Perbandingan Algoritma Tanda Tangan Digital Rabin dan Schnorr

Rachmansyah Budi Setiawan - 13507014
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
if17014@students.if.itb.ac.id

***Abstrak*—Informasi yang terkandung dalam suatu pesan atau dokumen merupakan hal yang penting. Namun bagaimana menjamin bahwa informasi yang terdapat di dalam suatu pesan atau dokumen masih asli sesuai dengan yang dimaksudkan oleh pembuatnya? Disinilah peran kriptorafi dengan layanan untuk menjamin integritas data. Integritas data dapat dijamin dengan pembubuhan tanda tangan digital, yang berfungsi layaknya tanda tangan biasa. Ada beberapa algoritma tanda tangan digital yang telah dikembangkan hingga saat ini. Dua di antaranya adalah algoritma Rabin dan Schnorr. Pada makalah ini, penulis mencoba membandingkan kedua algoritma tersebut.**

***Kata Kunci*—kriptografi, tanda tangan digital, Rabin, Schnorr**

I. PENDAHULUAN

Pada zaman sekarang, informasi merupakan hal yang berharga. Akan tetapi, arus pertukaran informasi sudah semakin sulit untuk dikendalikan disebabkan kebebasan dan keterbukaan, terutama di dunia internet. Oleh karena itu manusia menciptakan usaha untuk mengamankan pesan dan informasi yang terkandung di dalamnya. Salah satunya adalah dengan mempelajari dan mengembangkan kriptografi.

Kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan. Kriptografi timbul karena adanya usaha untuk mendapatkan informasi dari pesan oleh pihak – pihak yang sebenarnya tidak berhak. Pengamanan pesan dilakukan dengan menentukan sebuah sistem kriptografi yang terdiri dari algoritma kriptografi, plainteks, cipherteks, dan kunci.

Kriptografi menyediakan berbagai layanan yang mendukung pengamanan pesan. Layanan – layanan tersebut adalah:

1. Kerahasiaan (*confidentiality*)
   Layanan yang menjaga isi pesan dari siapapun yang tidak berhak membacanya.
2. Integritas data (*data integrity*)
   Layanan yang menjamin bahwa pesan masih asli/utuh atau belum pernah dimanipulasi selama pengiriman.
3. Otentikasi (*authentication*)
   Layanan yang mengidentifikasi kebenaran pihak-pihak yang berkomunikasi (*user authentication*) dan mengidentifikasi kebenaran sumber pesan (*data origin authentication*).
4. Nirpenyangkalan (*non-repudiation*)
   Layanan yang mencegah entitas yang berkomunikasi melakukan penyangkalan, yaitu pengirim pesan menyangkal melakukan pengiriman atau penerima pesan menyangkal telah menerima pesan.

Secara umum kriptografi dapat dibagi menjadi dua bagian, yaitu kriptografi kunci simetri dan kunci nirsimetri. Sesuai dengan namanya, pembeda dari kedua kategori kriptografi tersebut terletak pada kunci yang digunakan untuk proses enkripsi dan dekripsi. Kriptografi kunci simetri menggunakan kunci yang sama untuk melakukan proses enkripsi dan dekripsi, sementara itu kriptografi kunci nirsimetri menggunakan pasangan kunci yang berbeda untuk proses enkripsi dan untuk proses dekripsi.

Kriptografi kunci simetri dan nirsimetri memiliki kelebihan dan kekurangan masing – masing. Kriptografi kunci simetri memiliki kelebihan dalam waktu yang diperlukan untuk proses enkripsi dan dekripsi, ukuran kunci, serta kemudahan otentikasi pengirim pesan. Sementara itu, kelemahan kriptografi kunci simetri adalah kesulitan dalam pengiriman dan pengelolaan kunci. Sementara itu, kriptografi kunci nirsimetri memiliki kelebihan dan kekurangan yang berkebalikan dengan kekurangan dan kelebihan pada kriptografi kunci simetri.

Kriptografi kunci nirsimetri, atau sering juga disebut sebagai kriptografi kunci publik, memiliki banyak sekali manfaat dan penerapan. Salah satu di antaranya adalah untuk penggunaan tanda tangan digital atau *digital signature*. Tanda tangan digital ini juga memenuhi layanan kriptografi untuk integritas data, otentikasi, dan nirpenyangkalan.

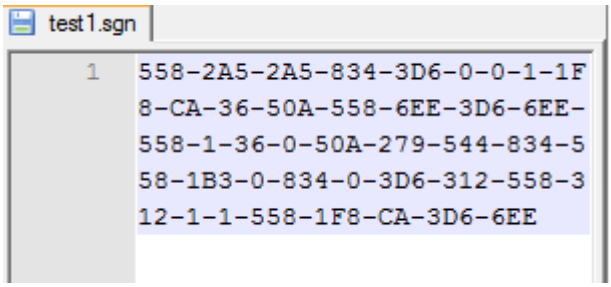
II. TANDA TANGAN DIGITAL

Tanda tangan yang biasa diberikan oleh seseorang pada suatu dokumen biasanya digunakan sebagai pendanda bahwa dokumen tersebut asli, sah, sudah diakui keberadaannya oleh pihak-pihak tertentu, dan semacamnya. Tanda tangan biasanya berasal dari nama

orang tersebut, inisal namanya, atau bentuk – bentuk yang menyulitkan sehingga sulit untuk ditiru.

Gambar 1. Tanda tangan dari seseorang bernama Jason Oliva

Tanda tangan digital memiliki fungsi yang sama dengan tanda tangan biasa. Namun, wujud dari tanda tangan ini bukanlah merupakan bentuk digital atau *image* dari tanda tangan biasa. Wujud tanda tangan digital umumnya adalah berupa rangkaian bit.



Gambar 2. Contoh wujud tanda tangan digital dalam representasi bilangan heksadesimal

Tanda tangan digital memiliki beberapa sifat, di antaranya adalah:

- Otentik, tak bisa, sulit ditulis atau ditiru oleh orang lain. Hal ini juga menjadi alasan tanda tangan digital dapat mencegah penyangkalan sang pembuat pesan bahwa ia tidak pernah menandatangani pesan tersebut.
- Hanya sah untuk dokumen atau pesan itu saja. Berlaku juga untuk salinannya yang sama persis.
- Dapat diperiksa dengan mudah, termasuk oleh pihak-pihak yang belum pernah bertemu langsung dengan penandatangan.

Penandatanganan suatu dokumen dalam bentuk digital dapat dilakukan dengan mengenkripsi dokumen tersebut atau menggunakan fungsi *hash* dan kriptografi kunci publik. Metode yang pertama relatif lebih mudah untuk dilakukan, namun metode ini tidak menyediakan mekanisme untuk anti-penyangkalan. Mekanisme tambahan harus dibuat untuk menangani anti-penyangkalan, misalnya dengan bantuan dari pihak ketiga. Sementara itu, metode yang kedua lebih umum digunakan karena tidak perlu menggunakan jasa dari pihak ketiga.

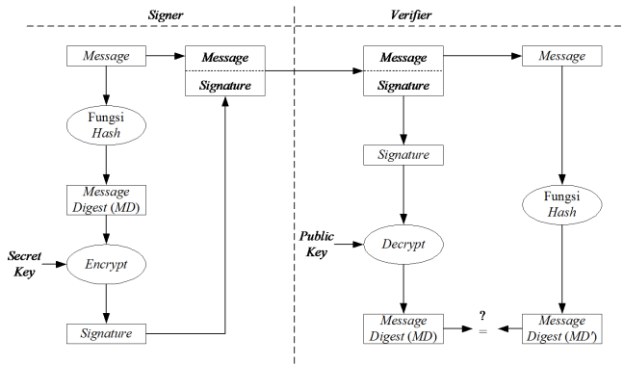
Penggunaan kriptografi kunci publik untuk tanda tangan digital agak berbeda dengan proses enkripsi-dekripsi biasa. Pada proses enkripsi biasa (hanya untuk menyembunyikan isi pesan/dokumen), enkripsi dilakukan dengan menggunakan kunci publik penerima dan dekripsi dilakukan dengan menggunakan kunci privat penerima. Sementara itu, untuk tanda tangan digital, enkripsi dilakukan dengan kunci privat pengirim dan dekripsi dilakukan dengan kunci publik pengirim. Dengan begitu, penerima dapat melakukan otentikasi pengirim pesan. Prosesnya kira – kira sebagai berikut:

$S = E_{SK}(M)$

$M = D_{PK}(S)$

Dengan SK = *secret key* atau kunci privat pengirim, PK = *publik key* atau kunci publik pengirim, E = fungsi enkripsi, D = fungsi dekripsi, M = pesan, dan S = tanda tangan digital.

Cara yang telah dibahas sebelumnya dapat memberikan dua fungsi, kerahasiaan dan kemudahan otentikasi pesan. Namun terkadang tanda tangan digital hanya diperlukan untuk mempermudah otentikasi pesan saja. Untuk kasus ini, dapat digunakan kombinasi fungsi *hash* dan kriptografi kunci publik. Caranya adalah dengan mendapatkan sebuah hasil pengolahan pesan dengan memberi masukan pesan ke dalam fungsi *hash* (sering disebut sebagai *Message Digest* atau disingkat MD), kemudian MD inilah yang akan dienkripsi dengan algoritma kriptografi kunci publik sehingga menghasilkan tanda tangan digital. Sementara itu, untuk proses verifikasi, yang perlu dilakukan adalah memdekripsi tanda tangan digital, kemudian membandingkan hasilnya dengan MD pada pesan yang memiliki tanda tangan digital tersebut.



Gambar 3. Skema proses penandatangan dan verifikasi menggunakan fungsi *hash* dan algoritma kriptografi kunci publik

Tanda tangan digital sudah memiliki standar. Standar yang dimaksud adalah *Digital Signature Standard* atau DSS. DSS terdiri atas dua komponen:

1. Algoritma tanda tangan digital (*Digital Signature Algorithm*/DSA)
2. Fungsi *hash* standar (*Secure Hash Algorithm*)

Dari standar tersebut, banyak tercipta algoritma tanda tangan digital. Ada beberapa algoritma tanda tangan digital yang sudah dikenal, di antaranya adalah algoritma Rabin dan algoritma Schnorr.

III. ALGORITMA RABIN

Algroritma Rabin mengandalkan suatu fungsi *hash* yang memetakan bit – bit dengan panjang sembarang menjadi sepanjang k. Dengan kata lain, fungsi *hash* yang digunakan, misalnya H, harus bisa memenuhi

H: $\{0,1\}^* \rightarrow \{0,1\}^k$

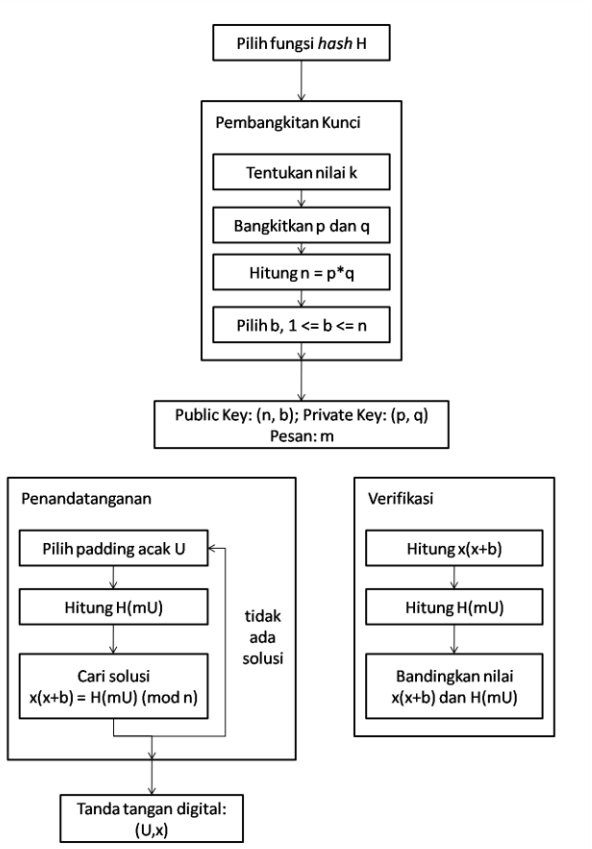
Selain itu, H juga diharapkan sebagai sebuah *random oracle*. *Random oracle* adalah fungsi matematika yang mampu memetakan semua *query*/permintaan yang mungkin menjadi respon acak dari domain keluaran fungsi tersebut, kecuali untuk beberapa *query* tertentu.

Algoritma Rabin sendiri dapat dibagi menjadi tiga bagian. Ketiga bagian tersebut adalah pembangkitan kunci, penandatanganan, dan verifikasi.

a. Pembangkitan kunci
   Langkah – langkahnya:
   - Pilih bilangan prima p dan q, masing – masing berukuran k/2 bit
   - Hitung n = pq
   - Pilih bilangan acak b di antara 1 sampai n.
   - Kunci publiknya adalah (n,b) dan kunci privatnya adalah (p,q)

b. Penandatanganan
   Langkah – langkahnya:
   - Penandatangan memilih padding secara acak (misalnya U)
   - Hitung H(mU) dengan m adalah pesan yang hendak ditandatangani
   - Cari solusi dari x(x+b) = H(mU) (mod n)
   - Jika tidak ada solusi, pilih U baru secara acak dan ulangi sampai ditemukan solusi untuk mendapatkan nilai x
   - Tanda tangan digitalnya adalah pasangan (U,x)

c. Verifikasi
   Langkah – langkahnya:
   - Hitung x(x+b) dan H(mU)
   - Jika nilai dari x(x+b) sama dengan H(mU), maka pesan m dengan tanda tangan digital (U,x) masih valid dan otentik, belum mengalami modifikasi sejak dibuat oleh pembuat aslinya.



Gambar 4. Diagram implementasi algoritma tanda tangan digital Rabin

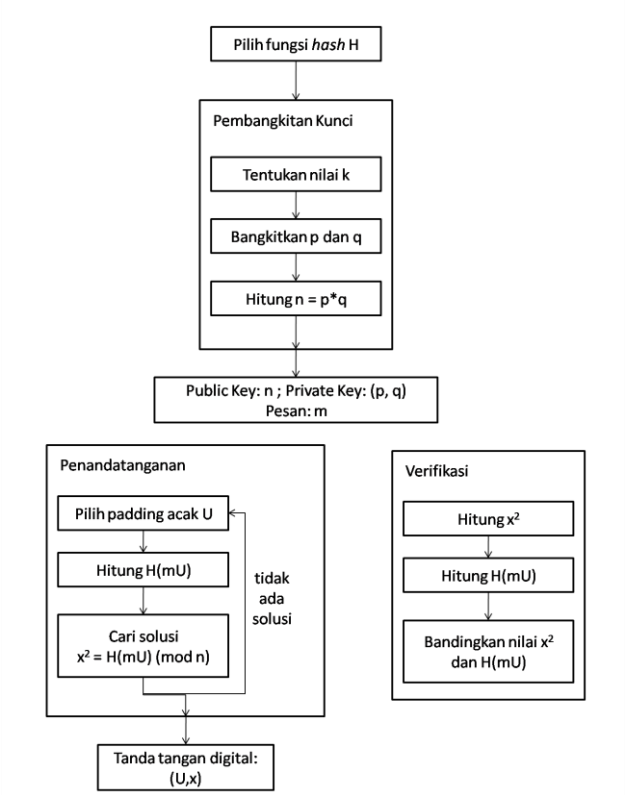
Cara di atas merupakan algoritma asli yang didefinisikan oleh pembuatnya, Michael O. Rabin. Sekarang algoritma Rabin juga seringkali direpresentasikan dengan cara yang lebih sederhana. Simplifikasi terletak pada tidak perlu lagi memilih bilangan acak b. Secara lengkap, langkah – langkah untuk ketiga bagian algoritma Rabin yang telah disimplifikasi adalah sebagai berikut:

a. Pembangkitan kunci
   Langkah – langkahnya:
   - Pilih bilangan prima p dan q, masing – masing berukuran k/2 bit
   - Hitung n = pq
   - Kunci publiknya adalah n dan kunci privatnya adalah (p,q)

b. Penandatanganan
   Langkah – langkahnya:
   - Penandatangan memilih padding secara acak (misalnya U)
   - Hitung H(mU) dengan m adalah pesan yang hendak ditandatangani
   - Ulangi langkah pertama jika H(mU) bukan square modulo n
   - Cari solusi dari x^2 = H(mU) (mod n)
   - Tanda tangan digitalnya adalah pasangan (U,x)

c. Verifikasi
   Langkah – langkahnya:

- Hitung x^2 dan H(mU)
- Jika nilai dari x^2 sama dengan H(mU), maka pesan m dengan tanda tangan digital (U,x) masih valid dan otentik, belum mengalami modifikasi sejak dibuat oleh pembuat aslinya.



Gambar 5. Diagram implementasi algoritma tanda tangan digital Rabin yang telah disederhanakan

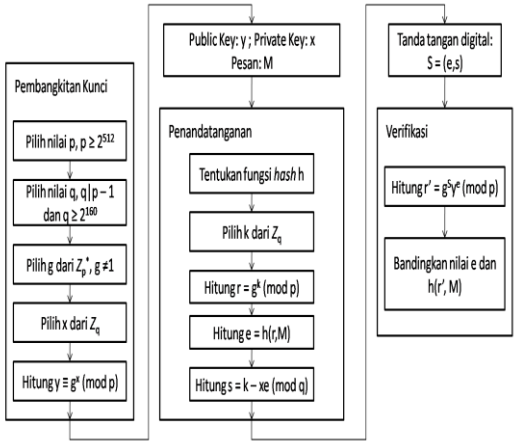
IV. ALGORITMA SCHNORR

Algoritma tanda tangan digital Schnorr memanfaatkan kesulitan beberapa permasalahan logaritma diskrit untuk dipecahkan sebagai dasar dari kemanannya. Seperti algoritma Rabin yang telah dijelaskan sebelumnya, algoritma Schnorr juga dibagi menjadi tiga bagian: pembangkitan kunci, penandatanganan, dan verifikasi.

a. Pembangkitan kunci
   Langkah – langkah yang dilakukan untuk membangkitkan kunci adalah sebagai berikut:
   - Pilih bilangan prima p, dengan syarat $p \geq 2^{512}$
   - Pilih bilangan prima q, dengan syarat q|p-1 dan $q \geq 2^{160}$
   - Pilih g dari Z_p^* dan $g \neq 1$. Z_p^* adalah sekumpulan bilangan yang kongruen dan relatif prima dengan p (atau *multiplicative group of integers modulo p*) untuk beberapa bilangan prima p.
   - Pilih x dari Z_q
   - Hitung $y \equiv g^x \pmod p$
   - Kunci publiknya adalah y dan kunci privatnya adalah x.

b. Penandatanganan
   Langkah – langkah untuk penandatanganan suatu pesan atau dokumen adalah sebagai berikut:
   - Tentukan fungsi *hash* yang akan digunakan (misalnya h)
   - Pilih k secara acak dari Z_q
   - Hitung $r = g^k \pmod p$
   - Hitung e = h(r, M) dengan M adalah pesan
   - Hitung $s = k - xe \pmod q$
   - Tanda tangannya adalah S = (e, s)

c. Verifikasi
   Untuk melakukan verifikasi, maka perlu dihitung $r' = g^S y^e \pmod p$
   Tanda tangan valid jika e = h(r', M).



Gambar 6. Diagram implementasi algoritma Schnorr

Algoritma Schnorr juga dapat direpresentasikan dengan cara lain. Cara lain ini memerlukan semua pengguna skema algoritma ini untuk menyetujui suatu grup G dengan pembangkit g dari *prime order* q yang memiliki permasalah logaritma diskrit yang sulit dipecahkan. Biasanya yang digunakan adalah *Schnorr group*. *Schnorr group* adalah kumpulan bilangan prima berorde besar yang merupakan *subgroup* dari Z_p^*. Selain itu, fungsi hash yang digunakan (misalnya H) juga sudah harus disetujui bersama. Setelah itu, langkah – langkah yang dilakukan mirip dengan penjelasan sebelumnya, namun dengan beberapa modifikasi.

a. Pembangkitan kunci
   Langkah – langkah untuk pembangkitan kuncinya adalah:
   - Pilih bilangan x sebagai kunci privat dengan syarat $0 < x < q$
   - Kunci publiknya sendiri adalah y dimana $y = g^x$

b. Penandatanganan
   Untuk melakukan penandatanganan pada pesan M, langkah – langkah yang harus dilakukan adalah:
   - Pilih bilangan acak k yang memenuhi persyaratan $0 < k < q$
   - Hitung $r = g^k$
   - Kemudian hitung e = H(M||r)

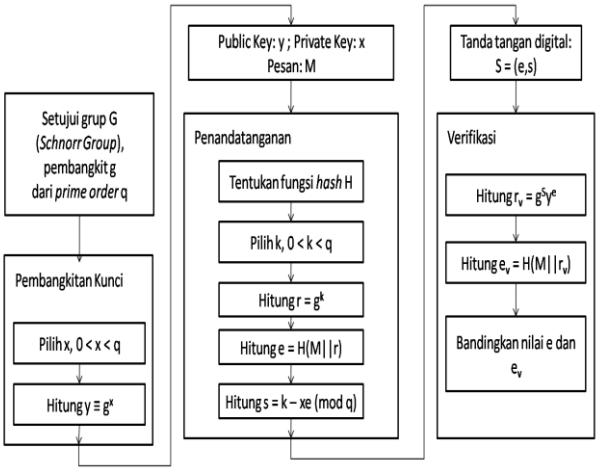
- Terakhir hitung s = (k – xe) mod q
- Tanda tangan digital yang dihasilkan ialah pasangan (e,s).

Sebagai catatan, nilai e seharusnya terletak di antara 0 dan q ($0 \leq e < q$).

c. Verifikasi

Untuk melakukan verifikasi terhadap pesan M dan pasangan tanda tangan digital (e,s), langkah – langkah yang harus dilakukan adalah:

- Hitung $r_v = g^s y^e$
- Kemudian hitung $e_v = H(M||r_v)$
- Jika $e_v = e$, maka tanda tangan digital (e,s) valid untuk pesan M. Dengan kata lain, jika diasumsikan (e,s) benar – benar tanda tangan digital untuk pesan M, maka pesan M valid dan masih otentik, sama dengan pesan asli yang dibuat oleh pembuatnya.



Gambar 7. Diagram implementasi algoritma Schnorr dengan menggunakan *Schnorr group*

Perlu diingat juga bahwa dengan cara yang kedua ini, komponen – komponen yang bersifat publik adalah G, g, q, y, s, e, dan r. Sementara itu, komponen – komponen yang bersifat privat adalah k dan x.

V. PERBANDINGAN

Karakteristik yang berbeda dari algoritma Rabin dan Schnorr dapat dilihat dari basis keamanan kedua algoritma tersebut. Algoritma Rabin mengandalkan keamanan pada fungsi *hash* yang berupa *random oracle* agar permasalahan pemfaktoran integer menjadi mustahil untuk diselesaikan. Di sisi lain, algoritma Schnorr mengandalkan kesulitan pemecahan algoritma yang serupa dengan pemecahan beberapa permasalahan logaritmadiskrit.

Dilihat dari sisi pembangkitan kunci, baik algoritma Rabin maupun algoritma Schnorr memerlukan pembangkitan dua bilangan prima acak yaitu p dan q. Namun pada algoritma Rabin besar p dan q dapat disesuaikan sesuai dengan nilai k atau ukuran keluaran dari fungsi *hash* yang dipilih. Sementara itu pada algoritma Schnorr diperlukan nilai p dan q yang cukup besar sehingga dapat membuat komputasi menjadi lebih kompleks.

Pada bagian penandatanganan, terlihat bahwa kedua algoritma masih mengandalkan aritmatika modulus. Pada algoritma Rabin, penandatanganan melibatkan penggunaan *padding* acak U. Kesalahan pemilihan padding dapat menyebabkan algoritma berputar pada kalang untuk mencari solusi bagi nilai x. Pada algoritma Schnorr, pada proses untuk mendapatkan nilai r, melibatkan operasi perpangkatan dengan orde k, yaitu panjang bit keluaran yang dipilih untuk fungsi hash. Perhitungan perpangkatan dengan orde sebesar ini memerlukan kemampuan ataupun waktu komputasi yang cukup signifikan.

Pada bagian verifikasi, algoritma Rabin terlihat unggul karena kecepatannya melakukan verifikasi. Verifikasi dilakukan dengan melakukan perbandingan antara dua bilangan. Kompleksitas tahap verifikasinya adalah antara perpangkatan berorde dua atau kompleksitas fungsi *hash* yang dipilih. Sementara itu, pada algoritma Schnorr, kompleksitasnya terlihat lebih tinggi. Hal ini dikarenakan seperti pada tahap penandatanganan, pada tahap verifikasi di algoritmaSchorr melibatkan perpangkatan dengan orde S dan e, yang kemungkinan besar lebih besar dari 2. Hal ini berarti proses verifikasi pada algoritma Schnorr melibatkan kemampuan atau waktu komputasi yang lebih besar.

Berikutnya perbandingan akan dilakukan berdasarkan kelebihan secara umum dari kedua algoritma. Dari sisi kelebihan, kelebihan dari algoritma Rabin adalah sulit diserang dengan serangan *existential forgery*. *Existential forgery* sendiri merupakan serangan dengan mencoba membuat tanda tangan digital yang valid untuk sebuah pesan tanpa mempedulikan arti dari pesan tersebut. Selain itu, algoritma Rabin juga memiliki kecepatan dalam hal verifikasi. Sementara itu, kelebihan dari algoritma Schnorr terletak pada kesulitan untuk memecahkan beberapa permasalahan logaritma diskrit.

Kekurangan umum dari kedua algoritma adalah keduanya masih menggunakan aritmatika modulus untuk penandatanganan dan verifikasinya. Jenis aritmatika ini lebih memakan waktu komputasi dibandingkan aritmatika linear yang non-modulo. Sementara itu menyimpulkan kekurangan masing – masing algoritma, kekurangan algoritma Rabin adalah dikhawatirkan pemilihan *padding* yang tidak tepat akan menyebabkan proses berputar pada kalang untuk mencari solusi. Sementara itu kekurangan algoritma Schnorr terletak pada pentingnya pemilihan fungsi *hash*, karena fungsi *hash* yang digunakan harus memenuhi beberapa persyaratan untuk menjamin keamanan dari *collision attack*.

VI. KESIMPULAN

Dari perbandingan algoritma Rabin dan algoritma Schnorr, dapat dilihat perbedaan karakteristik antara keduanya. Kompleksitas algoritma Rabin berbasiskan pada pemilihan fungsi *hash* yang diharapkan berupa *random oracle* sehingga sulit untuk melakukan pemfaktoran integer. Sementara itu kompleksitas algoritma Schnorr berbasiskan pada kesulitan untuk memecahkan beberapa persoalan logaritma diskrit. Secara umum, algoritma Schnorr lebih kompleks dibandingkan algoritma Rabin karena melibatkan perhitungan dalam pangkat yang besar. Hal ini bisa menjadi kelebihan sekaligus kekurangan. Kelebihan karena dapat mempersulit serangan terhadap algoritma Schnorr sendiri namun merupakan kelemahan juga karena dapat memakan *resource* untuk melakukan komputasi.

Untuk ke depannya, diharapkan ada algoritma tanda tangan digital lain yang dapat mengatasi kekurangan – kekurangan algoritma tanda tangan digital yang ada sekarang dan juga mampu memiliki kelebihan – kelebihan algoritma tanda tangan digital yang ada sekarang.

REFERENSI

[1] Munir, Rinaldi. *Kriptografi*. Institut Teknologi Bandung, 2006.
[2] http://dsns.csie.nctu.edu.tw/course/intro-security/2005/slides/07-Digital%20Signature%20Standards.pdf
[3] http://math.boisestate.edu/~liljanab/BOISECRYPTFall09/Sarang.pdf
[4] http://www.iacr.org/archive/eurocrypt2000/1807/18070091-new.pdf
[5] http://www.jscoron.fr/cours/mics3crypto/tp7-david.pdf
[6] http://userweb.cs.utexas.edu/~rashid/395Tcrypt/6_2.pdf
[7] http://go-kerja.com/tanda-tangan-digital/
[8] http://publications.csail.mit.edu/lcs/pubs/pdf/MIT-LCS-TR-212.pdf
[9] http://yourgodesign.com/images/signature-jpeg.jpg

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 16 Mei 2010

Rachmansyah Budi Setiawan - 13507014