Pemanfaatan Algoritma Kriptografi Dalam Pembuatan Smart Card

Martha Monica (13510080)
Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganeca 10 Bandung 40132, Indonesia
martha.monica@students.itb.ac.id

***Abstrak*— *Smart card* merupakan salah satu teknologi yang sekarang banyak digunakan dalam penyimpanan, identifikasi, dan autentikasi data serta pemrosesan aplikasi. Makalah ini akan membahas mengenai perbandigan dari berbagai algoritma kriptografi yang dapat dimanfaatkan dalam menjaga keamanan informasi dalam sebuah *smart card*.**

***Kata kunci*— *smart card*, kriptografi, keamanan informasi.**

I. PENDAHULUAN

Kriptografi merupakan ilmu yang mempelajari teknik-teknik, baik secara matematis ataupun intuitif, yang berhubungan dengan aspek keamanan informasi seperti bagaimana menjaga kerahasiaan dari suatu data dan informasi, bagaimana menjaga keabsahan dari suatu data, bagaimana agar integritas suatu data atau informasi dapat tetap dipertahankan, serta berbagai macam cara untuk autentikasi data / informasi.

Saat ini, penggunaan ilmu kriptografi berkembang sangat pesat. Tidak sedikit orang yang tertarik untuk mengembangkan berbagai macam cara dalam ilmu kriptografi ini. Semua ini berkaitan dengan salah satu tujuan mendasar dari ilmu kriptografi yaitu bagaimana kita dapat menjaga kerahasiaan dari suatu informasi.

Perkembangan ilmu kriptografi bisa dibilang cukup stabil dan pesat. Semenjak ilmu ini pertama kali ditemukan, banyak sekali orang yang ikut serta memberikan kontribusi dalam menciptakan berbagai teori dan algoritma baru dalam menjaga kerahasiaan informasi tertentu.

Seiring dengan perkembangan ilmu kriptografi, perkembangan teknologi juga sedang berkembang sangat pesat. Banyak sekali ditemukan berbagai macam teknologi baru yang digunakan dalam berbagai peralatan yang menunjang kehidupan manusia saat ini.

Salah satu permasalahan utama dari perkembangan teknologi ini adalah bagaimana cara untuk menjaga keamanan dari berbagai informasi yang tersimpan dalam seluruh teknologi ini. Banyaknya pembajakan dan kejahatan yang dilakukan dengan memanfaatkan teknologi semakin marak terjadi. Oleh karena itu, sangatlah penting untuk menjaga keamanan dari suatu data dan informasi.

II. SMART CARD

Smart card merupakan salah satu teknologi yang menggunakan kartu seukuran kantong yang mengandung *integrated circuit* ataupun *microchip* di dalamnya. *Smart card* saat ini banyak digunakan untuk menyimpan data, pembayaran tunai elektrik, identifikasi, dan berbagai pemrosesan aplikasi lainnya.

Smart card dapat menyimpan informasi lebih banyak dibandingkan kartu dengan *magnetic stripe.* Selain itu, beberapa *smart card* dapat diprogram secara khusus untuk keperluan tertentu, bahkan beberapa di antaranya mendukung penggunaan lebih dari satu aplikasi ataupun penambahan aplikasi baru dalam penggunaannya.

Saat ini, *smart card* digunakan dalam berbagai teknologi yang kita gunakan sehari-hari, mulai dari kartu yang digunakan untuk telepon selular, kartu *e-ticketing,* kartu identitas/identifikasi diri (SIM, KTP, KTM, dsb), bahkan sebagai *e-wallet* pada beberapa negara.

III. ALGORITMA

Pada makalah ini, dijelaskan berbagai algoritma yang digunakan dalam pengamanan informasi yang tersimpan di dalam *smart card*.

3.1 Algoritma ElGamal

Algoitma ElGamal merupakan algoritma dalam kriptografi yang termasuk algoritma asimetris. Algoritma ElGamal terdiri dari tiga proses, yaitu proses pembentukan kunci, proses enkripsi dan proses dekripsi. Algoritma ini termasuk ke dalam algoritma block chiper, yaitu melakukan proses enkripsi pada blok-blok plainteks yang akan menghasilkan blok-blok chiperteks yang kemudian digabungkan menjadi hasil secara keseluruhan.

Algoritma pembangkitan kunci dalam algoritma El-Gamal dapat dijelaskan sebagai berikut :

1. Pilih sembarang bilangan prima p (p dapat di-share di antara anggota kelompok)
2. Pilih dua buah bilangan acak, g dan x, dengan syarat g < p , yang dalam hal ini 1 √ k √ p – 2.
3. Hitung $y = g^x \bmod p$.

Hasil dari algoritma ini:

Kunci public : tripel (y, g, p)

Kunci privat : pasangan (x, p)

Algoritma enkripsi yang digunakan dalam algoritma El-Gamal dapat dijelaskan sebagi berikut :

1. Susun plainteks menjadi blok-blok m1, m2, …, (nilai setiap blok di dalam selang [0, p – 1].
2. Pilih bilangan acak k, yang dalam hal ini 1 √ k √ p – 2.
3. Setiap blok m dienkripsi dengan rumus

$$a = g^k \bmod p$$

$$b = y^k m \bmod p$$

Pasangan a dan b adalah cipherteks untuk blok pesan m. Jadi, ukuran cipherteks dua kali ukuran plainteksnya.

Algoritma dekripsi yang digunakan dalam algoritma El-Gamal dapat dijelaskan sebagi berikut :

1. Gunakan kunci privat *x* untuk menghitung $(a^x)^{-1} = a^{p-1-x} \bmod p$
2. Hitung plainteks *m* dengan persamaan:

$$m = b/a^x \bmod p \quad = b(a^x)^{-1} \bmod p$$

3.2 Algoritma RSA

Algoitma RSA merupakan algoritma dalam kriptografi yang termasuk algoritma kunci-publik yang paling terkenal dan paling banyak diaplikasikan dalam berbagai teknologi yang digunakan saat ini. Algoritma RSA terdiri dari tiga proses, yaitu proses pembentukan kunci, proses enkripsi dan proses dekripsi.

Algoritma pembangkitan kunci dalam algoritma RSA dapat dijelaskan sebagai berikut :

1. Pilih dua bilangan prima $p \neq q$ secara acak dan terpisah untuk tiap-tiap *p* dan *q*.
2. Hitung N dengan persamaan:

$$N = p\,q.$$

3. Hitung φ dengan persamaan:

$$\varphi = (p\text{-}1)(q\text{-}1).$$

4. Pilih bilangan bulat (*integer*) antara satu dan φ ($1 < e < \varphi$) yang juga merupakan *coprime* dari φ.
5. Hitung *d* dengan persamaan :

$$de \equiv 1 \pmod{\varphi}.$$

Hasil dari algoritma ini:

Kunci public : pasangan (N,e)

Kunci privat : pasangan (N,d)

Algoritma enkripsi yang digunakan dalam algoritma RSA dapat dijelaskan sebagai berikut :

1. Susun plainteks menjadi blok-blok m1, m2, …, (nilai setiap blok di dalam selang [0, p – 1].
2. Hitung chipertext c_i dengan rumus :

$$c_i = m_i^{\,e} \bmod N$$

Algoritma dekripsi yang digunakan dalam algoritma RSA dapat dijelaskan sebagai berikut :

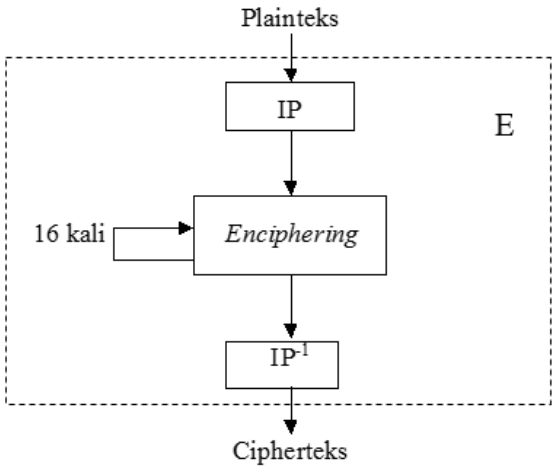
1. Gunakan kunci privat untuk menghitung $m_i = c_i^{\,d} \bmod N$
2. Carilah nilai m dengan rumus

$$c^d \equiv (m_i^e)^d \equiv m_i^{ed} \bmod N$$

Nilai m merupakan pesan semula yang dikirimkan.

3.3 Algoritma DES

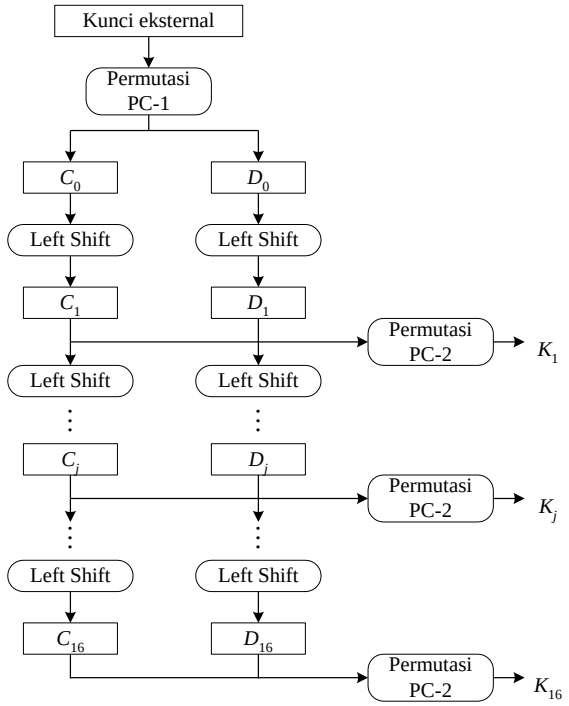
Algoitma DES merupakan algoritma dalam kriptografi yang termasuk algoritma kunci-simetri dan tergolong ke dalam jenis *block chiper*. Dalam algoritma DES, setiap blok akan diporses ke dalam 16 putaran di mana setiap putaran akan menggunakan kunci internal yang berbeda. Kunci internal itu sendiri dibangkitkan dengan menggunakan kunci eksternal. Setiap blok akan mengalami permutasi awal (IP), 16 putaran *enchipering*, dan inversi pemutaran awal (IP^{-1}) yang dapat digambarkan melalui skema berikut :



Skema Global Algoritma DES

Pembangkitan kunci internal dalam algoritma DES dibangkitkan dari kunci eksternal (64 bit) yang diberikan oleh pengguna. Kunci eksternal itu kemudian masuk ke proses pembangkitan kunci internal sebanyak 16 kali. Kunci internal yang dimaksud adalah kunci yang dihasilkan dari setiap putaran.

Untuk lebih jelasnya, proses pembangkitan kunci internal dapat digambarkan melalui skema berikut ini :



Skema Pembangkitan Kunci Internal

Setelah proses pembangkitan kunci internal berhasil dilakukan, setiap blok plainteks akan mengalami 16 kali putaran *enchipering.* Yang dimaksud dengan putaran *enchipering* ini adalah, setiap blok plainteks akan diproses ke dalam jaringan Feistel.

Model jaringan Feistel yang paling umum digunakan adalah sebagai berikut

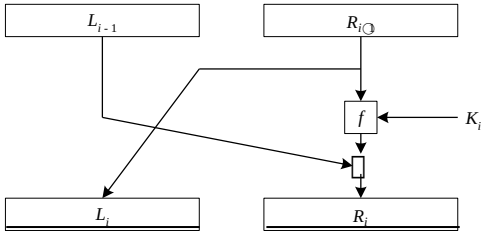
a. Bagi blok yang panjangnya n bit menjadi dua bagian, kiri(L) dan kanan (R), yang masing-masing memiliki panjang n/2
b. Block chiper diproses secara berulang di mana hasil dari putartan ke-i ditentukan dari hasil putaran sebelumnya dengan persamaan sebagai berikut

$$L_i = R_i + 1$$
$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$$

Keterangan :

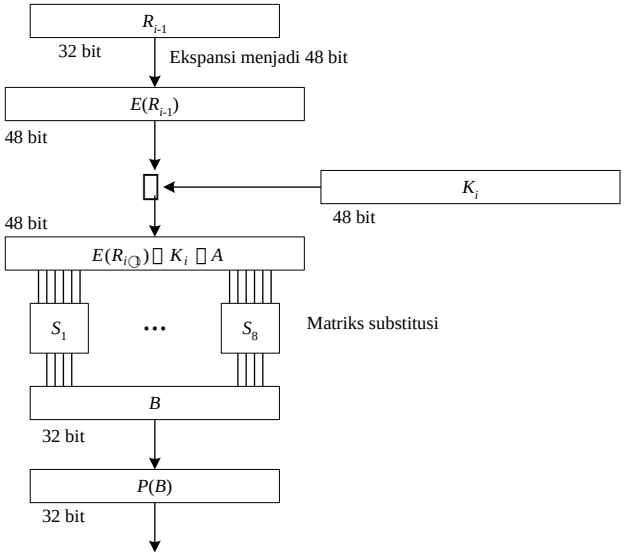
i = 1,2, …, r (r adalah jumlah putaran)
K_i = upa-kunci (subkey) pada putaran ke-i
f = fungsi transformasi (di dalamnya terdapat fungsi subsitusi, permutasi, dan/atau ekspansi, kompresi)

Skema pemrosesan pada jaringan feistel dapat dijelaskan melalui skema berikut ini :



Skema *Enchipering* pada Jaringan Feistel

Proses komputasi yang dilakukan dalam fungsi *f* dapat dijelaskan melalui skema berikut ini :



Skema Komputasi Fungsi *f*

Dalam komputasi fungsi *f* , E merupakan hasil ekspansi yang memperluas blok R_{i-1} yang berukuran 32 but menjadi 48 bit menggunakan matriks permutasi tertentu.

Kotak-S (S-box) merupakan matriks yang berisi substitusi sederhana yang memetakan satu atau lebih bit dengan satu atau lebih bit yang lain. Pada kebanyakan algoritma block chiper, kotak-S memetakan m bit masukan menjadi n bit keluaran, sehingga kotak-S tersebut dinamakan mxn S-box. Ada beberapa pendekatan yang dapat digunakan untuk mengisi S-box

a. Dipilih secara acak
   Untuk S-box yang berukuran kecil, cara pengisian secara acak tidak aman, namun untuk S-box yang besar cara ini cukup bagus.

b. Dipilih secara acak dengan aturan tertentu
   Cara ini sama seperti dengan cara pengisian secara acak, namun nilai acak yang dibangkitkan diuji terlebih dahulu apakah memenuhi sifat-sifat tertentu.

c. Dibuat oleh manusia (*man-made*)
   Nilai yang di entry ke dalam S-box dibangkitkan dengan teknik yang lebih intuitif.

d. Dibuat berdasarkan perhitungan matematis (*math-made*)
   Nilai yang di entry ke dalam S-box dibangkitkan berdasarkan prinsip matematika yang terbukti aman dari serangan kriptanalis.

Hasil ekspansi $E(R_{i-1})$ kemudian di XOR dengan kunci internal K_i menghasilkan vektor A 48 bit. Vektor A tersebut kemudian dikelompokkan menjadi 8 kelompok masing-masing 6 bit yang akan menjadi masukan ke proses subtitusi Kotak- S yang memberikan keluaran 4 bit.

Keluaran proses substitusi menghasilkan vektor B dengan panjang 48 bit yang kemudian akan diacak menggunakan proses matriks permutasi P. Bit-bit P(B) akan di XOR dengan L_{i-1} menghasilkan R_i.

Proses dekripsi terhadap chiperteks pada algoritma DES menggunakan algoritma yang sama dengan proses enkripsi. Pada proses dekripsi urutan kunci yang digunakan merupakan kebalikan dari urutan kunci pada proses enkritpsi. Chiperteks akan mengalami 16 putaran *dechipering* sama seperti proses *enchipering* pada proses enkripsi.

IV. Pembahasan dan Analisis

4.1 Algoritma ElGamal

Algoritma El-Gamal masih merupakan algoritma yang banyak digunakan dalam penjagaan keamanan informasi dalam *smart card*. Hal ini disebabkan adanya algoritma yang cukup kompleks, maka berbagai penyerangan yang dilakukan masih belum bisa menembus pertahanan dari algoritma El-Gamal ini.

Algoritma ini memiliki kelebihan karena pembangkitan kunci yang menggunakan perhitungan khusus dengan menggunakan kekuatan perpangkatan dan logaritma dengan angka yang cukup besar dan menghasilkan hasil enkripsi yang berukuran dua kali dari ukuran semula sehingga menimbulkan efek mengecoh untuk pihak-pihak yang mencoba menyerang.

Di sisi yang lain, hal itu juga menyebabkan algoritma ini memiliki kekurangan dikarenakan untuk pemrosesan algoritma membutuhkan *resource* dan *processor* yang mampu untuk melakukan komputasi yang berukuran besar dan kompleks terutama untuk proses pembangkitan kunci yang digunakan. Kemungkinan kesalahan yang dilakukan pada saat pemrosesan dengan menggunakan algoritma ElGamal bisa dibilang sangat kecil, sehingga hasil pemrosesan dari algoritma ini tergolong akurat.

Berdasarkan tingkat keamanan, mungkin Elgamal unggul dibandingkan dengan algoritma yang lain, namun dikarenakan keterbatasan memori dan pemrosesan yang dapat dilakukan pada *smart card* menyebabkan algoritma ElGamal tidak setenar penggunaan algoritma lainnya.

4.2 Algoritma RSA

Seperti algoritma Elgamal, algoritma RSA juga masih merupakan algoritma yang banyak digunakan dalam penjagaan keamanan informasi dalam *smart card*. Sedikit berbeda dengan algoritma ElGamal, keamanan dari algorima RSA disebabkan oleh pemrosesan yang cukup rumit dalam memfaktorkan beberapa bilangan prima yang digunakan dalam algoritma RSA.

Kekuatan algoritma RSA terletak pada kesulitan untuk memfaktorkan bilangan prima yang digunakan. Apabila sudah diketahui nilai-nilai bilangan prima yang digunakan, pesan yang dienkripsi akan dengan mudah ditemukan. Oleh karena itu, penemu dari algoritma ini menyarankan nilai bilangan prima yang digunakan tidak kurang dari 100 digit sehingga akan menjadi sulit untuk memfaktorkan bilangan tersebut. Dengan digit angka yang sebesar itu, maka dibutuhkan waktu komputasi 4 miliar tahun dengan asumsi algoritma pemfaktoran yang digunakan adalah algoritma tercepat saat ini dan komputer yang digunakan memiliki kecepatan pemrosesan 1 milidetik. Sama halnya seperti algoritma ElGamal, lemungkinan kesalahan yang dilakukan pada saat pemrosesan dengan menggunakan algoritma ElGamal bisa dibilang sangat kecil, sehingga hasil pemrosesan dari algoritma ini tergolong akurat. Hal inilah yang membentuk kekuatan untuk algoritma RSA.

Walapun begitu, algoritma RSA juga masih memiliki beberapa kekurangan. Waktu yang dibutuhkan untuk pemrosesan algoritma RSA ini cukup banyak sehingga dalam implementasinya algoritma ini lebih sering digunakan untuk proses enkripsi terhadap kunci publik dan kunci privat dibandingkan untuk proses enkripsi terhadap pesan itu sendiri.

Algoritma RSA bisa dibilang cukup unggul karena tidak memakan resource yang banyak namun tetap aman dan akurat untuk digunakan. Beberapa ahli kriptografi menyarankan penggunaan *padding scheme* pada pemrosesan algoritma ini untuk menambah tingkat keamanan dari pemrosesan pesan atau informasi tersebut.

4.3 Algoritma DES

Sedikit berbeda dengan kedua algoritma sebelumnya, algoritma DES ini tidak bermain dengan pemrosesan matematika yang rumit. Keamanan dari algoritma DES ini sendiri terletak pada kunci-kunci internal yang dibangkitkan pada saat pemrosesan algoritma ini.

Pemrosesan dari algoritma DES bisa dibilang menggunakan fungsi-fungsi sederhana. Sebagai pelengkap dari kesederhanaan ini, algoritma DES menggunakan pemrosesan yang berulang-ulang dengan penambahan beberapa proses permutasi maupun beberapa usaha

confusion yang dilakukan dengan ekspansi matriks selama pemrosesan algoritma DES itu sendiri.

Jika dibandingkan dengan kedua algoritma sebelumnya, algoritma DES merupakan algoritma yang memiliki tingkat keamanan yang paling kecil. Algoritma DES berhasil dipecahkan dalam kurun waktu pencarian dua hari walaupun dibutuhkan sangat banyak chip yang digunakan dalam proses pencarian dengan menggunakan *brutoforce.*

Kekuatan terbesar dari algoritma DES ini adalah kerahasiaan dari pengisian kotak-S serta jumlah putaran yang dilakukan dalam pemrosesan jaringan Feistel-nya. Menurut beberapa ahli, delapan putaran mungkin sudah cukup untuk membuat chiperteks sebagai fungsi acak dari setiap bit plainteks dan chiperteks. Namun, penelitian yang dilakukan membuktikan dengan jumlah putaran kurang dari 16 kali, algoritma ini lebih mudah dipecahkan dengan *known-plaintext attack.*

V. KESIMPULAN

Pemanfaatan berbagai algoritma kriptografi dalam pembuatan *smart card* masih terus berkembang. Dari ketiga algoritma yang telah dibandingkan sebelumnya, algoritma ElGamal memiliki keunggulan pada tingkat keamanan. Hal ini dikarenakan dalam pemrosesan pembangkitan kunci yang digunakan pada algoritma ElGamal lebih kompleks dibandingkan algoritma RSA dan DES. Semakin kompleks kunci yang dibangkitkan maka tingkat keamanan dari algoritma tersebut akan semakin tinggi, namun akan berdampak pula dengan semakin besar resource yang digunakan.

Teknik kriptografi DES merupakan cara yang lebih cocok digunakan apabila *resource* yang tersedia lebih sedikit dan sederhana karena pemrosesan yang dilakukan cukup sederhana namun memiliki tingkat keamanan yang paling kecil. Di sisi yang lain, algoritma ElGamal sangat cocok digunakan untuk proses komputasi yang didukung dengan *resource* dan *processor* yang cukup memadai sehingga walaupun tingkat keamanan yang dihasilkan paling tinggi, masih tidak terlalu banyak digunakan dalam pembuatan *smart card.*

Pemecahan yang mungkin bisa digunakan adalah penggunaan algoritma RSA. Walaupun tingkat keamanan yang tersedia tidak setinggi apabila menggunakan algoritma ElGamal, namun masih lebih aman dibandingkan DES. *Resource* yang dibutuhkan juga tidak sesedikit apabila menggunakan algoritma DES, namun tidak sebesar yang dibutuhkan pada algoritma Elgamal sehingga mungkin algoritma RSA dapat menjadi pilihan yang tepat untuk penjagaan keamanan informasi yang tersimpan dalam *smart card*.

REFERENSI

[1] http://www.rsa.com/rsalabs/staff/bios/bkaliski/publications/other/kaliski-smart-card-crypto-scia-1998.ppt diakses pada 24 April 2013

[2] http://searchsecurity.techtarget.com/definition/smart-card diakses pada 24 April 2013

[3] http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2012-2013/Algoritma%20ElGamal%20(2013).ppt diakses pada 19 Mei 2013

[4] http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2012-2013/Algoritma%20RSA%20(2013).ppt diakses pada 19 Mei 2013

[5] http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2012-2013/Algoritma%20Kriptografi%20Modern_bag2%20(2013).ppt diakses pada 19 Mei 2013

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah ini merupakan tulisan asli, bukan adaptasi dari jurnal pihak lain,dan bukan plagiarisme.

Bandung, 19 Mei 2013

Martha Monica
(13510080)