Undeniable Signature

Muhamad Pramana Baharsyah (13503052)

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung

if13052@students.if.itb.ac.id

Abstrak

Digital Signature atau yang juga dikenal dengan tanda tangan digital merupakan salah satu mekanisme untuk menjaga keotentikan pesan beserta pengirimnya. Dengan menggunakan tanda tangan digital, masalah keamanan sebuah pesan atau sebuah dokumen dapat di tangani. Masalah-masalah seperti otentikasi pengirim pesan, integritas pesan serta anti-penyangkalan dapat diselesaikan dengan menandatangani sebuah dokumen dengan tanda tangan digital.

Undeniable signature ialah skema tanda tangan digital dimana untuk memverifikasi tanda tangan digital diperlukan persetujuan pemberi tanda tangan, biasanya proses verifikasi berlangsung dengan sepengetahuan pemberi tanda tangan. Skema tersebut disebut juga *non-self-athenticating signature schemes*. Jika sebuah tanda tangan digital hanya dapat diverifikasi dengan persetujuan pemberi tanda tangan, pemberi tanda tangan tersebut mungkin saja menolak ketika diminta untuk mengotentifikasi dokumen yang ia tanda tangani. *Undeniable Signature* dapat memecahkan masalah seperti ini dengan menggunakan sebuah protokol yang disebut *disavowal protocol* pada skema tanda tangan digital. Skema *undeniable signature* pertama kali diusulkan oleh Chaum dan Antwerpen dalam makalah mereka ([CV90] dan [CV91]).

Skema undeniable signature ini diimplementasikan menggunakan kriptografi kunci publik yang berbasis masalah logaritma diskrit. Mekanisme penandatanganan dokumen mirip dengan mekanisme pada skema lainnya. Perbedaan hanya terletak pada proses verifikasi. Dengan suatu mekanisme tertentu, maka pemberi tanda tangan tidak akan bisa mengingkari bahwa ia telah menandatangani sebuah dokumen.

***Kata kunci** : undeniable signature, kriptografi kunci publik, tanda tangan digital.*

Pendahuluan

Sejak zaman dahulu, tanda tangan sudah digunakan untuk otentifikasi dokumen cetak. Tanda tangan digunakan sebagai alat untuk menguji otentikasi sebuah dokumen karena tanda tangan mempunyai karakteristik sebagai berikut:

1. Tanda tangan adalah bukti yang otentik.
2. Tanda tangan tidak dapat dilupakan.
3. Tanda tangan tidak dapat dipindah untuk digunakan ulang.
4. Dokumen yang telah ditandatangani tidak dapat diubah.
5. Tanda tangan tidak dapat disangkal.

Fungsi tanda tangan pada dokumen cetak tersebut juga dapat diterapkan pada data digital (pesan, dokumen elektronik). Layaknya sebuah dokumen cetak, dokumen digital juga memerlukan suatu mekanisme untuk otentifikasi penulis ataupun pengirim dokumen tersebut. Oleh karena itu digunakanlah tanda tangan digital (*Digital Signature*) untuk menjamin otentikasi sebuah dokumen.

Digital Signature

Digital signature atau tanda tangan digital ialah sebuah tanda tangan elektronis yang dapat digunakan untuk mengotentifikasi identitas pengirim pesan atau identitas penulis dokumen. Tanda tangan digital juga biasa digunakan untuk

memastikan isi pesan asli ataupun isi suatu dokumen yang diterima tidak mengalami perubahan dari isi asli yang telah ditandatangani oleh pengirim. Tanda tangan digital harus dengan mudah dapat dikirimkan bersama pesan dan tidak boleh dapat ditirukan oleh orang lain serta dapat ditambahkan waktu penandatanganan dokumen atau pesan tersebut oleh pengirim pesan. Kemampuan untuk memastikan bahwa dokumen atau pesan yang telah diberi tanda tangan digital tidak berubah dari isi aslinya berarti pengirim pesan atau penulis dokumen tidak dapat membantah bahwa ia telah menandatangani dokumen atau pesan tersebut.

Tanda tangan digital dapat digunakan pada sembarang jenis pesan, baik pesan terenkripsi maupun pesan biasa tanpa enkripsi. Oleh sebab itu, penerima pesan dapat merasa yakin akan identitas pengirim pesan dan yakin bahwa pesan yang dikirim lengkap dan sesuai dengan yang ditulis oleh pengirim, dengan kata lain pesan tidak mengalami perubahan.

Penggunaan tanda tangan digital dapat menangani masalah keamanan pada pesan yaitu mencakup :

1. Keabsahan pengirim (*sender authentication*)

   Hal ini berhubungan dengan kebenaran identitas pengirim pesan. Dengan kata lain pertanyaan seperti apakah pesan yang diterima benar-benar berasal dari pengirim sesungguhnya dapat dijawab dengan mudah.

2. Keaslian pesan (*message authentication*)

   Hal ini berhubungan dengan integritas pesan yang diterima. Maksudnya ialah pesan yang diterima dapat diuji apakah telah mengalami perubahan dari pesan aslinya. Perubahan yang terjadi dapat berupa penambahan isi pesan, penghapusan sebagian isi pesan, ataupun perubahan sejumlah karakter dalam pesan.

3. Anti-penyangkalan (*non-repudiation*)

   Hal ini berhubungan dengan pengirim pesan yang bersangkutan. Pengirim tidak dapat menyangkal bahwa suatu pesan ia kirimkan. Hal ini merupakan konsekuensi dari poin pertama dan kedua. Jika keabsahan pengirim pesan dan isi dari pesan itu sendiri sudah dapat diverifikasi, maka pengirim pesan tidak dapat menyangkal telah menandatangani pesan tersebut.

Pada dasarnya kriptografi kunci simetri sudah cukup untuk memenuhi masalah keamanan poin pertama dan kedua pada daftar masalah keamanan yang harus diperhatikan di atas. Hal tersebut didasari dengan fakta akan kapabilitas kriptografi kunci simetri beserta karakteristik yang dimilikinya. Kunci simetri hanya diketahui oleh pengirim dan penerima pesan, sehingga penerima dapat dengan mudah mempercayai bahwa pesan memang dikirim oleh pengirim dan belum mengalami perubahan apapun, baik isinya maupun perubahan lainnya, karena tidak ada orang lain, selain mereka berdua, yang mengetahui kunci pesan tersebut.

Masalah muncul ketika persyaratan ketiga pada daftar masalah keamanan pengiriman pesan di atas juga harus dipenuhi. Algoritma kriptografi kunci simetri tidak menyediakan mekanisme anti-penyangkalan (*non-repudiation*). Algoritma kriptografi kunci simetri tidak dapat menangani masalah yang akan muncul misalnya pengirim menyangkal telah mengirim pesan tersebut, atau penerima menyangkal isi pesan yang telah diterimanya. Oleh karena itu diperlukan sebuah mekanisme lain dalam menangani masalah otentikasi pesan.

Agar dapat mengatasi masalah penyangkalan, maka diperlukan pihak ketiga yang dapat dipercaya oleh kedua pihak, pengirim pesan dan penerima pesan. Pihak ketiga ini disebut penengah (arbitrase).

Misalkan BB (*Big Brothers*) adalah otoritas arbitrase yang dipercaya oleh Alice sebagai pengirim pesan dan Bob sebagai penerima pesan. BB memberikan kunci rahasia K_A kepada Alice dan kunci rahasia K_B kepada Bob. Hanya Alice dan BB yang mengetahui K_A, begitu juga hanya Bob dan BB yang mengetahui K_B. Jika Alice menyangkal telah mengirim pesan tersebut, maka pernyataan dari BB pada pesan yang diterima oleh Bob digunakan untuk menolak penyangkalan Alice. Oleh karena itu Alice tidak mungkin bisa menyangkal telah mengirim pesan kepada Bob.

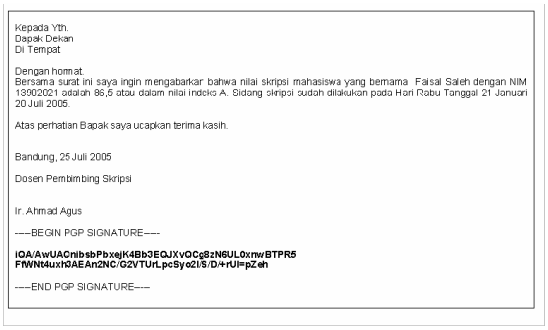
Bagaimana BB tahu bahwa pesan tersebut dari Alice dan bukan dari Charlie? Karena hanya BB dan Alice yang mengetahui kunci rahasia, maka

hanya Alice yang dapat mengenkripsi pesan dengan kunci tersebut.

Solusi lainnya ialah dengan menggunakan algoritma kriptografi kunci publik untuk menyelesaikan masalah tersebut. Algoritma kriptografi kunci publik ini dipilih untuk mengantisipasi penyangkalan yang mungkin dilakukan, baik oleh pengirim, maupun oleh penerima pesan.

Adapun cara untuk menandatangani sebuah dokumen dengan tanda tangan digital dapat dilakukan dengan menggunakan kombinasi antara fungsi *hash* dan kriptografi kunci publik.

Contoh pemberian tanda tangan digital pada pesan elektronis (email) dapat dilihat dari gambar berikut :


Kepada Yth.
Dapak Dekan
Di Tempat

Dengan hormat.
Bersama surat ini saya ingin mengabarkar bahwa nilai skripsi mahasiswa yang bernama Faisal Saleh dengan NIM 13902021 adalah 86,6 atau dalam nilai indeks A. Sidang skripsi sudah dilakukan pada Hari Rabu Tanggal 21 Januori 20 Juli 2005.

Atas perhatian Bapak saya ucapkan terima kasih.

Bandung, 25 Juli 2005

Dosen Pembimbing Skripsi

Ir. Ahmad Agus

-----BEGIN PGP SIGNATURE-----

iQA/AwUACnibsbPbxejK4Bb3EQJXvOCg8zN6UL0xnwBTPR5
FfWNt4uxh9AEAn2NC/G2VTUrLpcSyo2l/5/D/+rUl=pZeh

-----END PGP SIGNATURE-----

Pada bagian bawah pesan terdapat sebuah tanda tangan digital. Isi dari tanda tangan tersebut bergantung pada isi pesan dan kunci publik yang digunakan oleh pengirim.

Digital Signature Algorithm (DSA)

Ada berbagai algoritma yang digunakan untuk memberikan tanda tangan digital pada suatu pesan. Pada bulan Agustus 1991, NIST (*The National Institute of Standard and Technology*) mengumumkan algoritma tanda tangan digital yang disebut *Digital Signature Algorithm* (DSA). DSA kemudian dijadikan sebagai standar dari *Digital Signature Standard* (DSS). Selain menggunakan algoritma DSA, DSS juga menggunakan sebuah algoritma SHA(*Secure Hash Algorithm*) sebagai fungsi *hash*-nya.

DSA merupakan salah satu bentuk algoritma kriptografi kunci publik. Meskipun demikian, DSA tidak dapat digunakan untuk enkripsi. DSA mempunyai dua fungsi utama:

1. Pembentukan tanda tangan digital (*signature generation*)
2. Pemeriksaan keabsahan tanda tangan digital (*signature verification*)

Selayaknya algoritma kriptografi kunci publik pada umumnya, DSA menggunakan dua buah kunci, kunci publik dan kunci privat. Untuk membentuk tanda tangan digital digunakan kunci privat pengirim, sedangkan untuk memverifikasi atau menguji keabsahan tanda tangan yang terdapat dalam dokumen digunakan kunci publik milik pengirim pesan.

Algoritma DSA menggunakan parameter-parameter sebagai berikut :

1. p, adalah bilangan prima dengan panjang L bit, yang dalam hal ini
   $512 \leq L \leq 1024$
   dan L merupakan kelipatan 64.
   Parameter p bersifat publik dan dapat digunakan bersama-sama oleh orang di dalam kelompok.

2. q, bilangan prima 160 bit, merupakan faktor dari p – 1. Dengan kata lain,
   $(p - 1) \bmod q = 0$.
   Parameter q bersifat publik.

3. $g = h^{(p-1)/q} \bmod p$,
   yang dalam hal ini
   $h < p - 1$
   sedemikian sehingga
   $h^{(p-1)/q} \bmod p > 1$
   Parameter g bersifat publik.

4. x, adalah bilangan bulat kurang dari q. Parameter x adalah kunci rahasia.

5. Kunci publik y
   $y = g^x \bmod p$

6. m, pesan yang akan diberi tanda tangan digital.

Dari beberapa parameter-parameter tersebut di atas dibentuklah sebuah pasangan kunci, kunci publik dan kunci privat dengan langkah-langkah sebagai berikut:

1. Pilih bilangan prima p dan q, yang dalam hal ini
   $(p - 1) \bmod q = 0$.

2. Hitung
   $g = h^{(p-1)/q} \bmod p$,
   yang dalam hal ini
   $1 < h < p - 1$ dan
   $h^{(p-1)/q} \bmod p > 1$.

3. Tentukan kunci rahasia x, yang dalam hal ini $x < q$.

4. Hitung kunci publik
   $y = g^x \bmod p$.

Setelah pasangan kunci tercipta langkah berikutnya ialah membubuhkan tanda tangan digital pada dokumen atau pesan yang akan dikirimkan. Adapun langkah-langkah pemberian tanda tangan digital dapat dijelaskan sebagai berikut:

1. Ubah pesan m menjadi *message digest* dengan fungsi *hash* SHA, H. Penjelasan mengenai SHA akan diberikan pada bagian berikutnya.

2. Tentukan sebuah bilangan acak k di mana k < q.

3. Tanda tangan digital dari pesan m terdiri dari dua buah bilangan, r dan s. Penentuan nilai r dan s diperoleh dari perhitungan sebagai berikut:
   $r = (g^k \bmod p) \bmod q$
   $s = (k^{-1}(H(m) + x*r)) \bmod q$

Setelah nilai r dan s diperoleh, proses berikutnya sebelum pesan dikirimkan ialah membubuhkan tanda tangan digital kepada pesan yang akan dikirimkan. Nilai r dan s ditambahkan pada dokumen atau pesan yang akan dikirimkan terlebih dahulu. Setelah itu pesan yang telah ditandatangani dapat dikirimkan ke tujuannya.

Setibanya di tangan penerima, pesan dapat diverifikasi keabsahannya. Penerima pesan dapat melakukan verifikasi pesan yang diterimanya dengan langkah-langkah sebagai berikut:

1. Hitung
   $w = s^{-1} \bmod q$
   $u1 = (H(m)*w) \bmod q$
   $u2 = (r*w) \bmod q$
   $v = ((g^{u1} * y^{u2}) \bmod p) \bmod q)$

2. Bandingkan nilai v yang diperoleh dari perhitungan pada poin pertama dengan nilai r yang dikirimkan beserta pesan. jika nilai v yang diperoleh melalui perhitungan tersebut bernilai sama dengan nilai r yang diterima, maka pesan tersebut tidak mengalami modifikasi pada saat pengiriman. Pengirim pesan tersebut juga dapat dipastikan keabsahannya.

Pada dasarnya DSA merupakan sebuah algoritma tanda tangan digital yang *robust*, karena panjang nilai p yang berada antara 512 hingga 1024 bitdan nilai g sebesar 160 bit. Akan tetapi untuk mengimplementasikan algoritma ini pada perangkat lunak hampir tidak mungkin. Oleh karena itu nilai p dan q biasanya diberi batasan nilainya hingga 2^{32} saja.

Berikut ialah contoh penggunaan algoritma DSA untuk memberi tanda tangan digital pada dokumen dan untuk memverifikasi keabsahan dokumen tersebut:

- Pembentukan Sepasang Kunci
  1. Pilih bilangan prima p dan q, yang dalam hal ini (p – 1) mod q = 0.
     $p = 59419$
     $q = 3301$
     (memenuhi 3301 * 18 = 59419 – 1)

  2. Hitung $g = h^{(p-1)/q} \bmod p$, yang dalam hal ini $1 < h < p - 1$ dan $h^{(p-1)/q} \bmod p > 1$.
     $g = 18870$
     (dengan h = 100)

  3. Tentukan kunci rahasia x, yang dalam hal ini x < q.
     $x = 3223$

  4. Hitung kunci publik $y = g^x \bmod p$.
     $y = 29245$

  Didapat parameter-parameter algoritma DSA sebagai berikut
  $p = 59419$
  $q = 3301$
  $g = 18870$
  $x = 3223$
  $y = 29245$

- Pembentukan Sidik Dijital (*Signing*)
  1. Hitung nilai hash dari pesan, misalkan
     $H(m) = 4321$
  2. Tentukan bilangan acak k < q.

$k = 997$

$k^{-1} = 2907 \pmod{3301}$

3. Hitung r dan s sebagai berikut:

$r = (g^k \bmod p) \bmod q$

$= 848$

$s = (k^{-1}(H(m)+x*r)) \bmod q$

$= 7957694475 \bmod 3301$

$= 183$

4. Kirim pesan m dan sidik dijital r dan s.

Didapat nilai-nilai :

$r = 848$

$s = 193$

- Verifikasi Keabsahan Sidik Dijital

1. Hitung

$s^{-1} = 469 \pmod{3301}$

$w = s^{-1} \bmod q$

$= 469$

$u1 = (H(m) * w) \bmod q$

$= 2026549 \bmod 3301$

$= 3036$

$u2 = (r * w) \bmod q$

$= 397712 \bmod 3301$

$= 1592$

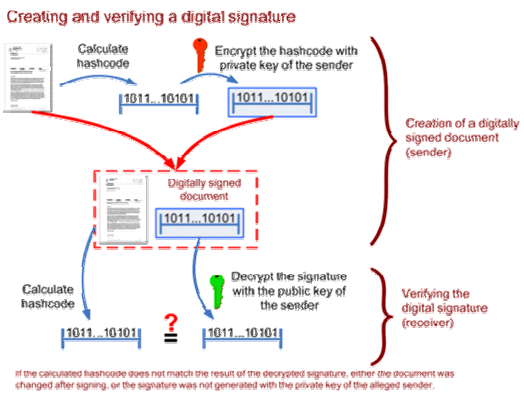
$v = ((g^{u1} * y^{u2}) \bmod p) \bmod q)$

$= 848 \bmod 3301$

$= 848$

2. Karena v = r, 848, maka tanda tangan digital sah. Dengan demikian, pesan tidak mengalami modifikasi sebelum diterima. Pengirim pesan juga adalah orang yang sah.

Secara umum, cara kerja tanda tangan digital dalam menangani masalah keamanan pesan seperti yang telah dijelaskan di atas dapat dipersingkat melalui gambar berikut



Secure Hash Algorithm (SHA)

Secure Hash Algorithm atau yang lebih dikenal dengan nama SHA adalah sebuah fungsi hash satu arah yang diciptakan oleh NIST (*The National Institute of Standard and Technology*). Penggunaan SHA biasanya ialah sebagai fungsi hash pada DSS(*Digital Signature Standard*) untuk menghasilkan *message digest* yang akan digunakan bersama-sama dengan DSA(*Digital Signature Algorithm*).

SHA merupakan pengembangan dari algoritma MD4 milik Ronald R Rivest. SHA menjadi standar fungsi *hash* karena tingkat keamanannya yang sangat tinggi. SHA dirancang sedemikian hingga tidak mungkin menemukan pesan yang berkoresponden dengan *message digest* yang diberikan. Algoritma ini juga mampu menangani pesan yang sangat panjang. Panjang pesan yang dapat diterima SHA sebagai masukan mencapai 2^{64} bit atau 2.147.483.648 *gigabyte* maksimum. Panjang *message digest* yang dihasilkan oleh SHA sendiri ialah 160 bit.

Secara umum, pembangkitan nilai *hash* dari suatu pesan terdiri atas 4 langkah :

1. Penambahan *padding bits*
2. Penambahan panjang pesan
3. Inisialisasi penyangga (*buffer*) *message digest*
4. Pengolahan pesan dalam blok-blok berukuran 512 bit

Masing-masing tahap tersebut akan dijelaskan sebagai berikut.

Tahap pertama yaitu penambahan *padding bits* pada pesan. Pesan ditambahkan sejumlah *padding bits* sehingga panjang pesan kongruen dengan 448 dalam modul 512. SHA mengolah pesan menjadi *message digest* dalam blok-blok pesan sebesar 512 bit masing-masingnya. Jika panjang pesan tepat 448, maka pesan tetap ditambahkan *padding bits* sebanyak 512 bit, sehingga panjang pesan menjadi 960 bit. Sehingga panjang *padding bits* berada diantara 1 hingga 512 bit. *Padding bits* yang ditambahkan ialah sebuah bit 1 yang diikuti dengan sejumlah bit 0, tergantung panjang *padding bits* itu sendiri. Misalnya *padding bits* yang perlu ditambahkan ialah sepanjang 7 maka *padding bits*-nya ialah **1000000**.

Setelah *padding bits* ditambahkan maka panjang pesan menjadi kongruen dengan 448 dalam modulo 512. Langkah berikutnya ialah menambahkan lagi 64 bit pada pesan sehingga

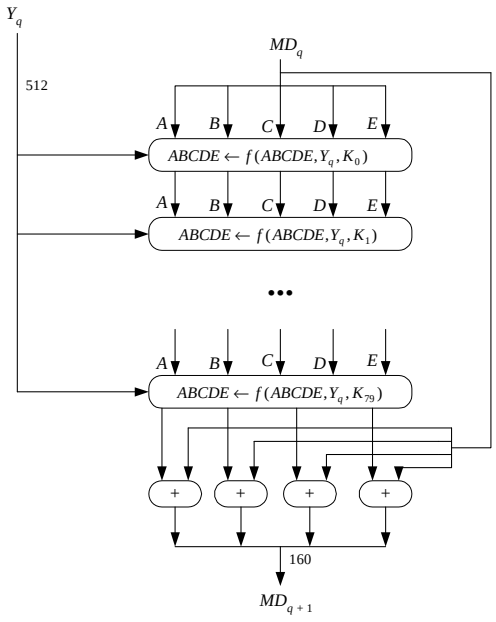
panjangnya menjadi kongruen dengan 0 dalam modulo 512. Dengan kata lain panjang pesan ialah kelipatan 512.

SHA membutuhkan 5 buah penyangga (*buffer*) yang masing-masing panjangnya 32 bit. Total panjang penyangga adalah 5 x 32 = 160 bit. Keempat penyangga ini menampung hasil sementara dan hasil akhir *message digest*. Kelima penyangga tersebut ialah sebagai berikut beserta nilai awalnya dalam notasi HEX

```
A = 67452301
B = EFCDAB89
C = 98BADCFE
D = 10325476
E = C3D2E1F0
```

Untuk menghasilkan *message digest* dari pesan, algoritma SHA terlebih dahulu mempartisi pesan menjadi sejumlah L blok yang masing-masingnya berukuran 512 bit ($Y_0 - Y_{L-1}$) .

Tiap-tiap blok yang berukuran 512 bit tersebut diproses bersama dengan kelima penyangga yang telah disebutkan sebelumnya. Proses ini, proses H_{SHA}, akan menghasilkan keluaran sepanjang 128 bit. Proses ini dapat digambarkan sebagai berikut:



Pada gambar tersebut terlihat bahwa untuk menghasilkan keluaran 128 bit dari blok pesan 512 bit dilakukan 80 putaran (80 t). Masing-masing putaran menggunakan sebuah bilangan K sebagai bilangan penambah. Bilangan K ini berubah tiap 20 kali putaran sehingga terdapat 4 buah nilai K yaitu :

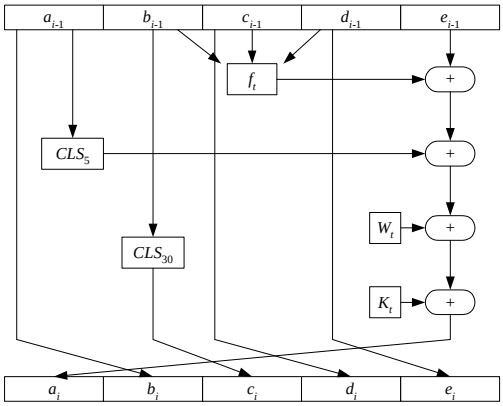
Putaran $0 \le t \le 19$ K_t = `5A827999`

Putaran $20 \le t \le 39$ K_t = `6ED9EBA1`

Putaran $40 \le t \le 59$ K_t = `8F1BBCDC`

Putaran $60 \le t \le 79$ K_t = `CA62C1D6`

Setiap putaran pada proses H_{SHA} menggunakan operasi yang sama (fungsi *f*). Operasi tersebut dapat dilihat pada gambar berikut:



Operasi dasar SHA yang diperlihatkan pada Gambar 3 dapat ditulis dengan persamaan sebagai berikut:

$$a \leftarrow (CLS_5(a) + f_t(b, c, d) + e + W_t + K_t)$$
$$b \leftarrow a$$
$$c \leftarrow CLS_{30}(b)$$
$$d \leftarrow c$$
$$e \leftarrow d$$

di mana

a, b, c, d, e	= variabel penyangga 32-bit
t	= putaran, $0 \le t \le 79$
f_t	= fungsi logika
CLSs	= *Circular Left Shift* sebanyak s bit
Wt	= *word* 32-bit yang diturunkan dari blok 512 bit yang sedang diproses
Kt	= konstanta penambah
+	=operasi penjumlahan modulo 2^{32}

Fungsi f_t ialah fungsi logika yang melakukan operasi *bitwise* pada putaran tertentu. Fungsi ini dapat dilihat pada tabel berikut

Putaran	$f_t(b, c, d)$
0 .. 19	$(b \wedge c) \vee (\sim b \wedge d)$
20 .. 39	$b \oplus c \oplus d$
40 .. 59	$(b \wedge c) \vee (b \wedge d) \vee (c \wedge d)$
60 .. 79	$b \oplus c \oplus d$

Nilai W_1 sampai W_{16} berasal dari 16 word pada blok yang sedang diproses, sedangkan nilai W_t berikutnya didapatkan dari persamaan :

$$W_t = W_{t-16} \oplus W_{t-14} \oplus W_{t-8} \oplus W_{t-3}$$

Setelah putaran ke-79, *a, b, c, d,* dan *e* ditambahkan ke *A, B, C, D,* dan *E* dan selanjutnya algoritma memproses untuk blok data berikutnya (Y_{q+1}). Keluaran akhir dari algoritma SHA adalah hasil penyambungan bit-bit di *A, B, C, D,* dan *E*.

Kelemahan Digital Signature

Salah satu karakteristik dari digital signature ialah *universal verifiability*. *Universal Verifiability* maksudnya ialah bahwa dokumen atau pesan yang telah dibubuhi tanda tangan digital dapat diverifikasi otentikasi dan integritasnya oleh siapapun yang mengetahui kunci publik pemberi tanda tangan. Di suatu sisi karakteristik ini menjamin bahwa pemberi tanda tangan pada dokumen atau pesan adalah orang yang memiliki otoritas untuk melakukannya. Namun di sisi lain, karakteristik ini pula yang menjadi kelemahan dari penggunaan tanda tangan digital.

Karakteristik *universal verifiability* terkadang tidak dibutuhkan pada suatu sistem, malah pada sistem-sistem tertentu karakteristik ini tidak diinginkan. Adanya kebutuhan dalam bentuk privasi pemberi tanda tangan serta keinginan untuk membatasi penyebaran tanda tangan digital seseorang merupakan beberapa diantaranya.

Pemberi tanda tangan digital pada suatu dokumen atau suatu pesan bersedia tanda tangannya diverifikasi oleh orang lain untuk menjamin integritas pesan yang ia sampaikan. Akan tetapi jika pesan yang ia kirimkan merupakan pesan yang bersifat privat dan rahasia tentunya ia hanya ingin identitasnya diketahui oleh penerima pesan tersebut. Jika ternyata ada orang lain yang menerima pesannya dan ingin memverifikasi apakah pesan itu memang ia yang menulis maka ia memiliki peluang untuk menyangkal bahwa ia yang menulis pesan tersebut. Hal tersebut berguna untuk menjaga privasi pengirim pesan.

Pada kesempatan lain pemberi tanda tangan digital juga tidak ingin tanda tangannya menyebar tidak terkendali. Oleh karena itu ia perlu untuk membatasi penyebaran tanda tangannya. Misalnya pada sistem yang terkait dengan usaha komersial, seseorang tentunya tidak mau semua orang dapat membaca dan mengkonfirmasi kontrak yang telah ia buat dengan pihak lain.

Sebagai contoh dapat diambil masalah seorang pedagang yang memberi harga spesial pada salah satu kliennya. Untuk menandai persetujuan, ia menandatangani sebuah kontrak digital dengan tanda tangan digitalnya. Akan tetapi klien lain mengetahui kontrak tersebut dan memverifikasi tanda tangan pedagang tersebut. Setelah diverifikasi tentunya pedagang tidak dapat menyangkal keabsahan tanda tangannya, akibatnya klien lain tersebut juga menuntut harga yang sama dengan yang tertera pada kontraknya dengan klien lain sebelumnya.

Tentunya dalam masalah tersebut di atas, pedagang tidak menginginkan tanda tangannya dapat diverifikasi oleh orang yang ia tidak inginkan. Namun karakteristik dari digital signature yaitu *universal verifiability* tidak memungkinkannya untuk membatasi verifikasi. Oleh karena itulah ia membutuhkan skema undeniable signature.

Undeniable Signature

Undeniable Signature ialah skema tanda tangan digital di mana untuk memverifikasi tanda tangan digital diperlukan persetujuan pemberi tanda tangan, biasanya proses verifikasi berlangsung dengan sepengetahuan pemberi tanda tangan. Skema tersebut disebut juga *non-self-athenticating signature schemes*. Jika sebuah tanda tangan digital hanya dapat diverifikasi dengan persetujuan pemberi tanda tangan, pemberi tanda tangan tersebut mungkin saja menolak ketika diminta untuk mengotentifikasi dokumen yang ia tanda tangani. *Undeniable Signature* dapat memecahkan masalah seperti ini dengan menggunakan sebuah protokol yang disebut *disavowal protocol* pada skema tanda tangan digital. Skema *undeniable signature* pertama kali diusulkan oleh David Chaum dan Hans van Antwerpen dalam makalah mereka ([CV90] dan [CV91]).

Skema undeniable signature diimplementasikan dengan menggunakan kriptografi kunci publik dengan berbasiskan masalah logaritma diskrit. Bagian penandatanganan dari skema ini mirip dengan algoritma lain yang juga menggunakan logaritma diskrit. Hanya saja pada proses verifikasi mengalami modifikasi. Proses verifikasi dilakukan dengan menggunakan protokol tanya-jawab di mana pihak yang ingin memverifikasi pesan atau dokumen mengirimkan pertanyaan kepada pemberi tanda tangan dan melihat jawaban yang diberikan untuk memverifikasi tanda tangan pada dokumen atau pesan yang ia terima. Proses penyangkalan juga berlangsung mirip dengan proses verifikasi. Penerima mengirimkan pertanyaannya dan respons dari pengirim pesan menunjukkan bahwa tanda tangan yang terdapat pada dokumen tersebut bukanlah tanda tangannya. Kemungkinan bahwa pemberi tanda tangan berhasil menipu pemverifikasi tanda tangan baik pada proses verifikasi maupun proses penyangkalan ialah sebesar 1/p di mana p adalah bilangan prima yang digunakan sebagai kunci privat pemberi tanda tangan. Rata-rata panjang kunci privat ialah 768 bit, oleh karena itu kemungkinan bahwa pemberi tanda tangan dapat melakukan penyangkalan bernilai sangat kecil sekali.

Undeniable signature memiliki dua buah fitur utama :

1. Verifikasi tanda tangan berlangsung secara interaktif. Pemberi tanda tangan dapat membatasi siapa yang dapat memverifikasi tanda tangannya

2. Protokol penyangkalan (*disaproval protocol*). Yaitu sebuah protokol untuk menyatakan bahwa tanda tangan digital yang diberikan merupakan tanda tangan yang palsu.

Fitur pertama mengimplikasikan bahwa pemberi tanda tangan dapat membatasi proses verifikasi sehingga hanya dapat dilakukan oleh orang-orang yang memiliki otoritas untuk membaca pesan atau dokumen yang ia tanda tangani. Akibatnya jika pesan atau dokumen tersebut terbaca oleh pihak lain yang tidak memiliki otoritas, maka pihak ketiga tadi tidak dapat melakukan verifikasi terhadap dokumen itu.

Akan tetapi akibat adanya fitur pertama tersebut, pemberi tanda tangan bisa jadi mengingkari tanda tangan yang sebenarnya valid. Untuk mencegah hal tersebut maka digunakanlah properti yang kedua, sebuah metode untuk membuktikan bahwa sebuah tanda tangan digital adalah tanda tangan digital yang palsu.

Secara umum undeniable Signature memiliki tiga buah karakteristik utama yaitu:

1. Tanda tangan digital hanya dapat di verifikasi dengan kerja sama pemberi tanda tangan, namun tetap memiliki kapabilitas anti-penyangkalan.
2. Pemverifikasi tidak dapat mengecek validitas sebuah tanda tangan digital sendiri.
3. Jika pemberi tanda tangan tidak bersedia untuk bekerja sama dalam memverifikasi pesan atau ia tidak dapat dimintai kerjasamanya maka penerima pesan juga tidak dapat memverifikasi pesan yang ia terima.

Kegunaan lain dari undeniable signature yaitu pada kasus perusahaan perangkat lunak yang menggunakan tanda tangan digital sebagai sarana untuk mengotentifikasi bahwa produk perangkat lunak yang mereka buat hanya dapat digunakan oleh pelanggan yang benar-benar berhak untuk menggunakannya. Misalnya hanya pelanggan yang telah membayar.

Telah banyak terdapat skema *undeniable signature* yang diusulkan. Di antaranya ialah *MOVA undeniable signature* yang diusulkan oleh Jean Monnerat dan Serge Vaudenay, *convertible undeniable signature* yang memiliki varian yang cukup banyak, *ID-Based undeniable signature*, *RSA-based undeniable signature*, *group undeniable signature* yang dajukan oleh Lyuu dan Wu, lalu ada pula *anonymous undeniable signature* dengan berbagai pengembangannya. Dari sekian banyak skema undeniable signature, diambil tiga buah skema yang cukup menarik untuk dibahas yaitu *confirmer signature scheme*, *fair anonymous undeniable signature* dan *group undeniable signature*. Pembahasan ketiga skema tersebut dapat ditemui pada bagian berikut.

Confirmer Signature Scheme

Confirmer signature scheme yang juga dikenal dengan nama *ID-Based undeniable signature* diusulkan oleh S. Han, K.Y. Yeung dan J. Wang. *Confirmer signature* ialah salah satu bentuk dari skema *undeniable signature* di mana tanda tangan digital juga dapat diverifikasi dengan melakukan interaksi dengan seorang *confirmer*

yang telah ditunjuk oleh pemberi tanda tangan. Konsep ini juga dikenal dengan istilah *pairing*.

Confirmer signature menggunakan parameter-parameter berikut {G_1, G_2, e, q, P, H, H_0}. G_1 adalah sebuah grup tambahan yang bersifat siklis yang diciptakan dari P. grup tersebut lalu diurutkan berdasarkan bilangan prima q. G_2 ialah hasil perkalian sendiri dari G_1, urutan pada G_2 juga berdasarkan bilangan prima q. e : G_1 X G_1 → G_2 ialah *pairing* bilinear. H dan H_0 adalah fungsi *hash* kriptografis. H : {0,1}* → Z_q dan H_0 : {0,1}* → G_1. Diketahui A, sebuah bilangan bulat kira-kira sebesar 10^{20} dan [A] = {1,2,...,A} diketahui oleh pemverifikasi dan pemberi tanda tangan digital.

Cara kerja skema *confirmer signature* dapat digambarkan dengan langkah-langkah sebagai berikut:

1. Langkah inisialisasi
   Key Generation Centre (KGC) memilih bilangan acak s $\in$ Z_q^* dan mengeset nilai P_{pub} = sP, serta menyimpan nilai s sebagai kunci utama yang tidak diketahui oleh pihak lain.

2. Langkah ekstraksi
   Pemberi tanda tangan memberikan informasi mengenai identitasnya, ID $\in$ {0,1}* kepada KGC. *Key Generation Center* lalu menghitung kunci publik dari pemberi tanda tangan sebagai Q_{ID} = H_0(ID). dan mengembalikan D_{ID} = sQ_{ID} dan L_{ID} = $s^{-1}Q_{ID}$ kepada pemberi tanda tangan sebagai kunci privatnya.

3. Pembubuhan tanda tangan
   Untuk memberikan tanda tangan pada pesan m, pertama-tama pemberi tanda tangan memilih suatu bilangan k $\in$ Z_q^* secara acak. Lalu dihitung nilai r = kP dan S = $k^{-1}D_{ID}$ + H(m)L_{ID}. Lalu r dan S bersama menjadi tanda tangan digital pada dokumen atau pesan m. Kemudian m dikirim beserta tanda tangan yang telah dibubuhkan oleh pemberi tanda tangan yaitu {r, S}.

4. Konfirmasi
   Untuk melakukan konfirmasi terhadap sebuah tanda tangan digital {r, S} pada pesan m :
   - pemverifikasi memilih suatu nilai x $\in$ [A], dan y $\in$ Z_q^* secara acak. Lalu nilai-nilai berikut ditentukan C_1 = xyr, C_2 = xyP lalu mengirimnya ke pemberi tanda tangan digital pesan m tersebut.
   - Pemberi tanda tangan menghitung nilai X = e(r + P_{pub}, P - L_{ID}) dan R = e(C_1, L_{ID}) dan mengirimkannya ke pemverifikasi.
   - Pemverifikasi lalu mengecek apakah
     $e(r,S)^x = e(P_{pub}, Q_{ID})^x R^{H(m)y^{-1}}$
     dan
     $R^{y^{-1}} X^x e(P, Q_{ID})^x = e(r + P_{pub}, P)^x$
     bernilai benar. Jika persamaan-persamaan di atas bernilai benar maka pemverifikasi dapat menyimpulkan bahwa tanda tangan tersebut otentik. Jika persamaan-persamaan tersebut tidak benar maka validitas tanda tangan digital pada pesan tidak dapat ditentukan.

5. Penyangkalan
   Untuk melakukan penyangkalan atas tanda tangan yang tidak valid:
   - pemverifikasi memilih suatu nilai x $\in$ [A], dan y $\in$ Z_q^* secara acak. Lalu nilai-nilai berikut ditentukan C_1 = xyr, C_2 = xyP lalu mengirimnya ke pemberi tanda tangan digital pesan m tersebut.
   - Pemberi tanda tangan lalu menghitung nilai B
     $B = \frac{e(C_1, S)}{e(C_2, D_{ID}) e(C_1, L_{ID})^{H(m)}}$
     lalu dikirimkan kepada pemverifikasi.
   - Pemverifikasi mengirim nilai C kepada penanda tangan.
     $C = B^{y^{-1}}$
   - Pemberi tanda tangan menghitung nilai x` dari C dan mengirimkan hasilnya, x`, kepada pemverifikasi.
   - Pemverifikasi mengecek apakah x` = x bernilai benar. Jika ya maka pemverifikasi mengetahui bahwa ia tidak berhak untuk melakukan proses verifikasi. Jika x` = x bernilai salah maka invaliditas tanda tangan tidak dapat ditentukan.

Pada konferensi bertajuk "*Fourth ACM Conference on Electronic Commercial*" para pencetus *confirmer signature* menyatakan bahwa konsep yang mereka bawa memiliki tingkat keamanan yang baik. Akan tetapi Fangguo Zhang, Raihaneh Safavi-Naini dan Willy Susilo berhasil membuktikan kelemahan *confirmer*

signature dengan dua jenis serangan. *Denial attack* dan *forge attack*

Denial attack yang diusulkan oleh Zhang dkk. dapat diartikan sebagai penyangkalan yang dilakukan oleh pemberi tanda tangan digital pada pesan meskipun tanda tangan yang akan diverifikasi adalah tanda tangannya yang benar-benar valid.

Misalkan {r, S} adalah tanda tangan digital yang valid miliki pengirim pesan, akan tetapi ia ingin menyangkal tanda tangan tersebut

1. pemverifikasi mengeset nilai-nilai berikut : C_1 = xyr, C_2 = xyP lalu mengirimnya ke pemberi tanda tangan digital pesan m tersebut.

2. pemberi tanda tangan memilih nilai $\alpha \in Z_q^*$ dan menghitung nilai B = e(C_2, αP) lalu mengirimkannya pada pemverifikasi.

3. Pemverifikasi menghitung nilai C dan mengirimkannya kepada penanda tangan
   $C = B^{y^{-1}}$

4. Pemberi tanda tangan menghitung nilai x` dan mengirimkannya kembali ke pemverifikasi.

Karena $C = B^{y^{-1}} = e(P, \alpha P)^x$, pemberi tanda tangan dapat menguji x` dari [A] untuk C dan menemukan nilai x` yang tepat. Pemverifikasi dapat mengecek x` = x bernilai benar, oleh karena itu ia percaya bahwa tanda tangan itu bukanlah tanda tangan milik pemberi tanda tangan.

Serangan berikutnya ialah misalkan Trudy ingin menipu penerima pesan dengan menggunakan identitas ID pada pesan yang ia tanda tangani. Lalu ia dapat membuktikan kepada pemverifikasi bahwa tanda tangan palsunya valid. Serangan seperti itu disebut *forge attack*.

Untuk melakukan *forge attack*, Trudy dapat melakukan

1. Pertama sekali Trudy memilih k, $\beta \in Z_q^*$ secara acak. Lalu menghitung r = kP_{pub} dan $S=k^{-1}(Q_{ID}+ \beta H(m)P)$. Lalu ia membubuhkan tanda tangan {r, S} pada pesan m.

2. Konfirmasi

   - pemverifikasi memilih suatu nilai x $\in$ [A], dan y $\in Z_q^*$ secara acak. Lalu nilai-nilai berikut ditentukan C_1 = xyr, C_2 = xyP lalu mengirimnya ke pemberi tanda tangan digital pesan m tersebut.

   - Trudy menghitung nilai X dan R
     $X = e(r + P_{pub}, P)e(P, Q_{ID})^{-1}e(P_{pub}, \beta P)^{-1}$
     $R = e(\beta P_{pub}, C_2)$
     lalu kedua nilai tersebut, X dan R, dikirimkan ke pemverifikasi

   - Pemverifikasi mengecek apakah
     $e(r, S)^x = e(P_{pub}, Q_{ID})^x R^{H(m)y^{-1}}$
     $R^{y^{-1}} X^x e(P, Q_{ID})^x = e(r + P_{pub}, P)^x.$
     bernilai benar. Akan tetapi karena

$$
\begin{aligned}
& e(r, S)^x \\
&= e(kP_{pub}, k^{-1}(Q_{ID} + \beta H(m)P))^x \\
&= e(P_{pub}, Q_{ID})^x e(P_{pub}, \beta H(m)P)^x \\
&= e(P_{pub}, Q_{ID})^x e(\beta P_{pub}, xyP)^{H(m)y^{-1}} \\
&= e(P_{pub}, Q_{ID})^x R^{H(m)y^{-1}}
\end{aligned}
$$

dan

$$
\begin{aligned}
& R^{y^{-1}} X^x e(P, Q_{ID})^x \\
&= e(\beta P_{pub}, C_2)^{y^{-1}} (e(r + P_{pub}, P) e(P, Q_{ID})^{-1} \\
&\qquad e(P_{pub}, \beta P)^{-1})^x e(P, Q_{ID})^x \\
&= e(r + P_{pub}, P)^x
\end{aligned}
$$

Pemverifikasi akan percaya bahwa tanda tangan {r, S} untuk pesan m adalah valid.

Dengan berhasilnya serangan yang dilakukan oleh Zhang dkk., maka skema yang diusulkan oleh Han dkk. ini terlihat jelas kelemahannya. Yaitu adanya kemungkinan pemberi tanda tangan memberikan penyangkalan terhadap tanda tangan yang telah ia berikan pada suatu pesan. Kelemahan lainnya ialah adanya kemungkinan seseorang menyamar sebagai pemberi tanda tangan yang sah dari suatu pesan.

Fair Anonymous Undeniable Signature

Sejauh ini sudah banyak skema undeniable signature yang telah diciptakan. Skema-skema tersebut menyediakan analisis mengenai properti-properti seperti *completeness*

(kelengkapan), *soundness* (kebaikan) dan *zero-knowledge*.

Akan tetapi, akan lebih menarik lagi jika menambahkan unsur anonim ke dalam skema *digital signature* yang digunakan. S. D. Galbraith dan W. Mao telah mengusulkan konsep ini dalam makalah mereka yang berjudul "*Invisibility and Anonymity of Undeniable and Confirmer Signatures*". Namun skema yang mereka buat hanya mengajukan keanoniman yang sempurna. Dalam hal ini skema yang mereka ajukan selalu menjaga privasi pemberi tanda tangan dengan demikian pemberi tanda tangan memiliki privasi yang sempurna. Oleh karena itu muncul pertanyaan seperti bagaimana bisa kita mengidentifikasi pemberi tanda tangan yang menggunakan skema ini bertindak secara legal atau tidak.

Song Han, Elizabeth Chang, Xiaotie Deng, Winson Yeung, dan Li Gao mengusulkan sebuah skema baru yang juga menyediakan unsur keanoniman, tetapi tidak memberikan pemberi tanda tangan privasi yang sempurna. Mereka menamakan skema mereka dengan *fair anonymous undeniable signature*.

Skema *fair anonymous undeniable signature*, selanjutnya disingkat menjadi *fair anonymous signature* saja, terdiri atas empat buah algoritma, *setup*, *key*, *sign* dan *DelAnonimity*. Selain empat buah algoritma tersebut juga terdapat dua buah protokol yaitu *confirmation protocol* dan *denial protocol*. Untuk setiap pilihan parameter keamanan k, terdapat sebuah ruang kunci publik $\underline{K}$, ruang pesan $\underline{M}$, dan ruang tanda tangan $\underline{S}$.

Setup ialah sebuah algoritma probabilistik waktu polinomial dengan masukan sebuah nilai k dan akan menghasilkan sekumpulan parameter yang dibutuhkan untuk skema *fair anonymous signature*.

Key ialah sebuah algoritma probabilistik waktu polinomial yang dieksekusi oleh pemberi tanda tangan digital dan oleh pihak yang dipercaya (*trusted center*). Masukan dari algoritma ini ialah parameter sistem yang didapat dari algoritma setup. Masukan lainnya berupa parameter-parameter yang bernilai acak yang disetujui oleh kedua pihak, pemberi tanda tangan dan *trusted centre*. Keluaran yang dihasilkan oleh algoritma ini berupa sebuah kunci publik pk $\in \underline{K}$, dan sebuah kunci privat sk yang berkoresponden dengan kunci publik yang dihasilkan.

Sign ialah sebuah algoritma probabilistik waktu polinomial yang menghasilkan keluaran berupa sebuah tanda tangan digital $Sig_{sk} \in \underline{S}$. Masukan dari algoritma ini yaitu kunci privat pemberi tanda tangan, sk, dan sebuah pesan $m \in M$. Secara umum terdapat banyak tanda tangan yang valid untuk pasangan nilai (m, pk) $\in \underline{M} \times \underline{K}$.

DelAnonymity ialah sebuah algoritma deterministik waktu polinomial yang dieksekusi hanya oleh *trusted center*. Masukan dari algoritma ini terdiri atas pesan $m \in \underline{M}$ tanda tangan digital $s \in \underline{S}$ dan sejumlah parameter sistem. Keluaran yang diberikan algoritma ini yaitu identitas pemberi tanda tangan digital $s \in \underline{S}$.

Confirmation ialah sebuah protokol determinisitik waktu polinomial antara pemberi tanda tangan dan pemverifikasi tanda tangan digital. Input dari protokol ini ialah sebuah pesan $m \in \underline{M}$, tanda tangan digital $s \in \underline{S}$ dan sebuah kunci publik pk yang telah tersertifikasi. Protokol ini mengizinkan pemberi tanda tangan untuk membuktikan kepada pemverifikasi tanda tangan bahwa untuk tanda tangan s bernilai valid untuk pesan m dan kunci publik pk. Jika pemverifikasi memiliki kunci publik yang bersesuaian maka proses pembuktian tidak perlu berjalan secara interaktif.

Denial ialah sebuah protokol deterministik waktu polinomial. Input dari protokol ini terdiri dari pesan $m \in \underline{M}$, tanda tangan digital $s \in \underline{S}$ dan kunci publik pk $\in \underline{K}$. Protokol ini menyediakan fasilitas bagi pemberi tanda tangan untuk menyangkal melakukan konfirmasi atas tanda tangannya.

Penjelasan lebih rinci langkah-langkah yang dilakukan untuk masing-masing algoritma dan protokol dijelaskan sebagai berikut

1. Algoritma *setup*
   Algoritma ini dieksekusi oleh *trusted center*. Langkah-langkah dari algoritma ini dapat digambarkan sebagai berikut:

   - *Trusted center* memilih N buah pasangan bilangan prima $\{p_i, q_i\}$ $(1 \leq i \leq N)$ dengan syarat $p_i \equiv q_i \equiv 3 \pmod 4$,

dan seluruh faktor prima dari $(p_i - 1)/2$ dan $(q_i - 1)/2$ bernilai lebih besar dari suat nilai batas B.

- *Trusted center* menghitung nilai-nilai berikut:
  $n_i \leftarrow p_i q_i$
  $e_i, d_i \xleftarrow{R} \phi_{n_i}$
  sehingga didapat
  $e_i d_i \equiv 1 \pmod{\varphi(n_i)}$

  di mana φ() adalah fungsi *phi euler*.

- Nilai – nilai $\{(n_i, p_i, q_i, e_i, d_i) \mid (1 \leq i \leq N)\}$ disimpan dalam basis data milik *trusted center*.

- *Trusted center* mengirim cipherteks hasil enkripsi dari *tuple* nilai $\{n_i, e_i, d_i\}$ kepada setiap pemberi tanda tangan yang telah terdaftar pada basis data. Dalam hal ini *trusted center* menggunakan sistem enkripsi kunci publik *Cramer-Shoup* untuk mengenkripsi nilai (n_i, e_i, d_i).

- Diberikan D = {0, 1}* yang merupakan ruang dokumen yang dapat diberi tanda tangan digital.

2. Algoritma *key*
   Langkah-langkah dari algoritma ini ialah sebagai berikut:

   - Tiap pemberi tanda tangan digital menerima cipherteks hasil enkripsi dari $\{n_i, e_i, d_i\}$ dari *trusted center* lalu melakukan proses dekripsi terhadap cipherteks tersebut.

   - Pemberi tanda tangan memilih nilai
     $x_i, y_i \xleftarrow{R} \phi_{n_i}^{*}$
     lalu menghitung nilai
     $g_i = (x_i, y_i)^2 \pmod{n_i}$
     $h_i = g_i^{e_i} \pmod{n_i}$

   - Akhirnya pemberi tanda tangan mendapatkan pasangan kunci publik dan kunci privatnya
     $SK_i \leftarrow \{n_i, e_i, d_i, x_i, y_i\}$ sebagai kunci privat dan $PK_i \leftarrow \{n_i, g_i, h_i\}$ sebagai kunci publiknya.

3. Algoritma *Sign*
   - Untuk menandatangani pesan M, pertama pemberi tanda tangan memilih
     $r \xleftarrow{R} \phi_{n_i}^{*}$
     lalu menghitung
     $s_1 \leftarrow x_i H(M \| r)^{(g_i - r)d_i} \pmod{n_i}$
     $s_2 \leftarrow y_i H(M \| r)^{rd} \pmod{n_i}$

   - Lalu pemberi tanda tangan menghitung nilai
     $s_c' = s_c + b_c n_i$
     untuk $c \in \{1, 2\}$
     dengan syarat b_c dipilih sedemikian sehingga $|s_c'| \leq k$

   - Tanda tangan digital yang dihasilkan ialah $\{s_1', s_2', H(M \| r)\}$ untuk pesan M.

4. Protokol *Confirmation*
   Protokol *Confirmation* ialah sebuah protokol deterministik waktu polinomial. Protokol ini bersifat interaktif antara pemberi tanda tangan dan pemverifikasi. Untuk mengkonfirmasi tanda tangan digital $\{s_1', s_2', H(M \| r)\}$, pemberi tanda tangan mengeksekusi bagian dari proses pembuktian yang tidak interaktif. Proses tersebut akan menghasilkan dua buah persamaan :
   1. $g_i \equiv h_i^{d_i} \pmod{n_i}$
   2. $(s_1' s_2' \pmod{n_i})^{2e_i} \equiv h_i H(M \| r)^{2g_i} \pmod{n_i}$

5. Protokol *Denial*
   Pada dasarnya protokol ini memiliki cara kerja yang sama dengan protokol *confirmation.*

6. Algoritma *DelAnonimity*
   - Jika pada suatu saat terjadi situasi "gawat", *trusted center* akan segera melakukan pencarian ke dalam basis data miliknya.

   - Untuk semua nilai i sebanyak N yang terdapat dalam basis data *trusted center* mengecek apakah
     $(s_1' s_2' \pmod{n_i})^{2e_i} \equiv h_i H(M \| r)^{2g_i} \pmod{n_i}$
     jika terdapat nilai i sehingga
     $(s_1' s_2' \pmod{n_i})^{2e_i} \equiv h_i H(M \| r)^{2g_i} \pmod{n_i}$
     maka *trusted center* dapat mengambil kesimpulan bahwa seorang pemberi tanda tangan yang memiliki indeks ke-i pada basis datanya telah memberikan tanda tangan digital $\{s_1', s_2', H(M \| r)\}$

Han dkk., pencetus skema *fair anonymous signatures* ini, mengklaim bahwa skema yang mereka buat memiliki tingkat keamanan yang tinggi dan performa yang handal. Untuk

meyakinkan bahwa skema tersebut benar-benar aman dan handal, maka berikut akan dijelaskan aspek keamanan dan performa dari skema tersebut.

Skema *fair anonymous signature* memiliki properti *unforgeability*, atau dengan kata lain tanda tangan digital yang dihasilkan tidak dapat dipalsukan dengan cara meniru tanda tangan orang lain.

Secara umum, dapat dikatakan bahwa *trusted center* adalah pihak yang memiliki otoritas paling tinggi untuk membentuk tanda tangan digital yang valid. Hal tersebut diakibatkan karena *trusted center*-lah yang mengetahui data mengenai pengguna secara lengkap. Oleh karena itu dapat dibuat pernyataan jika *trusted center* tidak dapat menirukan orang lain, maka tidak ada yang bisa.

Berdasarkan pengetahuan akan sifat logaritma diskrit yang merupakan masalah yang memiliki kompleksitas yang tinggi, dapat juga dinyatakan bahwa untuk nilai bilangan bulat g yang cukup besar, peluang untuk mendapatkan pasangan nilai x dan y yang mungkin dari persamaan $g_i = (x_i y_i)^e \pmod{n_i}$ sangat kecil, bahkan dapat diabaikan.

Selain itu skema ini juga diklaim aman terhadap serangan *chosen message attack*. Akan tetapi kondisi ini bergantung pada algoritma pemberian tanda tangan yang digunakan, apakah algoritma tersebut tahan akan *chosen message attack*. Jika algoritma tanda tangan digital yang digunakan tahan terhadap *chosen message attack*, misalnya menggunakan algoritma RSA, maka skema ini juga akan kebal terhadap serangan menggunakan *chosen message attack*.

Skema *fair anonymous signature* juga memenuhi ketiga syarat skema *undeniable signature*. *Completeness* dari skema ini dapat dilihat dari kebenaran bahwa jika nilai dari tanda tangan digital diketahui, dapat dibuktikan pula kebenaran dari pemberi tanda tangan.

Jika sebuah tanda tangan digital invalid, maka pemberi tanda tangan tidak memiliki peluang untuk menyatakan sebaliknya. Dalam hal ini syarat *soundness* dari sebuah skema *undeniable signature* terpenuhi.

Syarat terakhir yaitu *zero-knowledge* juga terpenuhi karena dengan skema ini, pada saat proses verifikasi tanda tangan digital, tidak terdapat informasi lain mengenai pemberi tanda tangan yang dapat diperoleh oleh pemverifikasi tanda tangan.

Performa dari skema ini dapat diukur dari kompleksitas algoritmanya. Operasi yang dilakukan didominasi oleh kalkulasi untuk logaritma diskrit dalam bentuk perpangkatan dan operasi modulo. Selain itu operasi yang juga dominan ialah pada proses enkripsi dan dekripsi menggunakan algoritma *Cramer-Shoup*. Jika waktu untuk melakukan operasi perkalian dan penjumlahan dalam modulo n dapat diabaikan, maka total *cost* untuk menghasilkan kunci dan memberikan tanda tangan digital pada pesan membutuhkan $O(k^3)$ operasi perpangkatan dalam modulo n.

Group Undeniable Signature

Skema *group undeniable signature* ialah sebuah skema di mana setiap anggota kelompok dapat memberikan tanda tangan atas nama kelompok tanpa menunjukkan identitasnya pribadi. Pada skema *group undeniable signature* ini proses verifikasi tanda tagan digital dilakukan dengan interaksi terhadap pengelola kelompok.

Pada aplikasi bisnis, *group undeniable signature* dapat digunakan ketika tanda tangan digital memiliki nilai komersial bagi kompetitor bisnis. Jika seorang anggota kelompok dituduh, secara salah, telah memberi tanda tangan digital pada suatu pesan ataupun dokumen, maka pengelola kelompok tersebut memiliki kemampuan untuk membuktikan bahwa anggota kelompoknya tidak bersalah. Untuk menghindari terjadinya perselisihan di kemudian hari, maka pengelola kelompok dapat menyusuri kembali anggota kelompok mana yang memberikan tanda tangan digitalnya atas nama kelompok.

Skema *group undeniable signature* menggabungkan konsep *undeniable signature* dengan konsep *group signature*. *Group signature* pertama sekali diajukan oleh Chaum dan van Heyst dalam makalah yang mereka tulis yang berjudul "*Group Signatures*". Lalu Camenisch dan Stadler juga menciptakan sebuah skema *undeniable signature* untuk kelompok di mana ukuran kunci publik dan ukuran tanda tangan digital tidak bergantung kepada jumlah anggota

kelompok. Skema mereka ini dipublikasikan dalam makalah yang berjudul "*Efficient Group Signature Schemes for Large Groups*" yang mereka tulis.

Lyuu dan Wu juga menawarkan sebuah skema *group undeniable signature* yang mereka beri nama *group undeniable signature*. Layaknya skema *undeniable signature* pada umumnya, skema yang mereka tawarkan ini juga memiliki properti *non-repudiatable* atau anti-penyangkalan dan *non-universally verifiable* atau dibutuhkan interaksi antara pemverifikasi pesan dan pemberi tanda tangan digital dalam proses konfirmasi tanda tangan digital yang terdapat dalam sebuah pesan.

Skema *group undeniable signature* mirip dengan skema *group signature* pada umumnya. Perbedaannya hanya terletak pada proses verifikasi yang berjalan secara interaktif dengan melibatkan pengelola kelompok. Penggunaan *group undeniable signature* ialah misalnya pada validasi daftar harga dan pada kontrak digital. Skema yang mereka buat diklaim tahan terhadap *chosen message attack*.

Skema *group undeniable signature* terdiri atas enam buah komponen:

1. *System* setup
   Pembentukan kunci publik dan kunci privat bagi kelompok. Nilai kunci-kunci ini diberikan pada pengelola kelompok.

2. *Join*
   Untuk menjadi anggota sebuah kelompok, seseorang pertama kali membentuk kunci privat dan kunci keanggotaannya. Pembentukan kunci ini dilakukan dengan persetujuan pengelola kelompok. Setelah proses itu, pengelola kelompok memberikan sertifikat keanggotaan bagi anggota baru tersebut.

3. *Sign*
   Seorang anggota kelompok memberikan tanda tangan digitalnya pada sebuah dokumen atau pesan dengan menggunakan kunci privat miliknya, kunci keanggotaan serta kunci publik milik kelompok.

4. *Signature Confirmation Protocol*
   Protokol yang digunakan mengkonfirmasi tanda tangan digital. Protokol ini membutuhkan interaksi dengan pengelola kelompok.

5. *Signature Denial Protocol*
   Pengelola kelompok dapat membuktikan kepada siapa pun bahwa sebuah tanda tangan invalid benar-benar tidak valid melalui penggunaan protokol ini.

6. *Open*
   Pengelola kelompok dapat menelusuri kembali siapa atau anggota kelompok mana yang memberikan tanda tangan digitalnya pada suatu pesan atau dokumen.

Group undeniable signature juga memiliki pertimbangan-pertimbangan keamanan sebagai berikut:

1. *Unforgeability*
   Hanya anggota kelompoklah yang dapat memberikan tanda tangannya atas nama kelompok.

2. *Unlinkability*
   Tidak ada yang dapat membuktikan bahwa lebih dari satu buah tanda tangan diberikan oleh anggota kelompok yang sama atau bukan kecuali pengelola kelompok.

3. *Anonymity*
   Hanya pengelola kelompok yang dapat mengetahui identitas pemberi tanda tangan digital.

4. *Non-transferability*
   Hanya pengelola kelompok yang dapat memberikan pernyataan akan validitas sebuah tanda tangan digital.

5. *Zero-knowledge*
   Protokol konfirmasi dan penyangkalan tidak menghasilkan informasi tambahan lainnya di luar informasi mengenai valid atau tidaknya sebuah tanda tangan.

6. *Exculpability*
   Tidak ada yang dapat memberikan tanda tangan digital atas nama kelompok lainnya oleh anggota suatu kelompok maupun pengelola kelompok tertentu.

7. *Traceability*
   Pengelola kelompok dapat menelusuri anggota kelompok yang mana yang memberikan tanda tangan digital.

8. *Coalition-resistance*

Beberapa orang anggota yang berkolusi tidak dapat memberikan tanda tangan digital mereka untuk menandatangani sebuah pesan tanpa dapat ditelusuri kembali oleh pengelola kelompok.

Efisiensi dari *group undeniable signature* dapat diukur dari parameter-parameter berikut ini:

- Ukuran tanda tangan
- Ukuran kunci publik kelompok
- Efisiensi dari *system setup*, *join*, dan *open*
- Efisiensi dari *sign* dan *verify* (termasuk di dalamnya *confirmation* dan *denial*)

Penutup

Tercetusnya skema *undeniable signature* oleh David Chaum dan Hans van Antwerpen merupakan salah satu sumbangan berharga pada dunia kriptografi pada umumnya dan pada bidang *digital signature* pada khususnya. Skema yang mereka hasilkan dapat memenuhi persyaratan tambahan yang muncul ketika menggunakan *digital signature*, yaitu *non-universal verifiable*.

Terdapat banyak sekali skema-skema *undeniable signature* yang telah berhasil diciptakan. Sebagian skema telah berhasil melewati ujian terhadap masalah keamanannya. Ada juga di antara skema tersebut yang telah berhasil ditemukan celah pada keamanannya.

Daftar Pustaka

1. D. Chaum, Hans van Antwerpen, "*Undeniable Signatures*", CRYPTO 1989, LNCS 435, Springer, pp.212-216, 1990

2. J.Boyar, D. Chaum, I.Damgard, T. Pedersen, "*Convertible Undeniable Signatures*" Advances in Cryptography – Crypto '90, LNCS 537, Springer, pp.189-205, 1990.

3. D. Chaum, "*Zero-knowledge Undeniable Signatures*", EUROCRYPT, 1990, LNCS 473, Springer, pp. 458-464, 1991.

4. I. Damgard, T. P. Pedersen, "*New Convertible Undeniable Signature Scheme*", EUROCRYPT 1996, LNCS 1070, Springer, pp. 372-386, 1996.

5. S. D. Galbraith, W. Mao, "*Invisibility and Anonymity of Undeniable and Confirmer Signatures*" CT-RSA 2003, LNCS 2612, Springer-Verlag, pp. 80-97, 2003.

6. S. D. Galbraith, Wenbo Mao, Kenneth G. Paterson, "*RSA-Based Undeniable Signatures for General Moduli*", CT-RSA 2002, LNCS 2271, pp.200-217, Springer, 2002.

7. S. Han, K. Y. Yeung, J. Wang, "*Identity based confirmer signatures from piring over elliptic curves*", Proceeding of ACM conference on Electronic Commerce, pp. 262-263. 2003.

8. S. S. M. Chow, L. C. K. Hui, S. M. Hiu, K. P. Chow, "*A Secure Modified ID-Based Undeniable Signature Scheme Based on Ha net al.'s Scheme against Zhang et al.'s attacks*" Cryptology ePrint Archive, pp. 262, 2003.

9. Jean Monnerat, S Vaudenay, "Undeniable Signatures" Advances in Cryptology PKC'04, Singapore, Lectures Notes in Komputer Science, Springer-Verlag, pp.69-85, 2004.

10. J. Monnerat, S Vaudenay, "Generic Homomorphonic Undeniable Signatures", Advances in Cryptology – Asiacrypt '04. LNCS 3329. Springer, pp. 354-371, Springer, 2004.

11. Jean Monnerat, Yvonne Anne Oswald, Serge Vaudenay, "*Optimization of MOVA Undeniable Signature Scheme*", EPFL, 2005.

12. Markus Michels, Markus Stadler, "*Efficient Convertible Undeniable Signature Scheme*", Fourth Internasional Workshop SAC '97. UBS, pp. 231-244, 1997.

13. Fangguo Zhang, Reihaneh Safavi-Naini, Willy Susilo, "*Attack on Ha net al.'s ID-Based Confirmer (Undeniable) Signatures at ACM-EC '03*", pp, 2003.

14. D. Chaum, "*Designated confirmer signatures*", EUROCRYPT 94, LNCS 950, pp. 86-91, Springer-Verlag, 1995.

15. J. Camenisch, M. Stadler. "*Efficient Group Signature Schemes for Large Groups*". In Advances in Cryptology CRYPTO '97, pp. 410–424, 1997.

16. D. Chaum, E. van Heyst. "*Group Signatures*". In Advances in Cryptology EUROCRYPT 91, pp. 257–265, 1991.

17. Yuh-Dauh Lyuu, Ming-Luen Wu, "*Group Undeniable Signature*", pp.

18. Rinaldi Munir. "*Diktat Kuliah IF5054 Kriptografi Program Studi Teknik Informatika Institut Teknologi Bandung*" 2006.