Perbandingan Algoritma RSA dan Algoritma berbasis Zero Knowledge untuk Autentikasi pada SmartCard

Ahmad Zamakhsyari Sidiq [1)]
1) Jurusan Teknik Informatika ITB, Jl. Ganesha 10, email: if14053@students.if.itb.ac.id

Abstract – *Smart card disebut juga* tamper resistant security devices*, adalah suatu teknologi chip VLSI yang berfungsi bukan hanya untuk menyimpan data tetapi dapat memproses suatu informasi dan mengontrol secara internal suatu algoritma kriptografi sehingga cocok digunakan untuk pengecekan identitas dan mengembangkan suatu sistim keamanan secara logika dan elektronik. Identifikasi terhadap pemegang kartu pintar merupakan suatu hal yang paling penting untuk meyakinkan seseorang mempunyai hak dalam pemakaian kartu tersebut. RSA adalah algoritma kunci public yang paling banyak digunakan. Algoritma RSA dibuat oleh 3 orang peneliti dari MIT, yakni Rivest, Shamir, dan Adleman.*
Keamanan algoritma RSA terletak pada sulitnya memfaktorkan bilangan yang besar menjadi bilangan-bilangan prima. Sedangkan Zero Knowledge adalah sebuah metode interaktif yang digunakan satu pihak untuk membuktikan kebenaran suatu pernyataan ke pihak yang lain tanpa memberikan suatu informasi selain keabsahan dari pernyataan itu sendiri.
Pada makalah ini penulis akan mencoba membandingkan penerapan dua algoritma dalam autentifikasi pada smart card, yaitu algoritma RSA dan Zero Knowledge.

Kata Kunci*: Zero Knowledge, RSA, Smartcard autentification, VLSI*

1. PENDAHULUAN

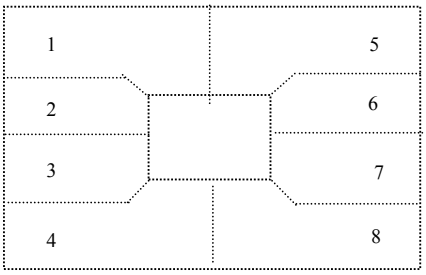
Very-large-scale integration (VLSI) adalah suatu teknologi yang menyatukan ribuan sirkuit berbasis transistor kedalam sebuah chip. VLSI mulai dikembangkan sekitar tahun 1970 ketika semikonduktor kompleks dan teknologi komunikasi tengah berkembang. Belakangan ini jumlah transistor yang digabungkan dalam satu chip mencapai angka ratusan juta transistor.

Smart Card atau Tamper Resistant Security Devices adalah suatu Kartu kredit modern dengan pengamanan pribadi yang merupakan versi kartu kredit baru pengembangan dari kartu kredit lama jenis magnetik yang biasa dikenal dengan Magnetic Strip Card. Kartu kredit modern ini memiliki suatu rangkaian pengendali mikro (*microcontroller*) berupa rangkaian terintegrasi (*Integrated Circuit*) yang dapat diprogram untuk menjalankan suatu aplikasi tertentu, misalnya mengontrol secara internal suatu algoritma kriptografi sehingga cocok digunakan untuk pengecekan identitas dan mengembangkan suatu sistim keamanan secara logika dan elektronik.

Smart card pada dasarnya mempunyai tiga elemen, yaitu :

- *processing power*,
- *data storage element;* dan
- *input/output data*.

Processing power akan didukung oleh *chip* mikroprosesor, *storage element* oleh *chip* memori, dan I/O data melalui kontak metal yang terdapat dalam lapisan atas kartu. Hal tersebut berlangsung dengan cara memasukkan kartu ke dalam slot unit pembaca/menulis (read/write) sehingga komunikasi data akan mengalir antara kartu dengan unit pembacanya.



Ket:

1 = Vcc (5V)	5 = Gnd
2 = R/W	6 = Vpp (12V)
3 = Clock	7 = I/O
4 = Reset	8 = Fuse

Gambar 1. Skema Kartu Pintar (Smart Card) dengan standard ISO

Aplikasi kartu pintar telah banyak digunakan pada saat ini, namun ada hal yang perlu diperhatikan dengan berkembang pesatnya penggunaan kartu pintar ini, itu adalah masalah keamanan. Proses identifikasi atau otentifikasi sangat diperlukan agar

informasi yang ada dalam kartu tidak bisa diubah atau dipakai oleh pihak yang tidak berhak. Sehingga dalam makalah ini, akan dibandingkan beberapa metode untuk proses otentifikasi kartu pintar (smart card), beserta keuntungan dan kelebihannya.

2. DASAR TEORI

2.1. RSA

RSA merupakan salah satu algoritma kriptografi kunci publik. Algoritma ini juga merupakan algoritma pertama yang dikenal cocok untuk tanda tangan digital. RSA dipublikasikan pada tahun 1977 oleh Ron Rivest, Adi Shamir, dan Leonard Adleman di MIT. Keamanan algoritma RSA terletak pada sulitnya memfaktorkan bilangan yang besar menjadi faktor-faktor prima. Pemfaktoran dilakukan untuk memperoleh kunci privat. Selama pemfaktoran bilangan besar menjadi faktor-faktor prima belum ditemukan algoritma yang mangkus, maka selama itu pula keamanan algoritma RSA tetap terjamin.[1]

Besaran-besaran pada algoritma RSA, yakni:

- p dan q bilangan prima (rahasia)
- $n = p \cdot q$ (tidak rahasia)
- $\Phi(n) = (p - 1)(q - 1)$ (rahasia)
- e (kunci enkripsi) (tidak rahasia)
- d (kunci dekripsi) (rahasia)
- m (plainteks) (rahasia)
- c (chiperteks) (tidak rahasia)

Algoritma RSA didasarkan pada teorema Euler yang menyatakan bahwa
$a^{\Phi(n)} \equiv 1 \pmod n$
dengan syarat:

1. a harus relatif prima terhadap n
2. $\Phi(n) = n (1 - 1/p_1) (1 - 1/p_2) \ldots (1 - 1/p_r)$, yang dalam hal ini $p_1, p2, \ldots, pr$ adalah faktor prima dari n.

Algoritma RSA melibatkan kunci publik dan kunci privat. Kunci publik boleh diketahui oleh siapa saja dan digunakan untuk mengenkripsi pesan. Pesan yang telah dienkripsi dengan kunci publik hanya bisa didekripsi dengan menggunakan kunci privat yang bersesuaian. Kunci untuk algoritma RSA dibangkitkan dengan cara sebagai berikut:

1. Pilih dua bilangan prima, a dan b (rahasia)
2. Hitung $n = a\ b$. Besaran n tidak perlu dirahasiakan.
3. Hitung $\phi(n) = (a - 1)(b - 1)$.
4. Pilih sebuah bilangan bulat untuk kunci publik, e, yang relatif prima terhadap $\phi(n)$.
5. Hitung kunci dekripsi, d, melalui $ed \equiv 1 \pmod m$ atau $d \equiv e^{-1} \bmod (\phi(n))$

Hasil dari algoritma di atas:

- Kunci publik adalah pasangan (e, n)
- Kunci privat adalah pasangan (d, n)

Proses enkripsi algoritma RSA adalah sebagai berikut:

- Nyatakan pesan menjadi blok-blok plainteks: $m_1, m_2, m_3, \ldots$ (harus dipenuhi persyaratan bahwa nilai m_i harus terletak dalam himpunan nilai 0, 1, 2, …, n-1 untuk menjamin hasil perhitungan tidak berada di luar himpunan)
- Hitung blok cipherteks c_i untuk blok plainteks p_i dengan persamaan $c_i = m_i^e$ **mod** n yang dalam hal ini, e adalah kunci publik.

Sedangkan proses dekripsi algoritma RSA adalah sebagai berikut:

- Proses dekripsi dilakukan dengan menggunakan persamaan $m_i = c_i^d$ **mod** n, yang dalam hal ini, d adalah kunci privat.

2.2 Zero-Knowledge

Zero-Knowledge adalah suatu protokol yang memungkinkan identifikasi, pertukaran kunci dan operasi-operasi kriptografi dasar lainnya terimplementasikan tanpa membocorkan suatu informasi rahasia dalam "percakapan"nya. Zero-knowledge membutuhkan proses komputasi yang lebih kecil jika dibandingkan dengan protocol kunci publik.[2]

Dalam protokol Zero-Knowledge, ada beberapa pihak yang terlibat yaitu:

- **Prover**, pihak yang memiliki informasi yang ingin dibuktikan kepada Verifier.
- **Verifier**, mengajukan beberapa pertanyaan kepada Prover, untuk mengetahui, apakah Prover benar-benar mengetahui secret atau tidak.
- **Eavesdropper** adalah pihak yang mendengarkan percakapan antara Prover dan Verifier. Protokol Zero-Knowledge yang baik akan memastikan bahwa pihak ketiga yang mendengarkan percakapan tidak akan mendapatkan pengetahuan apapun.
- **Malice**, mendengarkan lalu lintas protokol dan menambahkan, memodifikasi atau bahkan menghancurkan pesan. Protokol yang baik harus tahan terhadap gangguan dari Malice.

Beberapa terminologi yang digunakan dalam Zero-knowledge antara lain:

- *Secret*, suatu potongan informasi, bisa password, private key untuk public key.

Dalam Zero-knowledge, prover bisa meyakinkan verifier tanpa harus memberitahukan secret itu sendiri.

- *Accreditation*, adalah pembangunan kepercayaan dalam tiap iterasi dalam protokol.
- *Problem*, masalah yang ditawarkan oleh prover kepada verifier.
- *Cut-and-choose* protokol, protokol ini berkerja sebagai berikut: satu kegagalan adalah kegagalan keseluruhan protokol. Pprotokol dapat terus berjalan selama prover masih dianggap legitimate (tidak melakukan failure/kegagalan). Dan ketika tingkat kepercayaan telah mencapai level tertentu,protokol dianggap berhasil.

Beberapa prinsip Zero-Knowledge:

- Verifier tidak bisa mendapatkan informasi apapun dari protokol
- Prover tidak bisa mencurangi verifier
- Verifier tidak bisa mencurangi prover
- Verifier tida bisa berpura-pura menjadi prover kepada pihak ketiga.

Ada tiga mode yang bisa digunakan pada protokol Zero-Knowledge:

- *Interactive*, dimana Prover dan Verifier menjalankan protokol secara interaktif, membangun kepercayaan langkah demi langkah.
- *Parallel*, dimana Prover membuat beberapa problem dan Verifier menanyakan beberapa solusi dalam satu waktu. Bisa digunakan pada koneksi dengan respon yang lambat.
- *Off line*, dimana Prover membuat bebrapa problem dan kemudian menggunakan fungsi hash pada data dan kumpulan problem untuk memainkan peran verifier, untuk memilih solusi yang diinginkan secara acak, kemudian prover memasukkan solusinya pada pesan. Bisa digunakan untuk tanda tangan digital.

3. AUTENTIKASI SMART CARD

Proses otentikasi antara kartu pintar (*prover*) dengan perangkat penerima (*verifier*) dilaksanakan di tempat transaksi, langsung seperti di bank atau pasar swalayan. Proses identifikasi secara kriptografi dari kedua unit ini disebut *card authentication* atau *node authentication*. Dengan perangkat perantara ini, perangkat penerima akan memverifikasi otentikasi dari kartu pintar kemudian kartu pintar akan membuktikan identitasnya terhadap perangkat *verifier*. Sehingga hal ini akan memberikan suatu proses otentikasi yang saling mendukung untuk keamanan yang berlapis terhadap sistem informasi.

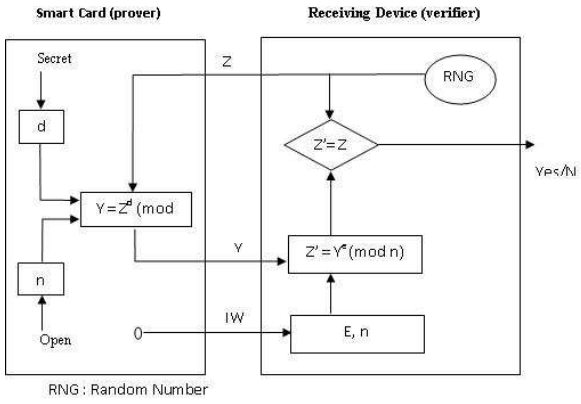
2.1 Autentikasi dengan RSA

Untuk proses otentikasinya, nomor acak Z didekripsi oleh kartu pintar dengan menggunakan kunci rahasia kartu, d.Proses verifikasi berlangsung dengan kunci publik (n,e) pada perangkat *verifier* yang mana perkalian dari n dengan p dan q disebarkan ke publik oleh *verifier* pada saat proses verifikasi tersebut.
Jika n = pq adalah hasil perkalian dua buah bilangan prima yang besar dan d adalah kunci rahasia yang dipilih oleh kartu sehingga gcd(d, $\Phi(n)$) = 1 , dimana $\Phi(n) = (p-1)(q-1)$ maka fungsi eksponensial modular $Y \equiv Z^d \pmod n$ adalah suatu fungsi *trapdoor* satu arah. Jika $Y \equiv Z^d \pmod n$ untuk suatu nomor acak Z maka hanya satu cara praktis untuk mencari Z adalah dengan menggunakan eksponensial publik e dengan $de \equiv 1 \pmod{\Phi(n)}$ dan menghitung $Y^e \pmod n$.

Adapun prosedur otentikasi kartu pintar dengan algoritma RSA adalah sebagai berikut:

1. Perangkat *verifier* memancarkan nomor acak ke kartu pintar
2. Kartu pintar mengirim balik *identification word* (IW) dan nomor acak Z yang telah didekripsi dengan kunci rahasia kartu, d dengan persamaan $Y \equiv Z^d \pmod n$.
3. Proses verifikasi berlangsung dengan mengenkripsi nomor acak Z yang sebelumnya telah didekripsi oleh kartu pintar dengan persamaan $Z' \equiv Z^e \pmod n$ dan kemudian dibandingkan hasilnya dengan nomor acak Z yang asli.

Secara diagram blok, proses otentikasi kartu pintar dapat terlihat dalam gambar 3. dibawah ini.



Gambar 2. Diagram blok proses autentikasi dengan algoritma RSA

2.2 Autentikasi dengan Zero-Knowledge

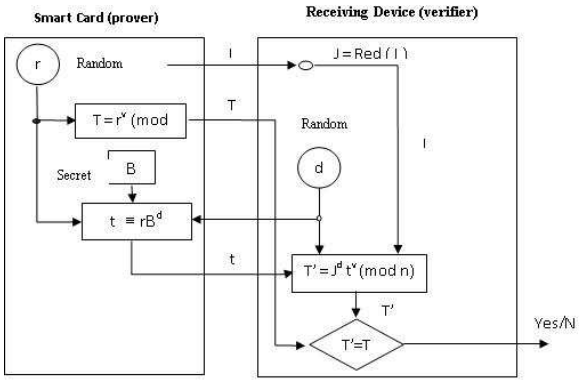
Protokol berbasis *Zero Knowledge* yang telah dikembangkan oleh Guillou dan Quisquater lebih tepat untuk otentikasi kartu pintar karena bekerja hanya menggunakan pertukaran satu nomor otentikasi B dalam setiap mikroprosesor dan hanya mengecek satu nomor saksi t. Selain itu, untuk melakukan proses ini, kartu *chip* hanya memerlukan memori lebih kecil dan komunikasi yang lebih sederhana, tetapi memerlukan waktu yang lebih lama.

Pada prinsipnya identifikasi berlangsung dengan dipilihnya dua bilangan prima yang besar, p dan q, (masing-masing 256 bit) oleh pembuat kartu dan dikalikan menjadi n = pq. Nomor n ini merupakan suatu konstanta publik yang diketahui juga oleh perangkat *verifier*. Sedangkan dua bilangan prima p dan q rahasia dan hanya diketahui oleh pembuat kartu. Selanjutnya bilangan n dengan v secara bersama dipublikasikan oleh pembuat kartu ke perangkat *verifier*. Ukuran bilangan v adalah 30 bit dan merepresentasikan paduan antara kecepatan dan keamanan. Bilangan J (ukurannya sama dengan n) merupakan bagian dari identitas kartu. Untuk mengidentifikasi kartu maka digunakan *redundancy rule* yaitu, J = Red (I) yang mana nomor otentikasi B harus secara rahasia dibangun untuk memenuhi persamaan $JB^v \equiv 1 \pmod n$ dengan J = Red (I).

Gambar 3 menunjukkan prosedur dari proses otentikasi kartu pintar dengan menggunakan protokol berbasis *Zero Knowledge* dan secara rinci diuraikan di bawah ini, yaitu :

1. Prosesor kartu secara rahasia memilih suatu bilangan integer r dengan range $0 < r < n-1$ secara acak dan menghitung T yang disebut nomor test dengan persamaan $T \equiv r^v \pmod n$. Kemudian kartu memancarkan T dengan identitas kartu, I ke perangkat *verifier*.
2. Perangkat *verifier* memilih suatu bilangan integer d secara acak dengan range $0 - v - 1$ dan mengirimkannya ke kartu (prover).
3. Kartu menghitung nomor saksi t, dengan persamaan $t \equiv rB^d \pmod n$ dan mengirimkannya ke *verifier*.
4. Untuk memverifikasi nomor saksi t maka *verifier* menghitung $J^d t^v \pmod n$ dan membandingkannya dengan nomor test T sebagai berikut :

$J^d t^v \pmod n \equiv J^d (rB^d)^v \pmod n$
$\equiv (JB^v)^d r^v \pmod n$
$\equiv T$ karena $JB^v \pmod n \equiv 1$.



Gambar 3. Diagram blok proses autentikasi dengan algoritma *Zero Knowledge*

4. ANALISA

Perbedaan prinsip secara rinci antar metode autentikasi tersebut membutuhkan diskusi tersendiri, tetapi tabel di bawah ini akan memberikan gambaran dasar secara umum dari setiap metode autentifikasi kartu pintar ditinjau dari prosedur yang harus dilakukan.

Tabel berikut merupakan resume ringkas dan dengan pendekatan diperoleh hasil perbandingan antara metode autentikasi algoritma kunci pablik RSA dan algoritma berbasis *zero knowledge*.

No	Karakteristik	Algoritma Kunci Publik (RSA)	Algoritma Berbasis Zero Knowledge
1	Persamaan Enkripsi	$Y \equiv Z^d \pmod n$	$T \equiv r^v \pmod n$
2	Persamaan Dekripsi	$Z' \equiv Y^e \pmod n$	$T = J^d t^v \pmod n$
3	Jumlah pemakaian kunci	Dua kunci (d dan n)	Satu kunci (B)
4	Perlu Random Generator	Tidak	Tidak
5	Pemakaian memori chip	Lebih besar	Lebih kecil
6	Komunikasi	Rumit	Sederhana
7	Tingkat keamanan sistem	Tinggi	Tinggi
8	Penanganan thd. trouble	Rumit	Rumit
9	Algoritma /Chiper	Sulit	Sulit
10	Kerumitan realisasi	Sedang	Sedang
11	Waktu proses	Sedang	Lebih lama
12	Perlu akses ke Card Issuer's Master Key	tidak	tidak

Tabel 1 Perbandingan RSA dan Zero-Knowledge

5. KESIMPULAN

Hasil perbandingan menujukan setiap algoritma memiliki kelebihan tersendiri, namun kelebihan algoritma dengan basis Zero-Knowledge, yakni pada bagian komunikasi yang lebih sederhana dan pemakaian memori yang lebih kecil membuat allgoritma ini lebih tepat digunakan dalam smart card.

DAFTAR REFERENSI

[1] R. Munir, "Diktat Kuliah IF5054", Program Studi Teknik Informatika Institut Teknologi Bandung, 2006.

[2] Aronsson, Hannu A. "Zero Knowledge Protocols and Small Systems".1995. http://www.tml.tkk.fi/Opinnot/

Rhee, Man Young, "Cryptography and Secure Communications", Mc Graw Hill, Singapore, 1994.

Choi, Soon-Yong dan Andrew B. Wilson, "Smart Cards: Enabling Smart Commerce in the Digital Age", *White Paper The University of Texas*, Austin, 2000.

http://ic.engin.brown.edu/classes/EN160S07/

http://lsiwww.epfl.ch/LSI2001/teaching/webcourse/ch01/ch01.html

http://www.wisdom.weizmann.ac.il/~oded/zk-tut02.html