APLIKASI TEORI BILANGAN UNTUK AUTENTIKASI DOKUMEN

Mohamad Ray Rizaldy - 13505073

Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung, Jawa Barat
e-mail: if15073@students.if.itb.ac.id

ABSTRAK

Makalah ini akan membahas tentang pemanfaatan teori-teori bilangan bulat untuk autentikasi atau memeriksa keabsahan dokumen. Cara yang biasa dipakai orang untuk autentikasi adalah pembubuhan tanda tangan. Dalam dunia komputer diperkenalkan teknik yang sama yaitu, tanda tangan digital (*digital signature*). Teknik tanda tangan digital ini merupakan gabungan dari dua aplikasi teori bilangan bulat aritmatika modulo, yaitu fungsi *hash* dan kriptografi.

Kata kunci: teori bilangan, aritmatika modulo, tanda tangan digital, fungsi *hash*, kriptografi.

1. PENDAHULUAN

Teori bilangan bulat, khususnya aritmatika modulo memiliki peranan penting dalam banyak hal. Beberapa contoh penggunaan aritmatika modulo misalnya fungsi hash dan juga kriptografi.

Fungsi hash berguna dalam mentransformasikan masukan string menjadi string lain. Sedangkan kriptografi digunakan untuk penyandian pesan. Jika dua penggunaan teori bilangan tersebut digabungkan, hasilnya adalah suatu teknik baru yang disebut sebagai teknik tanda tangan digital. Teknik ini berguna untuk autentikasi atau memeriksa keabsahan dokumen.

Sejak dulu untuk memeriksa keabsahan sebuah dokumen cetak, orang-orang menggunakan tanda tangan. Kini setelah memasuki era tanpa kertas (*paperless era*), menggunakan cara yang mirip, diperkenalkanlah teknologi tanda tangan digital.

2. TEORI BILANGAN BULAT

Bilangan bulat adalah bilangan yang tidak memiliki pecahan desimal misalnya 4, 29, 0, dan -3. Selain bilangan bulat, terdapat juga bilangan riil. Kebalikan dari bilangan bulat, bilangan riil adalah bilangan yang memiliki titik desimal seperti, 4.00, 28.75, dan 0.004 [1].

Sebuah bilangan bulat memiliki sifat pembagian sebagai berikut:

1. Misalkan a dan b adalah dua buah bilangan bulat dengan syarat $a \neq 0$. Kita menyatakan bahwa a habis membagi b jika terdapat bilangan bulat c sedemikian sehingga $b = ac$.
2. Notasi: $a \mid b$ jika $b = ac$, $c \in Z$ dan $a \neq 0$. (Z = himpunan bilangan bulat).

2.1 Aritmatika Modulo

Aritmatika modulo merupakan salah satu dari teori bilangan bulat yang penting yang digunakan untuk perhitungan bilangan bulat. Operator yang digunakan dalam aritmatika adalah **mod**, atau fungsi modulo. Fungsi modulo menghasilkan sisa pembagian.

Adapun fungsi modulo didefinisikan sebagai berikut:

- Misalkan a adalah bilangan bulat dan m bilangan bulat > 0. Operasi a **mod** m memberikan sisa jika a dibagi dengan m.
- a **mod** m dibaca 'a modulo m'
- notasi a **mod** m = r sedemikian sehingga $a = mq + r$, dengan $0 \leq r < m$.
- m disebut modulus atau modulo, dan hasil modulo m terletak di dalam himpunan {0,1,2, ..., m-1}

Contoh dari fungsi modulo misalnya 23 mod 5 = 3, 27 mod 3 = 0.

Untuk dua buah bilangan a dan b yang berbeda, bisa saja memiliki sisa yang sama jika dibagi dengan bilangan positif m. Hal ini bisa disebut bahwa a dan b kongruen dalam modulo m, yang dilambangkan dengan :

$$a \equiv b \ (\mathbf{mod}\ m) \qquad (1)$$

Misalnya 38 mod 5 dan 13 mod 5, hasil dari dua operasi tersebut adalah 3. Maka dapat dikatakan $38 \equiv 13 \pmod 5$.

2.2 Fungsi Hash

Secara sederhana fungsi ditujukan untuk pengalamatan *record* di memori. Bentuk dari fungsi *hash* adalah sebagai berikut:

$$h(k) = k \bmod m \qquad (2)$$

dengan k adalah kunci bilangan bulat dan m adalah jumlah lokasi memori yang tersedia. Sedangkan hasilnya h(k) adalah lokasi memori untuk record dengan kunci k [1].

Contoh: m = 11 mempunyai sel-sel memori yang diberi indeks 0 sampai 10. Akan disimpan data record yang masing-masing mempunyai kunci 15, 558, 32, 132, 102, dan 5. Maka:

$h(15) = 15 \bmod 11 = 4$
$h(558) = 558 \bmod 11 = 8$
$h(32) = 32 \bmod 11 = 10$
$h(132) = 132 \bmod 11 = 0$
$h(102) = 102 \bmod 11 = 3$
$h(5) = 5 \bmod 11 = 5$

Tabel 1 Tabel penempatan *record* pada memori dengan fungsi *hash*

132			102	15	5			32		558
0	1	2	3	4	5	6	7	8	9	10

Fungsi hash sering juga disebut sebagai *cryptographic checksum* karena bisa digunakan untuk mentransformasi masukan string dengan panjang sembarang menjadi sebuah string lain dengan panjang tetap. Hasil dari transformasi tersebut biasanya berukuran lebih pendek dibanding string masukannya. Hasil transformasi ini disebut juga nilai *hash* atau *message digest*. Jika dituliskan dalam notasi matematis akan jadi seperti:

$$MD = H(M) \quad (3)$$

dengan MD adalah *message digest*, dan M adalah string masukan.

Oleh fungsi *hash* sebuah string yang berukuran apapun diubah menjadi *message digest* yang berukuran tetap (128-512 bit).

Adapun sifat-sifat yang dimiliki oleh fungsi *hash* adalah sebagai berikut :

- Fungsi H dapat diterapkan pada blok data yang berukuran berapa saja.
- Nilai *hash* yang dihasilkan memiliki panjang yang tetap.
- Untuk setiap h yang diberikan, tidak mungkin menemukan suatu x sedemikian sehingga H(x)=h. Fungsi H tidak dapat mengembalikan nilai *hash* menjadi masukan awal.
- Untuk setiap x yang diberikan, tidak mungkin mencari pasangan $x \neq y$ sedemikian sehingga H(x)=H(y).

Ada dua jenis fungsi *hash* yang sering digunakan hingga sekarang. Fungsi *hash* yang pertama adalah MD5. Fungsi MD5 dibuat oleh Ron Rivest pada tahun 1994. Nilai *hash* yang dihasilkan oleh MD5 berukuran 128 bit. Fungsi *hash* yang lain adalah SHA (*Secure Hash Algorithm*) dikembangkan oleh NIST sebagai spesifikasi SHS (*Secure Hash Standard*). SHA sendiri memiliki banyak varian. Salah satunya adalah varian SHA-1. Varian SHA-1 ini menghasilkan nilai *hash* yang berukuran 160 bit.

2.3 Kriptografi

Kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan. Penggunaan kriptografi memungkinkan kita untuk menyimpan suatu informasi penting dan rahasia lalu mengirimkannya melalui komunikasi yang mungkin saja tidak terjamin keamanannya. Dengan begitu informasi tersebut tidak dapat diketahui kecuali oleh penerima informasi yang dimaksud [2].

Sejarah kriptografi dimulai ketika Julius Caesar mengirimkan pesan pada para jendralnya. Karena Caesar tidak mempercayai pembawa pesan, dia mengganti tiap huruf a pada pesannya dengan d, b menjadi e dan seterusnya tiap huruf diganti dengan tiga huruf setelahnya. Teknik kriptografi ini disebut juga *caesar cipher*.

Proses mengubah pesan menjadi pesan acak yang tidak dapat dibaca disebut dengan proses enkripsi. Proses kebalikannya untuk mendapat pesan asli disebut dekripsi. Dalam notasi matematika, enkripsi dan dekripsi dituliskan sebagai berikut:

$$Ci = E(Pi) \quad (4)$$
$$Pi = D(Ci) \quad (5)$$

dimana Pi adalah elemen pesan sesungguhnya (karakter) dan Ci adalah cipher hasil enkripsi.

Pada teknik *caesar cipher*, fungsi enkripsi dekripsinya dituliskan sebagai berikut:

$$Ci = E(Pi) = (Pi + 3) \bmod 26 \quad (6)$$
$$Pi = D(Ci) = (Ci - 3) \bmod 26 \quad (7)$$

Teknik kriptografi dikembangkan menjadi kriptografi modern sejak ditemukannya komputer. Algoritma kriptografi modern berbasis bit. Setiap operasi enkripsi dilakukan terhadap bit-bit data.

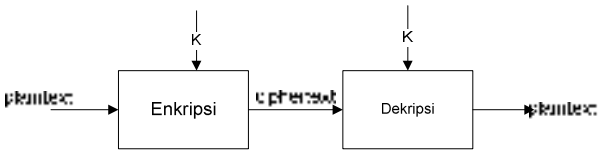
Terdapat dua jenis algoritma dalam kriptografi modern. Jenis ini dibedakan berdasarkan kunci yang digunakan dalam enkripsi dan dekripsi. Jenis pertama adalah kriptografi kunci simetri. Dan yang kedua adalah kriptografi kunci asimteri.

2.3.1 Kriptografi Kunci Simetri

Pada kriptografi kunci simetri, baik untuk mengenkripsi pesan maupun mendekripsi cipher digunakan kunci yang sama (simetrik). Dengan kunci yang bersifat simetrik tersebut, hal utama yang perlu dijaga adalah kerahasiaan kunci. Idealnya kunci yang digunakan memiliki panjang yang sama dengan panjang pesan aslinya.

Kelemahan dari kriptografi kunci simetrik adalah sulitnya mendistribusikan kunci. Untuk itu, sebuah kunci yang sudah digunakan tidak boleh digunakan. Hal ini dilakukan untuk menghindari pihak luar melakukan *eavesdropping* yang menyebabkan cipher akhirnya dapat dibongkar.

Jika digambarkan dalam bagan, kriptografi kunci simetrik ditunjukkan sebagai berikut:

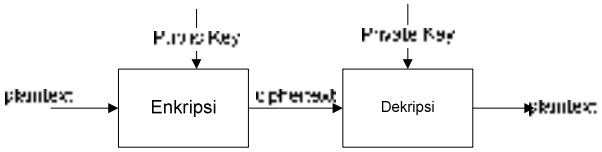


Gambar 1 Enkripsi & Dekripsi pada Kriptografi Kunci Simetri

2.3.2 Kriptografi kunci asimetri

Kriptografi kunci asimetri juga disebut sebagai kriptografi kunci publik. Pada kriptografi kunci asimetri, kunci yang digunakan untuk enkripsi dan dekripsi adalah kunci yang berbeda. Kunci yang digunakan untuk enkripsi bersifat publik disebut juga *public key*. Sedangkan kunci yang digunakan untuk dekripsi bersifat rahasia dan disebut *private key*.

Pada kriptografi jenis ini, pengirim pesan menggunakan kunci publik penerima untuk mengenkripsi pesan. Setelah pesan terenkripsi dan ditransmisi, penerima bisa mendekripsinya dengan kunci privat yang dimilikinya. Proses enkripsi dan dekripsi pada kriptografi kunci publik ditunjukan oleh gambar 2.



Gambar 2 Enkripsi & Dekripsi pada Kriptografi Kunci Publik

Ada beberapa algoritma yang dalam kriptografi asimetri. Salah satunya adalah algoritma RSA, yang akan dibahas dalam makalah ini.

Saat ini RSA bisa dikatakan sebagai algoritma kunci publik paling populer. Nama RSA sendiri diambil dari singkatan nama-nama penemu algoritmanya, yaitu Ron Rivest, Adi Shamir dan Leonard Adleman.

Dalam melakukan enkripsi dan dekripsi secara umum algoritma RSA memiliki besaran-besaran sebagai berikut:

- p dan q, bilangan prima rahasia
- n = p **.** q, tidak rahasia
- Φ(n) = (p – 1) **.** (q – 1), rahasia
- SK, kunci privat rahasia
- PK, kunci publik tidak rahasia.
- M, pesan yang dienkripsi.

Kunci publik PK dalam RSA merupakan hasil pembangkitan bilangan secara acak dari pencarian bilangan yang relatif prima terhadap Φ(n). Sedangkan kunci privat SK dibangkitkan dengan menggunakan persamaan:

$$PK.SK = 1 \left(mod\ \Phi(\mathrm{n})\right) \qquad (8)$$

Untuk membangkitkan pasangan kunci langkahnya adalah sebagai berikut:

1. Pilih dua bilangan prima, *a* dan *b* (rahasia).
2. Hitung $n = a\ b$. Besaran *n* tidak perlu dirahasiakan.
3. Hitung $m = (a - 1)(b - 1)$.
4. Pilih sebuah bilangan bulat untuk kunci publik, sebut namanya *e*, yang relatif prima terhadap *m*.
5. Hitung kunci dekripsi, *d*, melalui $d \equiv 1$ (**mod** *m*).

Selanjutnya untuk melakukan proses enkripsi dilakukan langkah berikut:

1. Nyatakan pesan menjadi blok-blok plainteks: $p_1, p_2, p_3, \ldots$ (harus dipenuhi persyaratan bahwa nilai p_i harus terletak dalam himpunan nilai $\{0, 1, 2, \ldots, n - 1\}$ untuk menjamin hasil perhitungan tidak berada di luar himpunan)
2. Hitung blok cipherteks c_i untuk blok plainteks p_i dengan persamaan:

$$C_i = P_i^e \ \mathbf{mod}\ n \qquad (9)$$

dalam hal ini, *e* adalah kunci publik.

Sedangkan untuk melakukan dekripsi dilakukan langkah sebaliknya yaitu dengan menghitung plaintext dari persamaan:

$$P_i = C_i^d \ \mathbf{mod}\ n \qquad (10)$$

dengan d adalah kunci privat penerima.

Kekuatan algoritma RSA ini terletak pada tingkat kesulitan dalam memfaktorkan bilangan non-prima menjadi faktor primanya, yang dalam hal ini adalah n = ab. Karenanya, nilai a dan b disarankan memiliki panjang lebih dari 100 digit.

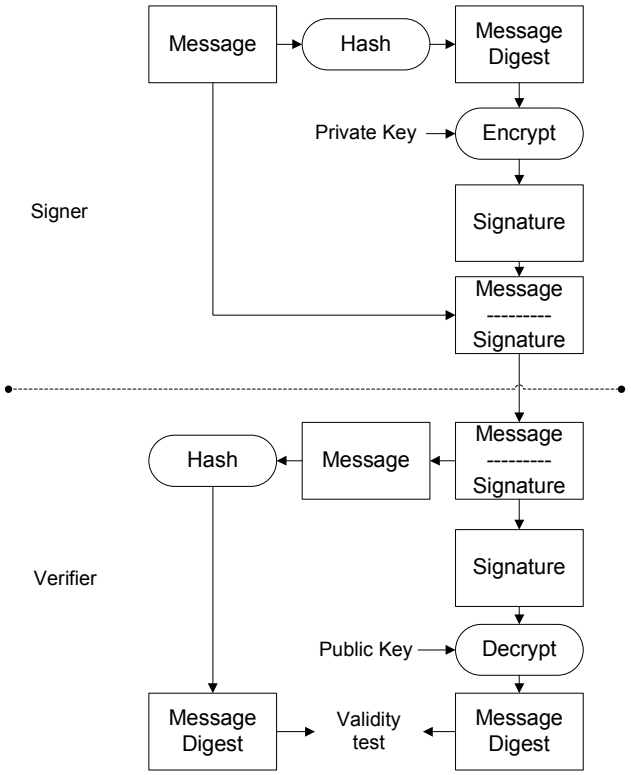
3. TANDA TANGAN DIGITAL

Sebagai sebuah alat pemeriksa keabsahan sebuah tanda tangan secara garis besar memiliki karakteristik sebagai berikut:

- Merupakan bukti yang otentik.
- Tidak dapat dilupakan.
- Tidak dapat dipindahtangankan.
- Tidak dapat disangkal.
- Dokumen yang telah ditandatangani tidak dapat berubah.

Fungsi tanda tangan pada dokumen kertas diterapkan untuk otentikasi pada data digital seperti pesan yang dikirim melalui saluran komunikasi dan dokumen elektronis yang disimpan di dalam memori komputer. Tanda tangan pada data digital ini dinamakan tanda tangan digital (*digital signature*).

Perlu ditekankan bahwa tanda tangan digital bukanlah tanda tangan manual (pada dokumen cetak) yang didigitasi dengan alat pemindai. Tanda tangan digital sebenarnya adalah suatu nilai kriptografis yang bergantung pada pesan dan pengirim pesan. Oleh karena itu tanda tangan digital seorang akan berbeda pada dua buah dokumen atau pesan yang berbeda.



Gambar 3 Skema Penandatanganan Digital

Pembubuhan tanda tangan digital pada sebuah dokumen, menggunakan *hash*. Dokumen yang hendak dikirim diubah terlebih dahulu menjadi bentuk yang ringkas yang disebut *message digest* MD.

Selanjutnya, *message digest* MD dienkripsi dengan algoritma kriptografi kunci-publik menggunakan kunci rahasia (SK) milik penandatangan atau pengirim dokumen. Hasil dari enkripsi inilah yang kita sebut sebagai tanda tangan digital (*digital signature)* S.

Dokumen M dan tanda tangan digital S kemudian dikirim melalui saluran komunikasi. Dalam hal ini, dokumen M telah ditandatangani oleh pengirim dengan tanda tangan S. Sebuah tanda tangan digital bisa dilekatkan (*embedded*) pada sebuah dokumen atau bisa juga disimpan di dokumen yang terpisah.

Setelah sampai di penerima, dokumen diverifikasi untuk dibuktikan keabsahannya. Adapun caranya adalah dengan mendekripsi tanda tangan digital S dengan kunci publik PK milik pengirim menjadi *message digest* MD kembali. Jika *message digest* hasil dekripsi cocok berarti dokumen yang diterima valid atau terbukti keabsahannya.

Dalam penandatanganan secara digital ini ada dua standar yang umum digunakan kedua standar tersebut adalah RSA (Rivest-Shamir-Adleman) dan DSA (Digital Signature Algorithm). Namun yang akan dibahas dalam makalah ini adalah penandatanganan digital dengan RSA.

3.1 Tanda tangan digital menggunakan RSA

Seperti dijelaskan sebelumnya bahwa teknik tanda tangan digital untuk autentikasi dokumen menggunakan fungsi *hash* dan kriptografi kunci publik atau asimetrik. Namun perlu diperhatikan bahwa terdapat perbedaan penggunaan kunci. Dalam enkripsi biasa pesan dienkripsi dengan kunci publik baru didekripsi dengan kunci privat. Namun pada praktek tanda tangan digital digunakan sebaliknya [4].

Adapun cara pembubuhan tangan digital dengan algoritma RSA dijelaskan sebagai berikut:

1. Pengirim menghitung nilai *hash* dari pesan *M* yang akan dikirim, misalkan nilai *hash* dari M adalah MD.
2. Pengirim mengenkripsi MD dengan kunci privatnya menggunakan persamaan enkripsi *RSA*.

Selanjutnya untuk verifikasi tanda tangan digital:

1. Penerima menghitung nilai *hash* dari pesan *M* yang akan dikirim, misalkan nilai *hash* dari *M* adalah MD'.
2. Penerima melakukan dekripsi terhadap tanda-tangan *S* dengan kunci publik si pengirim.

Penerima membandingkan MD dengan MD'. Jika MD = MD' maka dokumen beserta tanda-tangan digitalnya adalah otentik. Jika sebaliknya, berarti dokumen atau pengirimnya dinyatakan tidak valid.

4. KESIMPULAN

1. Teori bilangan bisa digunakan untuk membuat *hash* dan menyandikan pesan (kriptografi).
2. Aplikasi teori bilangan yaitu *hash* dan kriptografi dapat digabungkan dan menghasilkan teknik baru untuk autentikasi dokumen digital. Teknik ini disebut teknik tanda tangan digital (*digital signature*).
3. Dalam tanda tangan digital fungsi *hash* digunakan untuk menghasilkan *message digest* dari sebuah dokumen. Terdapat beberapa jenis fungsi *hash* yang bisa digunakan seperti MD5 dan SHA.
4. Kriptografi kunci publik digunakan untuk enkripsi *message digest* menjadi tanda tangan (*signature*). Sedangkan untuk menguji keabsahan dokumen signature didekripsi kembali menjadi message digest dan dicocokkan dengan hasil *hash* dokumen.

5. Salah satu algoritma yang dipakai untuk tanda tangan digital adalah algoritma RSA.
6. Pada tanda tangan digital dengan RSA penggunaan kunci publik dan privatnya berkebalikan dengan pada kriptografi biasa.

REFERENSI

[1] Munir, Rinaldi. 2006. Diktat Kuliah IF2153 – Matematika Diskrit. Departemen Teknik Informatika ITB.

[2] Munir, Rinaldi. 2004. Diktat Kuliah IF5054 – Kriptografi. Departemen Teknik Informatika ITB

[3] Wahyudi, Hedri. 2008. Catatan Kuliah Kriptografi I. http://hedriwahyudi.wordpress.com/2008/02/19/catatan-kriptografi/ diakses 20 Desember 2009.

[4] Rizaldy, Ray. 2009. Makalah Kuliah Kriptografi - Perbandingan Tanda Tangan Digital RSA dan DSA Serta Implementasinya untuk Antisipasi Pembajakan Perangkat Lunak.