PEMAKAIAN KRIPTOGRAFI KUNCI PUBLIK UNTUK PROSES ENKRIPSI DAN TANDATANGAN DIGITAL PADA DOKUMEN E-MAIL

Aji Supriyanto
Fakultas Teknologi Informasi, Universitas Stikubank Semarang
E-mail : ajimedia@yahoo.com

Abstraksi:

E-mail merupakan perangkat surat elektronik yang sudah umum digunakan oleh kebanyakan orang terutama yang sering menggunakan internet. Banyak keluhan yang terjadi atas penggunaan e-mail, seperti adanya virus, spam, dan bocornya dokumen dan sebagainya. Meskipun terdapat sebagian vendor webmail yang menyediakan keamanan standar sejenis PGP, namun belum diterapkan keamanan pada dokumen penyerta e-mail (*attachment*). Penelitian ini bermaksud untuk membuat model keamanan pada *attachment* e-mail dengan mengimplementasikan prinsip keaslian, integritas, dan kerahasian dokumen. Model yang dibuat yaitu dengan menggunakan algoritma kunci asimetris (kunci public) untuk proses enkripsi-dekripsi, dan tandatangan digital menggunakan hash SHA1. Pemakain algoritma asimetris yaitu dengan cara menerbitkan dua pasang knci public, dimana sepasang untuk proses enkripsi dekripsi dokumen attachment, dan sepasang kunci untuk proses enkripsi-dekripsi tandatangan digital (*digital signature*). Pengujian dilakukan dengan cara melakukan verifikasi dokumen yang dienkripsi, dan proses validasi digunakan untuk dokumen yang didekripsi. Hasilnya akan menunjukkan dokumen yang valid atau tidak valid.

Kata Kunci : *E-mail, attachment, enkripsi-dekripsi, hash, RSA, SHA1, dan valid.*

PENDAHULUAN

Pemakaian fasilitas e-mail untuk melakukan komunikasi atau pengiriman surat elektronik saat ini sudah jamak digunakan oleh oran, baik yang menggunakan webmail gratis, maupun menyediakan sendiri mail server atau menyewa melalui ISP. Banyak kasus yang terjadi tentang tidak amannya pemakaian e-mail saat ini misalnya terkena virus, trojan, spam, pembobolan, tidak otentikasinya e-mail, *man in them midle attack* dan gangguan yang lain.

Sebuah dokumen dikatakan aman jika memenuhi tiga aspek utama yaitu kerahasiaan (*confidentiality*), integritas (*integrity*), dan ketersediaan (*Availability*). Ketiga aspek tersebut dapat dilakukan dengan cara menerapkan sistem keamanan dengan memanfaatkan kriptografi.Untuk itu dalam penelitian ini akan dibahas cara yang pertama, yaitu bagaimana cara mengamankan dokumen e-mail (*attachment*) dengan memanfaatkan kriptografi kunci publik untuk proses penandatangan dokumen e-mail dan sekaligus untuk proses enkripsi-dekripsinya. Prinsip keamanan yang dibangun akan dapat membuktikan keasliannya dokumen *attachment* e-mail dengan proses verifikasi dokumen e-mail yang dikirimkan dan proses validasi dokumen e-mail yang diterima oleh yang berhak. Latarbelakang tersebut menjadi dasar untuk dilakukan pembahasan penelitian yaitu Bagaimana cara membuat dokumen *attachment* e-mail yang aman dengan memanfaatkan kriptografi kunci publik dan dapat dibuktikan keasliannya.

Tujuan dari penelitian ini adalah membuat model keamanan dokumen e-mail khususnya dokumen yang disertakan (*attachment*) agar terjaga kerahasiaan, integritas, dan keasliannya. Model keamanan yang dibangun dengan merancang sebuah kriptografi yang menggunakan kunci publik RSA dan fungsi hash SHA1 yang digunakan untuk proses enkripsi-dekripsi dan penandatangan sebuah dokumen e-mail.

PEMBAHASAN

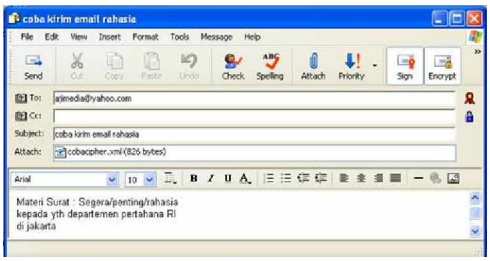
Model Sistem

Model sistem yang digunakan untuk menyelesaikan permasalahan yaitu dengan metodologi riset dan pengembangan (*Research and Development*/R&D). Riset dapat dilakukan terhadap objek e-mail yang disediakan oleh vendor penyedia layanan e-mail baik yang gratis seperti yahoo dan gmail, atau yang berlisensi seperti outlook express milik Microsoft. Dalam penelitian ini objek yang akan dilakukan uji coba adalah outlook express, yang saat ini mudah dijumpai pada

para pengguna e-mail. Sedangkan pengembangan software yaitu teknik analisis, desain, implementasi, dan evaluasi sistem dilakukan dengan teknik pengembangan model *prototyping*.

Analisis Sistem

Materi atau bahan penelitian yang digunakan untuk analisa diperoleh melalui observasi langsung terhadap literatur-literatur yang membahas tentang e-mail sebagai sarana untuk pengiriman dan penerimaan atau pembacaan surat elektronik. Selanjutnya melakukan pengamatan terhadap objek yaitu e-mail dalam hal ini dicoba menggunakan dengan outlook express, yang mana file atau dokumen attachment dan sistem keamanannya diperlihatkan dalam gambar seperti dibawah ini.



gambar 1. *Sistem Pengamanan Standar E-mail Outlook Express*

Dari gambar. diatas ditunjukkan bahwa penerapan system keamanan baik tandatangan dan proses enkripsi dilakukan hanya pada proses membuka e-mail yaitu ketika akan dibuka oleh calon penerima, namun tidak melakukan pengamanan terhadap isi dokumen attachment. Bentuk penerapan keamanannya menggunakan teknik sejenis PGP, yang pada akhir-akhir ini banyak fakta laoran yang menyebutkan bahwa system keamanan dengan PGP ternyata sudah dapat dibobol atau ditembus keamanannya. Sehingga jika suatu ketika system keamanan standart tersebut dapat ditembus atau dibobol, maka dokumen attach juga akan dengan begitu mudah untuk dibaca atau disadap. Untuk alasan tersebut, maka file atau dokumen attach perlu dilakukan atau diberikan system keamanan tersendiri.

Atas dasar alasan tersebut, maka perlu dilakukan analisis kebutuhan untuk melakukan teknik penyandian (enkripsi-dekrpsi) dan teknik penndatangan digital (*digital signature*) terhadap dokumen e-mail *attachment*. Kebutuhan analisisnya berupa algoritma kriptografi kunci publik, dalam hal ini digunakan RSA, dan fungsi hash SHA1.

Algoritma RSA digunakan untuk membangkitkan dua pasang kunci publik yang masing-masing digunakan untuk pengirim dan calon penerima dokumen e-mail. Pengirim memiliki satu pasang kunci (privat dan publik), dimana kunci privat untuk melakukan penandatanganan dokumen (verifikasi), sedangkan kunci publiknya untuk proses enkripsi dokumen e-mail. Calon penerima e-mail juga memiliki satu pasang kunci (privat dan publik) lain, dimana kunci privat digunakan untuk mendekripsi dokumen e-mail dan kunci publik digunakan untuk memvalidasi tandatangan pengirim.

Agar dokumen e-mail dapat diimplementasikan tandatangan digital dan enkripsi-dekripsi pada bagian tertentu dari dokumen maka perlu dilakukan hal-hal berikut ini:

1. Dokumen harus dalam bentuk attachment dari e-mail
2. Dokumen attachment tersebut dibagi menjadi dua bagian yaitu bagian kepala surat (*header document*) dan bagian isi surat (*content document*).
3. Format dokumen harus dalam bentuk XML untuk memudahkan pemilihan bagian dokumen yang akan diamankan (enkripsi dan tandatangan).
4. Syarat dokumen XML harus memenuhi atauran yang *well-formed*.
5. Menentukan pemilihan bagian dokumen yang diamankan yaitu kepala surat (*header*), isi surat (*content*), dan keseluruhan dokumen (*all document*).

Langkah Penelitian

Langkah-langkah yang perlu dilakukan dalam menyelesaikan penelitian ini adalah :

1. Membuat methode *RandomePrime*, yaitu bilangan acak prima yang dalam hal ini tersedia dalam library program java.
2. Membuat methode *ExtendedEuclid*, dari algoritma euclida, untuk menguji kebenaran bilangan prima.
3. Membuat methode RSA untuk mengolah bilangan prima acak terpilih, menjadi pasangan kunci asimetris (privat dan publik). Methode RSA ini harus menghasilkan dua buah pasangan kunci asimetris, dimana satu pasang untuk pengirim dan satu pasang untuk calon penerima. Kunci privat pengirim digunakan untuk proses penandatanganan,

dan kunci publik pengirim untuk proses enkripsi dokumen. Sedangkan Kunci privat calon penerima digunakan untuk mendekripsi dokumen terenkripsi, dan kunci publik digunakan untuk memvalidasi tandatangan pengirim.

4. Membuat methode untuk mengimplementasikan fungsi hash SHA sehingga menghasilkan *digest value* dari sebuah pesan, untuk proses tandatangan digital.
5. Membuat program untuk menggabungkan proses tandatangan (*signature*) dan proses enkripsi.
6. Membuat program untuk membaca dokumen e-mail format XML yang *well-formed* . Selanjutnya membuat nama file cipher-nya dari file XML tersebut.
7. Dari file cipher yang dihasilkan akan dibaca dengan memilih bagian yang akan dtandatangani dan dienkripsi yaitu bagian kepala (*header*), bagian isi (*content*) dan seluruh bagian dokumen (*alldocument*).
8. File dokumen XML yang di-*cipher*-kan tersebut dikirim melalui aplikasi attachment e-mail misalnya menggunakan outlook express.
9. Membuat program untuk mendekripsi dan memisahkan dokumen asli dengan tandatangan yaitu dengan membuang tag tanda tangan dari dokumen XML bertanda tangan jika tanda tangan telah terbukti.
10. Menentukan validaitas dokumen e-mail yang diterima.

Rancangan Sistem

1. Membuat File Data XML

File XML yang akan dilakukan pengujian diberi nama file COBA.XML. Dalam membuat dokumen XML yang baik, format susunan element diatas harus disusun dengan terstruktur atau baik (*well-formed*.. Maka struktur file COBA.XML yang *well-formed* adalah sebagai berikut :

```
<DOKUMEN>
    <KEPALA>
        <KDDOK>RHB-0001</KDDOK>
        <JNSDOK>SANGAT RAHASIA</JNSDOK>
        <JDUL>OPERASI PKL</JDUL>
        <TGLDOK>10-10-2006</TGLDOK>
        <PENGIRIM>DRS. IMAM SYAFII, MM</PENGIRIM>
        <NIP>12345001</NIP>
        <KOORDINAT>0.0000</KOORDINAT>
    </KEPALA>
    <ISI>AKAN DILADAKAN OPERASI PKL DI SEKITAR
SIMPANG LIMA HARI KAMIS 12-10-2006 JAM 10.30 WIB
HARAP KUMPUL KOORDINASI 1 JAM SEBELUMNYA DI
HALAMAN KANTOR WALIKOTA
    </ISI>
</DOKUMEN>
```

Listing Program 1. File ISIDOK.XML yang *well-formed*

Proses enkripsi dokumen XML nantinya dilakukan hanya pada sebagian elemen dokumen XML yang dipilih, yang terdiri dari tiga pilihan yaitu header (elemen <KEPALA>), isi (elemen <ISI>, dan keseluruhan elemen <DOKUMEN>).

2. Rancangan Susunan Kelas

Kelas yang dibuat dalam penelitian ini meliputi kelas RandomPrime, kelas ExtendedEuclid, kelas RSA, kelas Otentikasi, dan kelas pembacaan data. Kelas-kelas tersebut saling berkaitan untuk membentuk proses penandatanganan dan enkripsi pada dokumen XML yang telah dibuat sebelumnya.

a. Kelas Bilangan Prima

Kelas bilangan prima dalam program ini dinamakan kelas RandomPrime. Kelas tersebut dipergunakan untuk menghasilkan bilangan prima secara acak (*random*). Aplikasi kelas dibuat dengan menggunakan bahasa pemrograman Java, yang menyediakan representasi BigInteger untuk memenuhi kebutuhan pemakaian bilangan yang sangat besar untuk menghasilkan bilangan prima yang besar pula. Kelas ini adalah kelas yang pertama dibangun karena kelas ini nantinya akan digunakan dalam algoritma RSA, yaitu sebagai bahan untuk mengisikan data p, data q dan data e.

Kelas yang menghasilkan bilangan prima secara acak adalah sebagai berikut:

```
/*********************************
 *  Kompilasi :  javac RandomPrime.java
 *  Eksekusi  :  java RandomPrime N
 *  Membangkitkan nilai dengan N-bit integer
 *  prima.
 ******************/
import java.math.BigInteger;
import java.util.Random;
import java.security.SecureRandom;

public class RandomPrime {
    public static void main(String[] args) {
        int N = Integer.parseInt(args[0]);
        SecureRandom random = new
SecureRandom();
        BigInteger prime =
BigInteger.probablePrime(N, random);
        System.out.println(prime);
    }
}
```

b. Kelas RSA

Kelas ini mengimplementasikan algoritma RSA dengan memanfaatkan kelas RandomPrime yang telah dibuat sebelumnya. Kelas RSA akan menghitung nilai n, phi atau

totien n, dan menentukan nilai pasangan kunci publik (e) dan privatnya (d). Nilai-nilai tersebut dapat dihasilkan jika nilai p dan q terlebih dahulu ditentukan, yang dapat dicari dengan file RandomPrime.java seperti diatas.

Langkah-langkah menghitung nilai n, phi, e, dan d adalah sebagai berikut :

- Menyeleksi nilai p dan q bilangan prima acak yang hampir sama besarnya.
- Menghitung nilai n = p*q
- Menghitung nilai phi = (p-1)(q-1)
- Pilih bilangan bulat (integer) e, dimana e=gcd{phi,e}=1 dengan nilai e adalah (1<e<phi)
- Menghitung nilai $d=e^{-1}$ mod phi

c. Kelas Otentikasi

Kelas otentikasi merupakan kelas yang paling penting, karena akan melakukan implementasi algoritma kriptografi kunci publik menggunakan RSA dan fungsi hash SHA1 untuk membuktikan bahwa komunikasi dokumen XML yang dilakukan kedua belah pihak (pengirim dan penerima) dapat dinyatakan dengan aman. Dalam kelas ini keamanan dokumen XML akan dibuktikan keamanannya yang memenuhi tiga aspek sekaligus yaitu otentikasi (*otentication*), terintegrasi (*integration*), dan Kerahasiannya (*Confidential*).

d. Kelas Pembacaan Dokumen Attachment.

Kelas ini merupakan kelas pertama atau sebagai program utama yang berfungsi memanfaatkan kelas-kelas lain yang telah dibuat sebelumnya, untuk membaca dan menguji semua kelas yang telah dibuat.

Hasil Penelitian

Model pengujian yang dilakukan yaitu dengan menerapkan seluruh langkah-langkah yang telah dibahas sebelumnya.

1. Pengujian kelas RandomPrime

Kelas Random Prime diuji dengan membaca sejumlah N digit data untuk menghasilkan bilangan prima sesuai dengan nilai N data tersebut. Nilai N diinputkan dengan nilai desimal seperti berikut ini :

```
F:\bab-tesis\aji1\aji-rev-
program\final>java RandomPrime 12
2339

F:\bab-tesis\aji1\aji-rev-
program\final>java RandomPrime 12
2797
```

2. Pengujian RSA

Pengujian nilai RSA akan menhasilkan nilai kunci publik(e), nilai kunci privat (d), dan modulus (n). Agar dapat menunjukkan kebenaran hasilnya, maka dibuat contoh pesan yang akan dienkripsi, kemudian contoh hasil enkripsinya dan dekripsinya, seperti berikut ini:

```
F:\bab-tesis\aji1\aji-rev-
program\final>java RSASimpleApp

***Pembangkitan kunci PUBLIK dan kunci
PRIVAT***
p = 3391
q = 3023
n = 10250993
phi = 10244580
e = 179
d = 8470379
Kunci Publik {n,e} = {10250993,179}
Kunci Privat {n,d} = {10250993,8470379}

****=====Contoh Data=====****
Data yang di Enkripsi = 1234567
Nilai Ciphertext  = 5646604
Proses Dekripsinya ...
Data Plaintext = 1234567
```

3. Pengujian Dengan Data

Pengujian dengan data adalah pengujian yang dilakukan terhadap pembacaan data pada file dokumen XML dengan struktur yang *well-formed* dan diuji kebenaran hasilnya setelah dilakukan pemilihan terhadap elemen yang diotentikasi.

a. Pengujian dengan Data Valid

Pengujian dengan hasil data valid dilakukan pada masing-masing elemen yang dipilih sebagai berikut :

1) Elemen <KEPALA>

Otentikasi dilakukan pada elemen <KEPALA> hasil data valid-nya adalah sebagai berikut:

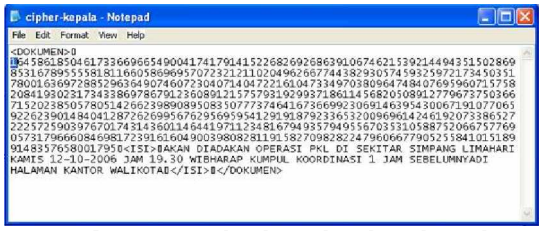
```
C:\rsa>java Rsa10

Tulis File XML yang akan di Enkrip :
coba.xml

Tulis Nama File Cipher :
cipher-kepala

 Silahkan Pilih :
 [1] header
 [2] isi
 [3] semua
 Pilihan anda? : 1
```

Hasil file : **cipher-kepala.xml** adalah:



gambar 2. *Hasil File cipher-kepala.xml*

2) Elemen <ISI>

Otentikasi dilakukan pada elemen <ISI> hasil data valid-nya adalah sebagai berikut :

C:\rsa>java Rsa10

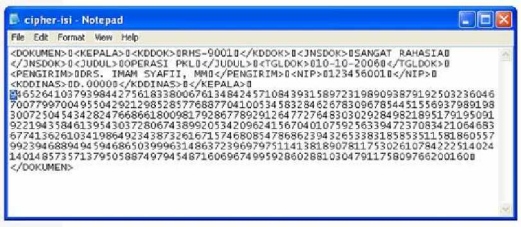
Tulis File XML yang akan di Enkrip :

coba.xml

Tulis Nama File Cipher : **cipher-isi**

Silahkan Pilih :
[1] header
[2] isi
[3] semua
Pilihan anda? : **2**

Hasil file : **cipher-isi.xml** adalah :



gambar 3. *Hasil File cipher-isi.xml*

3) Elemen <DOKUMEN>

Otentikasi dilakukan pada elemen <DOKUMEN> hasil data valid-nya adalah sebagai berikut:

C:\rsa>java Rsa10

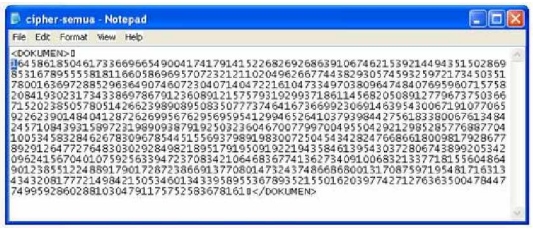
Tulis File XML yang akan di Enkrip :

coba.xml

Tulis Nama File Cipher : **cipher-semua**

Silahkan Pilih:
[1] header
[2] isi
[3] semua
Pilihan anda?: **3**

Hasil file : **cipher-isi.xml** adalah :



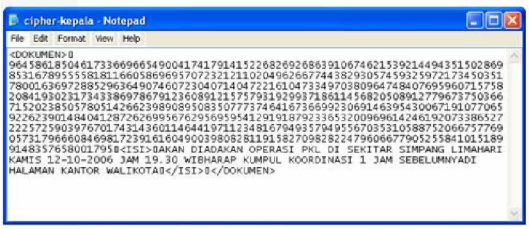
gambar 4. *Hasil File cipher-semua.xml*

B24. Hasil Bob: **Valid signature**

b. Pengujian dengan Data Invalid

1) Merubah isi File cipher-kepala.xml

Pada file cipher-kepala.xml yang terlihat seperti gambar 4.2 angka "1" yang tersorot dirubah dengan angka "9", seperti berikut ini :



gambar 5. *Perubahan isi data File cipher-kepala.xml*

maka hasilnya adalah sebagai berikut :

B24. Hasil Bob: **Invalid signature**

2) Pengujian dengan Kunci

Pengujian dilakukan dengan merubah kunci Publik Alice (e) yang digunakan untuk mendekripsi *digital signature* dengan nilai sembarang yang tidak sama dengan nilai e yang benar, sehingga akan tampil kesalahan "invalid signature" seperti berikut ini :

```
B21. Bob's decoded extended
digital signature:
sD",k;]AnSmx;/6?>⌂>?JM
2#P*O$=^xn
O<w/8.?<)+E|~Uq

B22. Bob's decoded digital
signature:
sD",k;]AnSmx;/6?>⌂>?JM
2#P*O$=^xnO<w/8.?

B23. Bob's digest:
0BF43E03AEF938D57F51D2784743FC9B
6F79E17E
B24.
```

Hasil Bob: **`Invalid signature`**

PENUTUP

Simpulan

Dari hasil penelitian yang dilakukan, maka diperoleh kesimpulan sebagai berikut:

a. Sistem keamanan dokumen attachment pada sebuah e-mail dapat dilakukan dengan menggunakan model enkripsi-dekripsi dan tandatangan digital menggunakan algoritma kunci publik RSA dan hash SHA1.
b. Sistem keamanan ini dapat memberikan keamanan otentikasi dokumen e-mail, untuk menghindari jebolnya atau tembusnya keamanan yang diterapkan e-mail standar seperti PGP.
c. Pengujian validitas dapat dilakukan berdasarkan elemen dokumen XML yang dipilih dengan cara membandingkan ”data valid” dengan perubahan data yang menghasilkan ”data invalid”.
d. Hasil validasi dapat juga dilakukan jika kunci publik dan kunci privat yang dimiliki masing-masing pihak (pengirim dan penerima) adalah sinkron. Dan apabila kunci yang digunakan tidak sinkron maka akan didapatkan hasil ”invalid”.

Saran

Hasil pembahasan penelitian ini dapat dikembangkan untuk menjadikan sebuah bentuk keamanan yang komprehensif, dengan melakukan hal-hal sebagai berikut :

a. Ujicoba dilakukan dengan menggunakan format dokumen berbentuk XML, dan belum diterapkan dengan membaca dokumen atau file attachment seperti data aslinya.
b. Dalam percobaan aplikasi, kunci-kunci yang dihasilkan dari algoritma RSA masih diambil secara langsung dari program. Hal ini dapat dikembangkan dengan model pembacaan secara terpisah, sehingga program aplikasi tinggal membaca kunci terpilih tanpa harus mengganti perubahan kunci melalui *source code*.
c. Model distribusi kunci kepada masing-masing pihak belum dibahas dalam penelitian ini, begitu juga penerapan sertifikat digital untuk pengamanan masing-masing kunci yang digunakan. Penelitian ini akan lebih bermanfaat jika kedua model tersebut dapat diterapkan.
d. Model percobaan dilakukan dengan menerapkan pembentukan file baru sebagi bukti file ciphernya. Dan belum dilakukan pengujian secara langsung pada pengirim dan penerimaan melalui e-mail. Untuk percobaan langsung perlu dilakukan modifikasi program.