkan kurang dari 2^{69} operasi (pencarian brute force akan membutuhkan 2^{80} operasi).

Penulisnya menuliskan: "khususnya, analisis kamu dibangun berdasarkan serangan-serangan yang aslinya untuk SHA-0, serangan near collision pada SHA0, tehnik multiblock collision, begitu juga dengan tehnik perubahan pesan yang digunakan pada pencarian kolisi pada MD5. menghancurkan SHA-1 tidak akan mungkin tanpa tehnik-tehnik tingkat tinggi ini "[3]. Penulisnya mempresentasikan sebuah kolisi utnuk 58 putaran SHA-1, ditemukan dengan 2^{33} operasi hash. Paper dengan deskripsi serangan langkapnya diumumkan pada Agustus 2005 pada konfrensi CRYPTO.

Pada sebuah wawancara, Yin berkata bahwa, "secara kasar, kami mengeksploitasi dua kelemahan berikut: satu adalah langkah preprosesing file tidak cukup kompleks. Lainnya adalah operasi matematik tertentu pada 20 putaran pertama memiliki masalah keamanan yang tidak diinginkan" [3].

Pada 17 Agustus 2005, sebuah perbaikan pada serangan untuk SHA-1 diumumkan oleh Xiaoyun Wang, Andrew Yao dan Frances Yao pada sesi Rump CRYPTO 2005, menurunkan kompleksitas diperlukan untuk menemukan kolisi pada SHA-1 dengan 2^{63}.

Dalam kriptografi akademis, setiap serangan memiliki lebih sedikit kompleksitas daripada pencarian brute forece dipertimbangkan sebagai sebuah gebrakan. Ini bukan berarti, bagaimanapun, perlu untuk mengeksploitasikan serangan tersebut.

Dalam hal keamanan praktis, perhatian utama dari serangan baru in adalah mungkin saja ini memudahkan dalam menemukan cara yang lebih efesien. Walaupun ini adalah kasusu sudah dilihat atau tidak, tetapi perpindahan ke hash yang lebih kuat diyakini diperlukan. Sebuah serangan kolisi tidak merepresentasikan jenis yang sama dengan bahata serangan yang seharusnya. Banyak aplikasi yang menggunakan kriptografi hash, seperti tempat penyimpanan password atau penanda tanganan dokumen, adalah hanya minimal disebabkan oleh serangan kolisi. Dalam kasus tanda tangan digital, penyerangnya tidak dapat dengan mudah memalsukan tanda tangan dari dokumen yang ada.

Pada sesi Rump dari CRYPTO 2006, Christian Rechberger dan Christophe De Cannière menyebutkan bahwa telah menemukan serangan kolisi pada SHA-1 yang memungkinkan penyerangnya menemukan seidaknya bagian dari pesan.

Variasi lebih lanjut

NIST telah mempublikasi empat fungsi hash tambahan ke dalam keluarga SHA, masing-masing dengan digest yang lebih panjang, dinamai dengan SHA-2. variasi masing-masing dinamai sesuai dengan panjang digest mereka (dalam bit): "SHA-256", "SHA-384", dan "SHA-512". Mereka pertama kali dipublikasikan pada tahun 2001 pada draft FIPS PUB 180-2, yang mana review waktu dan komentar dapat diterima. FIPS PUB 180-2, juga mengikutsertakan SHA-1, juga dikeluarkan sebagai standar official pada 2002. Pada Februari 2004, sebuah perubahan ditemukan dan dipublikasikan untuk FIPS PUB 180-2, menspesifikasikan variasi-variasi tambahan, "SHA-224", didefinisikan

untuk menemukan kunci dengan panjang dua buah kunci Triple DES. Variasi ini dipatenkan di Amerika Serikat.

SHA-256 dan SHA-512 adalah fungsi hash yang sangat bagus yang dikomputasi dengan 32- dan 64-bit kata-kata. SHA-224 dan SHA-384 adalah versi baru dari dua sebelumnya, dikomputasi dengan nilai inisiasi yang berbeda.

Fungsi hash yang baru ini belum mendapatkan kepercayaan dari masyarakat komunitas kriptografi saperti yang didapatkan SHA-1, dan juga keamanan kriptografis mereka juga belum dimapankan. Gilbert danHandschuh (2003) telah mempelajari variasi yang lebih baru dan menemukan tidak ada kelemahan.

- **Algoritma tanda tangan Rabin**

  Dalam kriptografi, **Rabin signature algorithm** adalah protocol tanda tangan digital yang ditemukan oleh Michael Rabin. Ini dekat dengan algoritma ankripsinya dan keamanannya bergantung pada *intractability* dari *integer factorization*.

  Pesan pada system ini harus sebuah quadratic residue modulo *n*, dan harus mengandung redudansi untuk mencegah *existential forgeries*. Dikarenakan akar unik yang berbeda-beda ini, tanda tangannya menjadi lebih sulit untuk diperiksa bila dibandingkan dengan tanda tangan dengan skema lainnya.

- **Algoritma tanda tangan Pointcheval-Stern**

  Dalam kriptgrafi, **Pointcheval-Stern signature algortihm** adalah sebuah skema tanda tangan digital yang berbasiskan pada skema ElGamal. Algorritma ini sedikit merubah skema ElGamal untuk menghasilkan sebuah algoritma yang dapat menawarkan keamanan yang kuat melawan *adaptive chosen-plaintext attacks*.

  David Pointcheval dan Jacques Stern membangun tehnik *forking lemma* dalam membuat bukti mereka terhadap algoritma ini. Ini sudah digunakan oleh investigasi keamanan lainnya dari beberapa variasi algoritma kriptografi.

- **Schnorr signature**

  Dalam kriptografi, **Schnorr signature** adalah sebuah tanda tangan digital yang dilakukan dengan algoritma tanda tangan Schnorr. Keamanannya berdasarkan pada kemampuan untuk berinteraksi dari beberapa permasalahan logaritma yang berlainan. Ini dipertimbangkan sebagai skema tanda tangan digital yang paling simple yang terbukti aman dalam model random oracle.ini efesien dan membangun tanda tangan yang pendek.

  Memilih parameter
  - Semua pengguna dalam skema tanda tangan setuju dengan G dengan pembangun g dari factor prima q yang mana permasalah logaritmik diskrit adalah sulit.
  - Semua pengguna menyetujui pada fungsi hash H.

  Pembangkitan kunci
  - Pilih sebuah kunci privat x yang mana $0 < x < q$.
  - Kunci public adalah y dimana $y = g^{-x}$.

  Pemberian tanda tangan
  Untuk memberi tanda tangan pada pesan M:
  - Pilih bilangan acak k yang mana $0 < k < q$
  - Hitung $r = g^k$
  - Hitung $e = H(M \,||\, r)$
  - Hitung
    $$s = (k + xe) \mod q$$

  Tanda tangan digitalnya adalah pasangan (e,s). catatlah bahwa $0 \leq e < q$ dan $0 \leq s < q$; bila Schnorr group digunakan dan $q < 2^{160}$, ini berarti bahwa tanda tangan dapat digunakan sampai 40 bytes.

Verifikasi

- Hitung $r_v = g^s y^e$
- Hitung $e_v = H(M \,||\, r_v)$

Bila $e_v = e$ maka tanda tangan tersebut tepat.

2.7 Tanda tangan digital dalam kehidupan sehari-hari

Semua skema tanda tangan digital memiliki beberapa kebutuhan awal yang tanpanya tanda tangan tersebut tidak akan berarti, baik itu teori kriptografi maupun masalah legalitas dari tanda tangan tersebut.

a. Algoritma yang berkualitas. Beberapa algoritma kunci public sudah diketahui ketidakamanannya, serangan terhadap algoritma-algoritma tersebut telah diketahui.
b. Implementasi berkualitas. Implementasi dari sebuah algoritma yang baik (atau protokol) yang memiliki kesalahan tidak akan berhasil.
c. Kunci prifat harus dijaga kerahasiaannya. Bila ada pihak lain yang mengetetahui kunci tersebut, pihak tersebut dapat membuat tanda tangan digital yang sempurmna di dokumen apapun.
d. Distribusi dari kunci public harus diselesaikan dengan baik sehingga kunci public tersebut dapat diterima dengan baik oleh orang yang seharusnya menerimanya. Biasanya masalah ini diselesaikan dengan menggunakan infrastruktur kunci public dan asosiasi yang digunakan pengguna kunci public.
e. Pengguna (dan perangkat lunaknya) harus menjalankan protocol tanda tangan digital dengan benar.

Hanya jika semua kondisi diatas terpenuhi maka tanda tangan digital benar-benar dapat menjadi bukti siapa yang mengirimkan pesan, dan siapa yang bertanggung jawab akan isi didalamnya. Hukum yang telah ada sejak dahulu tidak dapat merubah realitas ini bahwa ada kemungkinan rekayasa seperti ini, walau seberapa tidak mencerminkan benar-benar seperti ini.

Legalisasi, menjadi importuned oleh bisnis mengharapkan keuntungan dengan menalankan PKI, atau dengan memecahkan masalah lama dengan cara baru dengan bantuan teknologi, telah memiliki status praktek dan/atau regulasi di benyak otorisasi juridikasi, endorsing, memberanikan, atau mengizinkan tanda tangan digital dan menawarkannya untuk aspek legal mereka. Adopsi dari standar teknik untu ktanda tangan digital telah ada dibelakang banyak legislasi, menunda banyak atau sedikit posisi perekayasa yang bersatu dalam interoperability, pemilihan algoritma, panjang kunci, dan seterusnya adalah apa yang diinginkan oleh ilmu rekayasa untuk dihasilkan.

3. *Blind Digital Signature* dan *Group Digital Signature*

3.1 *Blind Digital Signature*

Dalam kriptografi, sebuah **blind signature**, seperti yang telah diperkenalkan oleh David Chaum, adalah sebuah bentuk tanda tangan digital yang mana isi dari pesan disamarkan sebelum pesan tersebut diberikan tanda tangan. Hasil dari blind signature dapat secara publik dibuktikan keasliannya bila dibandingkan algoritma aslinya, pesan yang tidak disamarkan seperti pada tanda tangan digital biasa. Blind signatures biasanya digunakan dalam protokol yang berhubungan dengan prifasi dimana pemberi tanda tangan dan pembuat pesan adalah pihak yang berbeda. Contohnya adalah pemilihan kriptografis dan skema digital cash.

Sebuah analogi yang sering digunakan untuk kriptografi blind signature adalah kegiatan menutup sebuah pesan dalam sebuah amplop yang disegel dan diberi tanda tangan oleh pemberi tanda. Untuk itu, pemberi tanda tangan tidak melihat isi pesan, akan tetapi pihak ketiga dapat melakukan verifikasi terhadap tanda tangan tersebut apakah valid atau tidak selama dalam batas skema tanda tangan digital.

Blind signatures dapat juga digunakan untuk memberikan *unlinkability*, yang mana mencegah pemberi tanda tangan untuk menghubungkan pesan yang disamarkan dengan surat yang tidak disamarkan yang mungkin akan digunakan untuk verifikasi. Dalam kasus ini, pemberi response yang dilakukan tanda tangan pertama-tama adalah "tidak menyamarkan" prioritas untuk verifikasi dalam cara yang mana tanda tangan tetap validuntuk pesan yang disamarkan. Ini bisa

berguna dalam skema saat dibutuhkan *anonymity*.

Skema blind signature dapat diimpleentasikan dengan menggunakan beberapa skema kunci public, seperti RSA dan DSA. Untuk membuat tanda tangan seperti ini, pesan tersebut pertama disamarkan, dengan cara menggabungkannya dengan suatu cara dengan "factor penyamar". Pesan yang disamarkan diberikan pada pemberi tanda tangan, yang akan memberi tanda tangan dengan algoritma biasa yang telah disebutkan pada bab sebelumnya. Pesan hasilnya, beserta factor penyamar, dapat diverifikasi dengan kunci publik pemberi tanda tangan. Dalam beberapa skema blind signature, seperti RSA, bahkan dimungkinkan untuk memindahkan factor penyamar dari tanda tangan sebelum diverifikasi. Dalam skema ini, keluaran akhir (pesan/tanda tangan) dari skema blind signature adalah identik dengan yang dihasilkan protocol tanda tangan digital biasa.

3.1.1 Kegunaan

Skema blind signature schemes melihat aspek penting dari kegunaan aplikasi dimana prifasi pengirim adalah penting. Ini menyertakan beberapa skema "digital cash" dan protokol ambil suara.

Sebagai contoh, kesatuan dari beberapa system pengambilan suara elektronikthe mungkin membutuhkan tiap bilik disertifikasi oleh sebuah organisasi pemilihan sebelum dapat diterima untuk dihitung. Ini memungkinkan organisasi tersebut untuk memeriksa kelengkapan yang diperlukan untuk memberikan suara, dan memeriksa bahwa tiap pemilih hanya mengeluarkan suaranya satu kali saja. Secaraqsimultan, adalah penting bagi organisasi ini tidak melihat pilihan suara sang pemilih. Sebuah blind signature yang tidak terhubung dapat menjamin hal ini, dimana pihak yang memeriksa tidak akan melihat isi pilihan dari tiap bilik, dan tidak dapat menghubungkan bilik yang disamarkan dengan bilik yang tidak disamarkan yang menerima untuk penghitungan.

3.1.2 Skema Blind Signature

Skema blind signature dapat digunakan oleh banyak protocol tanda tangan digital kunci public. Satu contoh diberikan berikut. Dalam tiap contoh, pesan yang akan diberikan tanda tangan mengandung nilai *m. m* dapat dipertimbangkan sebagai beberapa masukan yang legitimasi untuk fungsi tanda tangan digitalnya.

3.1.2.1 Blind RSA Signatures

Salah satu skema yang paling simple adalah yang berbasiskan pada tanda tangan digital RSA. Tanda tangan digital RSA yang tradisional dikomputasi dengan melakukan eksponen terhadap pesan *m* dengan eksponen rahasisa *d*, semua mod sebuah modulus publik *N*. Versi penyamaranmenambahkan sebuah nilai acak *r*, dimana *gcd*(r, *N*) = 1. *r* tersebut dihitung dengan eksponen publik *e* (mod *N*), dan nilai dari r^e digunakan sebagai faktor penyamar. Pihak pemberi tanda tangan menerima tanda tangan dan faktor penyamar $m(r^e) \bmod N$, yang mana menyamarkan pesan tersebut. Blinded signature *s'* kemudian dihitung dengan cara:

$$s' \equiv (m(r^e))^d \pmod N$$

Pemilik pesan dapat menyingkirkan factor penyamar untuk mendapatkan *s*, tanda tangan RSA yang valid dari *m*:

$$s \equiv s' * r^{-1} \pmod N$$

Ini dapat bekerja karena kunci RSA memenuhi persamaan

$$r^{ed} \equiv r \pmod N$$

Dan dimana

$$s \equiv s' * r^{-1} \equiv m^d r^{ed} r^{-1} \equiv m^d r r^{-1} \equiv m^d \pmod N,$$

dimana *s* adalah tentunya tanda tangan dari *m*.

3.2 *Group Digital Signature*

Sebuah skema **Group signature** adalah sebuah metode yang memungkinkan sebuah grup untuk secara anonym memberi tanda tangan pada sebuah pesan atas nama grup. Konsep ini diperkenalkan oleh David Chaum dan Eugene van Heyst pada 1991. Sebagai contoh, sebuah skema group signature dapat digunakan oleh seorang karyawan dari sebuah perusahaan yang besar dimana diperukan untuk mengetahui bahwa pesan yang diberikan diberikan tanda tangan oleh seorang karyawan dari perusahaan tersebut, tapi bukan untuk mengetahui dengan tepat siapa yang memberikan tanda tangan tersebut. Pengaplikasian lainnya adalah untuk kartu akses untuk membatasi area yang tidak sepantasnya untuk melihat setiap gerak-gerik dari karyawan, namun diperlukan untuk

mengamankan area hanya untuk karyawan dalam grup.

Esensial untuk skema group signature scheme adalah sebuah *group manager*, yang mana bertanggung jawab untuk menambahkan anggota grup dan memliki kemampuan untuk memperlihatkan pemberi tanda tangan yang asliny pada public. Dalam beberapa sistem tanggung jawab untuk menambahkan orang dan memperlihatkan pemberi tanda tangan aslinya dipisahkan pada dua orang yang berbeda yaitu *membership manager* dan *revocation manager*. Banyak skema yang telah diajukan, namun semuanya harus melihat kebutuhan dasarnya dahulu seperti berikut:

- **Soundness and Completeness**: tanda tangan yang valid akan selalu benar saat diverifikasi, dan tanda tangan yang tidak valid akan selalu tidak dapat melewati proses verifikasi.
- **Unforgeable**: Hanya anggota grup yang dapat membuat tanda tangan kelomok yang valid.
- **Signer ambiguous**: Apabila diberikan pesan beserta tanda tangannya, identitas dari individu pemberi tanda tangan tidak dapat ditentukan tanpa kunci rahasia *revocation manager*.
- **Unlinkability**: Apabila diberikan dua buah pesan dan tanda tangannya, kita tidak dapat mengetahui bahwa kedua pesan tersebut diberikan tanda tangan oleh orang yang sama atau tidak.
- **No Framing**: Walaupun bila semua angota kelompok (beserta manager) bersatu, mereka tidak dapat membuat tanda tangan untuk pihak yang tidak berpartisipasi dalam kelompok.
- **Unforgeable tracing verification**: *Revocation manager* tidak dapat salah menyatakan bahwa individu tersebutlah yang memberi tanda tangan padahal sebenarnya tidak.

4. *Group Blind Digital Signature*

4.1 Penjelasan

Group blind digital signature mengkombinasikan properti dari grup signature dan blind signature. Pembuatan pertama akan group blind digital signature adalah oleh Lysyanskaya dan Ramzan. Beberapa aplikasi dari group blind digital signature termasuk desain dari sistem pembayaran elektronik yang menyangkut beberapa bank, dan pemilihan online yang menyangkut beberapa server pemilihan.

Kebutuhan keamanan dari Group Blind Digital Signature sangat mirip dengan yang dimiliki Group Digital Signature. Penambahan yang ada hanyalah kita membutuhkan properti penyamaran dalam tanda tangan. Berikut kebutuhan-kebutuhannya:

- **Blindness of Signature:** pemberi tanda tangan harus tidak dapat melihat isi dari pesan yang ia beri tanda tangan. Lebih lanjut, pemberi tanda tangan harus tidak tahu ia telah memberi tanda tangan pada dokumen mana, walaupun ia (atau orang lainnya) dapat memverifikasi bahwa tanda tangan tersebut valid.
- **Soundness and Completeness**: tanda tangan yang valid akan selalu benar saat diverifikasi, dan tanda tangan yang tidak valid akan selalu tidak dapat melewati proses verifikasi.
- **Unforgeable**: Hanya anggota grup yang dapat membuat tanda tangan kelomok yang valid.
- **Signer ambiguous**: Apabila diberikan pesan beserta tanda tangannya, identitas dari individu pemberi tanda tangan tidak dapat ditentukan tanpa kunci rahasia *revocation manager*.
- **Unlinkability**: Apabila diberikan dua buah pesan dan tanda tangannya, kita tidak dapat mengetahui bahwa kedua pesan tersebut diberikan tanda tangan oleh orang yang sama atau tidak.
- **No Framing**: Walaupun bila semua angota kelompok (beserta manager) bersatu, mereka tidak dapat membuat tanda tangan untuk pihak yang tidak berpartisipasi dalam kelompok.
- **Unforgeable tracing verification**: *Revocation manager* tidak dapat salah menyatakan bahwa individu tersebutlah yang memberi tanda tangan padahal sebenarnya tidak.
- **Undeniable Signer Identity**: Manager dari kelompok selalu dapat memberi tahu siapa yang memberika tanda tangan yang valid/ lebih lanjut, ia juga dapat membuktikan pada beberapa entitas (seperti juri) siapa anggota yang memberikan tanda tangan tanpa

berkompromi dengan anonim dari pemberian tanda tangan.

- **Coalition Resistance**: Semua anggota kelompok (tidak termasuk manager kelompok) harus tidak dapat membuat group signature yang valid yang tidak dapat dilacak. Jelasnya, kita ingin mencegah serangan yang mana koalisi dari anggota kelompok bersama-sama, memberikan informasi mereka, dan membuat tanda tangan yang lolos verifikasi.

4.2 Skema Group Blind Digital Signature oleh Zulfikar Amin Ramzan

Skema yang dibuat oleh Zulfikar Amin Ramzan menggunakan skema group digital signature yang telah dibuat oleh Camenish dan Steadler dan memodifikasinya sehingga skema tersebut juga memiliki properti penyamaran juga[1].

4.2.1 Pemberian tanda tangan dan verifikasi

Protokol pemberian tanda tangan:

1. Hitung $q \in_R \mathcal{Z}_n^*$ dan sekumpulan
   $\tilde{g} := g^q$ dan
   $\tilde{z} := \tilde{g}^y$.
2. Hitung random
   $2^\lambda \leq u_i \leq 2^{\lambda+\mu} - 1, \text{ for } 1 \leq i \leq l$
   dan
   $P_i^{SKLOGLOG} := \tilde{g}^{(a^{u_i})}, \text{ for } 1 \leq i \leq l.$
3. Hitung random
   $v_i \in \mathcal{Z}_n^*,$
   Untuk
   $1 \leq i \leq l$
   Dan hitung
   $P_i^{SKROOTLOG} := \tilde{g}^{(v_i^e)}, \text{ for } 1 \leq i \leq l.$
4. Kirim
   $(\tilde{g}, \tilde{z}, \{P_i^{SKLOGLOG}\}, \{P_i^{SKROOTLOG}\})$
   kepada pengguna.

Selanjutnya pengguna melakukan langkah berikut:

1. Ada bilangan
   $b \in_R \{1 \ldots 2^\lambda - 1\}$
   dan
   $f \in_R \mathcal{Z}_n^*,$
   Lalu hitung
   $w := (af)^{eb} \pmod n.$
2. Hitung
   $\hat{g} := \tilde{g}^w,$
   $\hat{z} := \tilde{z}^w,$
   $\hat{P}_i^{SKLOGLOG} := (P_i^{SKLOGLOG})^w,$
   dan
   $\hat{P}_i^{SKROOTLOG} := (P_i^{SKROOTLOG})^w$
3. Eksekusi langkah 2, 3, dan 4 dari protokol blind *SKLOGLOG*, dan ambil $\{\hat{P}_i^{SKLOGLOG}\}$ sebagai nilai komitmen. Isi respon $\{t_i^{SKLOGLOG}\}$ dengan menambahkan *eb*.
4. Eksekusi langkah 2, 3, dan 4 dari protokol blind *SKROOTLOG*, dan ambil $\{\hat{P}_i^{SKROOTLOG}\}$ sebagai nilai komitmen. Isi respon $\{t_i^{SKROOTLOG}\}$ dengan menghitung $(af)^b$.

Kita sekarang telah memiliki:

$$V_1 = SKLOGLOG_l[\alpha \mid \hat{z} = \hat{g}^{a^\alpha}](m)$$

dan

$$V_2 = SKROOTLOG_l[\beta \mid \hat{z}\hat{g} = \hat{g}^{\beta^e}]$$

Hasil tanda tangan dari pesan *m* berupa $(\hat{g}, \hat{z}, V_1, V_2)$ dan dapat diverifikasi dengan memeriksa nilai dari V_1 dan V_2.

4.3 Penggunaan dalam kehidupan sehari-hari

4.3.1 Pengaplikasian Group Blind Digital Signature pada Distributed Electronic Banking

Pihak-pihak yang terkait dengan pengaplikasian kita adalah Alice, Bob, BankA, dan BankB. Disini Alice adalah konsumen yang menggunakan BankA. Alice ingin membeli beberapa barang dari sebuah vendor Bob. BankB adalah Bank yang digunakan oleh Bob.

4.3.1.1 Setup

Bank dari skema kita membentuk sebuah kelompok, dan tiap kelompok memilki manager. Bila ada bank lain yang ingin masuk ke dalam kelompok, maka dapat dilakukan dengan sangat mudah. Tiap Bank melakukan protokol penggabungan dengan manager kelompok, dan mampu untuk membuat tanda tangan atas nama kelompok dari Bank.

4.3.1.2 Pengambilan uang

Untuk mengambil uang, langkah-langkah berikut dilakukan:

1. Alice membuat koin elektronik *C*. Koin ini berisi beberapa nomor seri, beberapa informasi seperti harga, dll.
2. Alice kemudian meminta BankA untuk memasukkan Group Blind Digital Signature pada *C*.
3. BankA memasukkan tanda tangan ke *C*, dan mengambil sejumlah uang dari rekening Alice. Alice sekarang memiliki koin *C* lengkap dengan group signature yang vslid pada koin *C*.

4.3.1.3 Pembelanjaan

Untuk membelanjakan uangnya, langkah berikut digunakan:

1. Alice memberikan koin *C* beserta tanda tangan dari Bank pada vendor Bob.
2. Bob memeriksa apakah tanda tangan yang ada pada koin *C* adalah asli. Ini bisa dilakukan dengan mudah dengan menggunakan *Group Public Key*.
3. Bila semua pemeriksaan telah selesai, Bob memberikan koin tersebut kepada BankB untuk di deposito, dan menunggu response dari Bank.

4.3.1.4 Deposito

Untuk mendeposito koin tersebut, langkah berikut harus dilakukan:

1. Bob mengambil koin *C* beserta tanda tangan dari BankA dan memberikannya pada BankB
2. BankB memeriksa koin tersebut, dan memverifikasi validasi koin tersebut dengan memeriksa tanda tangan pada koin tersebut. Catat bahwa BankB tidak perlu tahu siapa dari BankA yang memberikan tanda tangan tersebut. Ini bisa dilakukan dengan mudah menggunakan *group public key*.
3. BankB kemudian memeriksa apakah koin tersebut telah digunakan, bank mungkin harus menyimpan catatan mengenai koin-koin yang telah digunakan untuk menyelesaikan pemeriksaan dua arah ini.
4. Bila semua pemeriksaan selesai, maka BankB menambahkannya pada rekening Bob sesuai dengan jumlah yang diberitahu.
5. Bob dapat memberikan barang yang dibeli Alice pada Alice.

5. Kesimpulan

Kesimpulan yang diambil adalah bahwa tanda tangan digital sudah semakin banyak digunakan di dunia ini. Karena semakin banyak digunakannya tanda tangan digital ini, maka aspek-aspek keamanan dari tanda tangan digital harus semakin diperhatikan. Hukum-hukum dan undang-undang mengenai tanda tangan digital harus dbuat agar penggunaan tanda tangan digital dapat menjadi lebih efesien.

Untuk meningkatkan keamanan pada tanda tangan digital, maka perlu dibuat skema tanda tangan digital untuk kasus yang sesuai. Misalkan Grou pbLind Digital Signature pada kasus diatas, atau tanda tangan digital biasa pada percakapan antara dua individu saja.

Pada Group Blind Digital Signature, banyak keuntungan yang dapat diambil seperti yang telah disebutkan sebelumnya. Penggunaan Group Blind Digital Signature di Indonesia akan sangat berguna apabila Indonesia telah membuat undang-undang mengenai tanda tangan digital.

DAFTAR PUSTAKA

[1] Amin Ramzan, Zulfikar, *Group Blind Digital Signature: Theory and Application*, Thesis, Departement of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1999.

[2] Munir, Rinaldi. (2006). Bahan Kuliah IF5054 Kriptografi. Departemen Teknik Informatika, Institut Teknologi Bandung.

[3] http://wikipedia.org diakses pada tanggal 14 Desember 2006