Implementasi *Blind Signature* dalam Melakukan *Electronic Voting*

Timotius Grady Limandra (13504082)
Jurusan Teknik Informatika ITB, email: if14082@students.if.itb.ac.id

Abstraksi – Makalah ini akan membahas mengenai bagaimana penggunaan *Blind Signature* dalam melakukan *Electronic Voting*. *Blind Signature* merupakan tandatangan digital yang memungkinkan untuk melakukan otentikasi yang menjaga privasi si pembubuh tandatangan. Biasanya kasus seperti ini terjadi pada kejadian yang membutuhkan otentikasi, tetapi isi dari dokumen yang bersangkutan bersifat rahasia. Kasus seperti ini terjadi pada saat kita melakukan *Electronic Voting* yang sangat membutuhkan otentikasi dari pemilih yang tepat, dan melihat hasil pilihannya tetapi kerahasiaan dari pilihan si pemilih harus tetap terjaga. Makalah ini nantinya akan membahas bagaimana *Blind Signature* dapat diimplementasikan untuk membantu menyelesaikan masalah ini.

Kata Kunci: *Blind Signature*, *Digital Signature*, *Electronic Voting*, RSA

1. PENDAHULUAN

Digital Signature, atau tanda tangan digital merupakan salah satu cara dalam kriptografi untuk melakukan penandatanganan pada suatu dokumen. Tanda tangan ini memiliki nilai kriptografis yang bergaantung pada pesan dan si pengirim pesan (tidak sama dengan tanda tangan konvensional yang sama pada tiap pesan asalkan pengirimnya sama). Dengan menggunakan tanda tangan digital ini, maka intregitas data dapat dijamin. Selain itu, dapat juga digunakan untuk membuktikan keabsahan pengirim dan nirpenyangkalan.

Blind Signature merupakan salah satu bentuk dari tanda tangan digital di mana pesan yang akan ditandatangani disamarkan terlebih dahulu sebelum dilakukan pembubuhan tanda tangan digital. Hal seperti ini biasanya digunakan jika dalam sebuah pesan dibutuhkan otentikasi si pengirim dan pesan tersebut tidak boleh diketahui milik siapa tetapi isi dari pesan tersebut nantinya harus dapat digunakan untuk dijadikan data dari suatu kegiatan. Contoh kasus dari penggunaan dari *blind signature* ini adalah ketika melakukan *electronic voting* atau transaksi digital. Pada makalah ini yang akan dibahas adalah penggunaan *blind signature* pada *electronic voting*.

Electronic Voting merupakan salah satu cara dalam melakukan pemilihan yang dilakukan secara digital. Dengan teknologi ini, suatu voting dapat dilangsungkan dalam waktu yang relatif jauh lebih singkat dibandingkan dengan cara konvensional. Selain itu, kelebihan dari teknologi ini adalah lebih memudahkan para calon pemilih untuk melakukan pemilihan karena pemilihan dapat dilaksanakan asalkan memiliki koneksi ke jaringan yang bersangkutan.

Electronic voting sudah diterapkan di berbagai negara seperti: Australia, Belgia, Brazil, Kanada, Estonia, Perancis, Jerman, India, Irlandia, Itali, Belanda, Norwegia, Romania, Swiss, dan Inggris. Meskipun demikian di beberapa negara seperti di Amerika Serikat, *electronic voting* dianggap masih sangat rawan terhadap gangguan dari pihak – pihak yang mempunyai maksud tertentu.

2. RSA

Dalam penerapan *blind signature*, salah satu cara paling mudah adalah dengan menggunakan RSA. Algoritma RSA ini sendiri dianggap cukup aman karena pemfaktoran yang dilakukan untuk mendapatkan kunci privat akan mengalami kesulitan saat memfaktorkan suatu bilangan yang besar menjadi faktor – faktor prima.

[1] **Perumusan Algoritma RSA**
Pertama – tama kita akan membahas mengenai algoritma RSA yang biasa. Algoritma ini memiliki besaran – besaran sebagai berikut:

	Besaran	Sifat
1	p, q (bilangan prima)	rahasia
2	$n = p \cdot q$	tidak rahasia
3	$\Phi(n) = (p-1) \cdot (q-1)$	rahasia
4	e (kunci enkripsi)	tidak rahasia
5	d (kunci dekripsi)	rahasia
6	m (plainteks)	rahasia
7	c (chiperteks)	tidak rahasia

Tabel 1

Perumusan algoritma ini adalah sebagai berikut:

$$a^{\Phi(n)} \equiv 1 \pmod{n} \qquad (1)$$

Dengan syarat:
1. a relatif prima terhadap n
2. $\Phi(n) = n(1-1/p1)\ (1-1/p2)\ldots\ (1-1/pr)$ dengan catatan $p1,p2,\ldots,pr$ adalah faktor prima dari n

Berdasarkan sifat $a^k = b^k \pmod{n}$ untuk k bilangan

bulat ≥ 1

$$a^{k\Phi(n)} \equiv 1^k \pmod{n} \quad (2)$$

atau

$$a^{k\Phi(n)} \equiv 1 \pmod{n} \quad (3)$$

Bila a diganti dengan m, maka persamaan menjadi

$$m^{k\Phi(n)} \equiv 1 \pmod{n} \quad (4)$$

Berdasarkan sifat $ac \equiv bc \pmod{n}$, maka bila dikali dengan m akan menjadi

$$m^{k\Phi(n)+1} \equiv m \pmod{n} \quad (5)$$

Misalkan e dan d dipilih sedemikian sehingga

$$e \,.\, d \equiv 1 \pmod{\Phi(n)} \quad (6)$$

atau

$$e \,.\, d \equiv k\Phi(n) + 1 \quad (7)$$

Sulihkan persamaan (7) dan (5) menjadi

$$m^{e \,.\, d} \equiv m \pmod{n} \quad (8)$$

Persamaan (8) dapat ditulis kembali menjadi

$$(m^e)^d \equiv m \pmod{n} \quad (9)$$

Berdasarkan persamaan (9), maka enkripsi dan dekripsi dirumuskan sebagai berikut

$$Ee(m) = c \equiv m^e \bmod n \quad (10)$$

$$Dd(c) = m \equiv c^d \bmod n \quad (11)$$

Karena e . d = e . d . e, maka enkripsi diikuti dengan dekripsi ekivalen dengan dekripsi diikuti enkripsi

$$Dd(Ee(m)) = Ee(Dd(m)) \equiv m^d \bmod n \quad (12)$$

Oleh karena $m^d \bmod n \equiv (m + jn)^d \bmod n$ untuk sembarang bilangan bulat j, maka tiap plainteks m, $m + n$, $m + 2n$,..., menghasilkan cipherteks yang sama yang mengakibatkan transformasinya dari banyak ke satu. Agar transformasinya dari satu ke satu, maka m harus dibatasi dalam himpunan {1, 1, 2,..., n-1} sehingga enkripsi dan dekripsi tetap benar seperti pada persamaan (10) dan (11).

[1] **Algoritma Membangkitkan Pasangan Kunci**
Berikut merupakan langkah – langkahnya:
1. Pilih dua buah bilangan prima sembarang, p dan q
2. Hitung $n = p \,.\, q$ (sebaiknya $p \neq q$)
3. Hitung $\Phi(n) = (p\text{-}1)(q\text{-}1)$
4. Pilih kunci publik, e, yang relatif prima terhadap $\Phi(n)$
5. Bangkitkan kunci privat dengan menggunakan persamaan (6), yaitu $e \,.\, d \equiv 1 \pmod{\Phi(n)}$

maka:

$$d = {}^{1 + k\Phi(n)} / _{e} \quad (13)$$

Hasil dari algoritma di atas:
- Kunci publik adalah pasangan (e, n)
- Kunci privat adalah pasangan (d, n)

[1] **Algoritma Enkripsi Dekripsi**
Enkripsi:
1. Ambil kunci publik penerima pesan, e, dan modulus n
2. Nyatakan plainteks m menjadi blok – blok $m1$,$m2$,..., sedemikian sehingga setiap blok merepresentasikan nilai di dalam selang [0, n – 1]
3. Setiap blok mi dienkripsi menjadi blok ci dengan rumus

$$c_i = m_i^e \bmod n \quad (14)$$

Dekripsi:
1. Setiap blok cipherteks c_i didekripsi kembali menjadi blok m_i dengan rumus $m_i = c_i^d \bmod n$

3. BLIND SIGNATURE

Konsep dari *blind signature* dapat diilustrasikan dalam contoh kasus yang sering terjadi pada keseharian kita, yaitu pada saat melakukan suatu voting. Masalah akan muncul ketika si pemilih inigin memastikan bahwa suara yang ia berikan pada kertas suara adalah benar dan terhitung, tetapi tidak ingin kerahasiaan pilihannya diketahui oleh pihak lain.

Pada teknik tanda tangan secara konvensional, yang dapat dilakukan adalah dengan menggunakan amplop yang memiliki bagian tertentu yang terbuat dari karbon, sehingga si pemilih dapat menandatangani amplop tersebut pada bagian luar amplop dan akan meninggalkan tandatangan di dalamnya agar nantinya bagian dalam amplop tersebut dikirimkan kembali kepadanya. Yang dilakukan setelahnya adalah pihak yang berwenang akan menghitung suara yang sesuai dengan pilihan si pemilih kemudian membubuhi tanda tangan yang menunjukkan bahwa suara tersebut telah terhitung. Bagian dalam dari amplop ini yang berisi tanda tangan si pemilih akan dikirimkan kembali ke alamat si pemilih tanpa ada pihak lain mengetahui pilihan yang telah dilakukan si pemilih.

Pada teknik *digital signature*, secara garis besar ada 3 fungsi yang harus dimiliki:
1. Fungsi yang pertama misalnya adalah fungsi s' yang hanya diketahui oleh si pemilih dan fungsi lain s (invers dari fungsi s') yang diketahui oleh pihak lain. Dalam hal ini fungsi s dan s' ini harus dapat memenuhi hubungan

$$s(s'(x)) = x \quad (15)$$

dan juga fungsi *s* tidak boleh memberikan petunjuk apa pun yang berhubungan dengan fungsi *s'*

2. Fungsi pengubah *c* dan pasangan inversnya, yaitu fungsi *c'* yang memenuhi syarat

$$c'(s'(c(x))) = s'(x) \quad (16)$$

c(x) dan *s'* tidak boleh memberikan petunjuk apa pun yang berhubungan dengan nilai x

3. Fungsi *r* yang dapat mengecek secara redundan untuk melakukan pengecekan ulang terhadap tanda tangan

Protokol penggunaan fungsi – fungsi ini adalah sebagai berikut:

1. Pihak yang berwenang memberikan nilai acak pada *r*(x) dan form *c*(x) kemudian memberikan *c*(x) kepada si pemilih
2. Pemilih membubuhi tanda tangan pada *c*(x) dengan menggunakan fungsi *s'* dan mengembalikan nilai *s'*(*c*(x)) kepada pihak yang berwenang.
3. Pihak yang berwenang kemudian melakukan pembubuhan tanda tangan dengan menggunakan fungsi *c'* yang akan menghasilkan *c'*(*s'*(*c*(x))) = *s'*(x)
4. Keabsahan nantinya dapat dicek dengan menggunakan fungsi *r* dan kunci publiknya, *r*(*s*(*s'*(x)))

4. PENGGUNAAN DAN IMPLEMENTASI

Penggunaan

Dari fungsi dan protokol yang telah dibahas pada bab 3, kita dapat menentukan penggunaan fungsi dan protokol tersebut adalah sebagai berikut:

1. *Digital signature*, pengecekan dapat dilakukan dengan kunci privat *s*' untuk mengecek dokumen *s'*(x)
2. *Blind signature*, pemilih tidak dapat mengetahui hubungan antara dokumen yang ia tanda tangani *s'*(x) dengan dokumen yang sudah diubah *s'*(*c*(x))
3. Pihak yang berwenang hanya dapat membuat satu dokumen yang unik dari setiap dokumen yang telah dibubuhi tanda tangan oleh pemilih (*s'*(*c*(x)) tidak mungkin sama dengan *s'*(*c*(y)) jika $x \neq y$)

Implementasi

Implementasi dari *blind signature* dapat dilakukan dengan menggunakan dasar dari algoritma RSA. Untuk lebih jelasnya akan kami berikan contoh implementasi dari *blind signature*.

Besaran – besaran dalam algoritma ini adalah:

	Besaran	Keterangan
	m	plainteks
	m'	cipherteks
	e	kunci enkripsi
	d	kunci dekripsi
	N	dibentuk dari 2 bilangan prima yang bersifat rahasia
	r	bilangan acak yang relatif prima terhadap *N*

Tabel 2

Misalnya si pemilih membuat suatu dokumen *m* dengan bilangan acak *r*, dengan nilai *N* dan kunci publik *e* yang sudah ditentukan dengan algoritma RSA biasa:

$$m' \equiv mr^e \pmod N \quad (17)$$

Kemudian *m'* dikirin ke pihak yang berwenang. Karena nilai *r* merupakan bilangan acak, maka *m'* tidak akan membocorkan informasi apa pun tentang dokumen *m*. Setelah itu pihak yang berwenang akan mengassign *m'* menjadi:

$$s' \equiv (m')^d \pmod N \quad (18)$$

Nilai *s'* akan dikirim kembali kepada si pemilih yang dapat melakukan verifikasi pilihan dengan mencari nilai *s* dengan cara membuang *blind factor*:

$$s \equiv s' * r^{-1} \pmod N \quad (19)$$

Hal ini berlaku karena $r^{ed} = r$; oleh karena itu:

$$\begin{aligned} s &\equiv s' * r'^{-1} \pmod N \\ &\equiv (m')^d r^{-1} \pmod N \\ &\equiv m^d r^{ed} r^{-1} \pmod N \\ &\equiv m^d r r^{-1} \pmod N \\ &\equiv m^d \pmod N \end{aligned} \quad (20)$$

5. ANCAMAN TERHADAP ALGORITMA

Ancaman terhadap algoritma *blind signature* dapat muncul karena sebnarnya algoritma ini merupakan pengembangan dari RSA biasa. Hal ini menyebabkan serangan yang dapat dilakukan terhadap algoritma RSA dapat dijadikan dasar untuk menyerang algoritma *blind signature*.

Pada *blind signature* proses pembubuhan tanda tangan secara tidak langsung melakukan enkripsi dengan kunci privat. Serangan dapat dilancarkan jika si penyerang dapat melakukan proses kebalikan dari algoritma *digital signature* pada dokumen yang telah dibubuhi tanda tangan. Hal ini menyebabkan si penyerang akan mendapatkan plainteks *m*.

$$m'' = m'r^e \pmod n$$

$$= (m^e \pmod n * r^e) \pmod n$$

$$= (mr)^e \pmod n \qquad (21)$$

m' adalah dokumen yang telah dibubuhi tanda tangan. Ketika dokumen telah ditanda tangani oleh si pemilih, maka plainteks m dapat diambil dengan cara:

$$s' = m''^d \pmod n$$

$$= ((mr)^e) \pmod n))^d \pmod n$$

$$(mr)^{ed} \pmod n$$

$$m * r^{-1} \pmod n \qquad (22)$$

Algoritma *digital signature* memang tidak menandatangani secara langsung dokumen (melakukan fungsi *hash* terlebih dahulu) tetapi serangan masih tetap bisa dilakukan. Namun, serangan yang dilakukan tidak bisa secara langsung sehingga tingkat keamanan dari algoritma ini masih cukup baik.

6. KESIMPULAN

Blind signature merupakan algoritma dalam kriptografi yang merupakan pengembangan dari RSA biasa. Karena itu, algoritma ini juga mewarisi kelebihan dan kekurangan dari algoritma RSA.

Algoritma *blind signature* biasanya digunakan jika kita menemukan kasus di mana seseorang harus membuktikan bahwa sebuah dokumen memang otentik miliknya dan data di dalamnya akan digunakan oleh pihak yang berwenang, tetapi privasi pemilik dari dokumen itu harus dijaga. Contoh kasusnya adalah ketika melakukan voting di mana si pemilih harus memilih kemudian memastikan bahwa pilihannya tersebut telah dipergunakan sebagaimana mestinya tanpa harus membuka identitasnya. Yang ditekankan dalam kasus ini adalah pilihan harus dapat dilihat oleh pihak berwenang tetapi identitas si pemilih bisa tetap terjaga.

Ada beberapa kelebihan dari *blind signature* :

1. Pemilih dapat melakukan verifikasi pilihannya tanpa harus diketahui identitasnya
2. Pihak yang berwenang hanya dapat membuat 1 dokumen yang unik dari tanda tangan si pemilih sehingga menutup kemungkinan terjadinya 2 dokumen dari pemilih yang sama

Kelemahan dari *blind signature* adalah rentan terhadap serangan yang dapat dilakukan pada algoritma RSA biasa. Memang serangan tetap bisa dilakukan terhadap algoritma *blind signature* tetapi tidak bisa secara langsung karena algoritma ini juga membawa kelebihan algoritma RSA biasa yang menggunakan fungsi *hash* (fungsi yang melakukan enkripsi terlebih dahulu sebelum dilakukan pembubuhan tanda tangan)

DAFTAR REFERENSI

[1] Diktat Kuliah IF5054 Kriptografi, 2006, Ir. Rinaldi Munir, M.T.

http://www.elections.act.gov.au/Elecvote.html,

http://www.electronic-voting.org/

http://csdl2.computer.org/persagen/DLAbsToc.jsp?resourcePath=/dl/proceedings/&toc=comp/proceedings/aina/2005/2249/02/2249toc.xml&DOI=10.1109/AINA.2005.63

http://www.patentdebate.com/PATAPP/20070192607

http://www.ingentaconnect.com/content/els/01403664/2000/00000023/00000017/art00254

http://www.freshpatents.com/Electronic-voting-process-using-fair-blind-signatures-dt20070816ptan20070192607.php?type=description

http://linkinghub.elsevier.com/retrieve/pii/S0096300304008549