Implementasi Algoritma Kunci Publik Kriptografi Kurva Eliptik pada Aplikasi *Email Mozila Thunderbird*

Ibnul Qoyyim

Laboratorium Ilmu dan Rekayasa Komputasi
Departemen Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung.

E-mail : if14066@students.if.itb.ac.id

Abstrak – Pada makalah tugas akhir ini, dilakukan implementasi algoritma kunci publik kriptografi kurva eliptik sebagai add-on pada aplikasi email mozila thunderbird yang dilengkapi dengan otomatisasi manajemen kunci agar pengguna yang tidak mengenal kriptografi dapat dengan mudah menggunakannya. Implementasi pada sisi client menggunakan Javascript dan XUL (XML User Interface Language), sementara implementasi pada sisi server menggunakan bahasa php dan database MySQL. Berdasar pengujian yang dilakukan, semua fitur telah diuji berjalan dengan baik sesuai dengan tujuan yang dicapai dalam tugas akhir ini.

***Kata Kunci**: kriptografi kurva eliptik, email, manajemen kunci.*

1. PENDAHULUAN

Saat ini *email* sudah sangat populer di kalangan pengguna internet, namun dengan cepatnya perkembangan *email* juga menyebabkan adanya kebutuhan akan keamanan data yang dikirim, Salah satu cara mengamankan data yang dikirim melalui *email* adalah dengan menggunakan kriptografi. Salah satu algoritma kriptografi adalah algoritma *Eliptic Curve Cryptography (ECC)* atau disebut juga kriptografi kurva eliptik yang termasuk algoritma kunci publik.

Algoritma kriptografi kurva eliptik adalah teknik kriptografi yang didasarkan pada pendekatan matematika dengan menggunakan kurva eliptik. Algoritma ini dipilih karena algoritma tersebut dapat diaplikasikan baik untuk enkripsi dekripsi maupun tanda tangan digital dan terbukti aman. Selain itu algoritma kriptografi kurva eliptik memiliki keuntungan dengan panjang kunci lebih kecil dibandingkan algoritma kunci publik lainnya tetapi sudah memiliki tingkat keamanan yang relatif sama [TRI05].

Kekuatan sistem kriptografi bergantung pada keamanan kunci. Kunci perlu dilindungi selama daur hidupnya. Pada umumnya pengguna layanan *email* tidak tahu tentang manajemen kunci seperti diatas karena itu implementasi algoritma kunci publik pada sebuah aplikasi email diharapkan dapat mengotomatisasi manajemen kunci publik yang terdiri dari beberapa langkah tersebut.

Mozila Thunderbird adalah aplikasi *email* yang *stand alone*, *open source*, dan tersedia untuk berbagai macam sistem operasi (*Windows* dan *Linux*), dan mendukung penggunaan *add-on*. Meskipun *Mozila Thunderbird* memiliki banyak kelebihan namun belum memiliki fasilitas untuk otomatisasi enkripsi-dekripsi dan tanda tangan digital.

2. KRIPTOGRAFI

Kriptografi adalah ilmu sekaligus seni untuk menjaga kerahasiaan pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya[MUN06]. Kriptografi menyediakan empat layanan, yaitu kerahasiaan, integritas data, otentikasi dan nipernyangkalan.

Berdasarkan kunci yang dipakai untuk enkripsi dan dekripsi, kriptografi dibedakan menjadi kriptografi kunci simetri (*symmetric-key cryptography*) dan kriptografi kunci nirsimetri (*asymmetric-key cryptography*). Kriptografi kunci simetri memiliki kunci enkripsi dan dekripsi yang sama. Kriptografi kunci nirsimetri atau kriptografi kunci publik memiliki kunci enkripsi dan dekripsi yang berbeda.

Kriptografi kurva eliptik adalah pendekatan kriptografi kunci publik yang mendasarkan keamanannya pada persoalan logaritma diskrit dari kurva eliptik pada bidang terbatas. Dalam tugas akhir ini akan menggunakan kurva eliptik pada bidang terbatas Fp sementara untuk enkripsi-dekripsi maupun tanda tangan digital karena kriptografi kurva eliptik memiliki banyak pendekatan akan digunakan salah satu pendekatan misal untuk enkripsi-dekripsi digunakan pendekatan yang paling sederhana yang dijelaskan pada [SCH96] sementara untuk tanda-tangan digital digunakan protokol ECDSA (*Elliptic Curve Digital Signature Algorithm*).

3. EMAIL

Email adalah singkatan dari electronic mail merupakan metode untuk mengirim dan menerima

pesan melalui system komunikasi elektronik/internet. Email memiliki beberapa protokol yang menyediakan layanan kriptografi : S/MIME, TLS, OpenPGP, *Identity based encryption, Mail sessions encryption*.

3.1. S/MIME

Sistem infrastruktur kunci publik pada S/MIME bersifat *centralized* artinya sertifikat yang mengikat kunci publik pada nama seseorang harus ditandatangani secara digital oleh CA (*Certificate Authority*). CA sendiri harus disahkan oleh RA (*Regional Authority*) yang juga harus disahkan oleh root.

Body yang terenkripsi pada email yang menggunakan S/MIME memiliki MIME type sendiri yaitu application/pkcs7-mime. Sementara tanda tangan digital pada S/MIME sama seperti tanda tangan digital pada umumnya yaitu merupakan hasil enkripsi hash dari pesan dengan menggunakan kunci privat pengirim. Tanda tangan digital tersebut dipisah dari body email sebagai bagian tersendiri yang memiliki MIME subtype application/(x-) pkcs7-signature.

3.2. PGP

Sistem infrastruktur kunci publik pada PGP tidak bersifat *centralized* seperti pada S/MIME namun skema *uncentralized* yang bernama vetting scheme, sementara model *trust*-nya disebut *web of trust.* Pendistribusian publik key membutuhkan *identity certificate* yang dibuat sedemikian rupa sehingga perubahan sertifikat ini oleh pihak yang tidak berhak akan terdeteksi. *identity certificate* menyatakan bahwa publik key terikat pada nama tertentu dan ditandatangani secara digital oleh pihak ketiga.

Sistem enkripsi PGP merupakan sistem *hybrid* menggabungkan kecepatan enkripsi sistem kunci simetri dan keamanan sistem kunci publik. Untuk mengenkripsi suatu pesan mula-mula PGP memampatkan pesan tersebut untuk dengan tujuan menghemat *bandwidth* dan *disk space* sekaligus menghilangkan pola pesan yang biasanya dimanfaatkan kriptanalis, kemudian PGP membuat suatu *session key* untuk sistem kunci simetri, pesan yang telah dimampatkan kemudian dienkripsi dengan *session key* tersebut yang hanya dipergunakan sekali, terakhir *session key* tersebut dienkripsi dengan kunci publik penerima pesan dan dimasukkan dalam pesan. Untuk mendekripsi pesan dilakukan kebalikannya penerima pesan mendekripsi *session key* menggunakan kunci privatnya, kemudian *session key* itu digunakan untuk mendekripsi pesan.

Tanda tangan digital pada PGP memiliki format yang sama yaitu merupakan hasil enkripsi *hash* dari pesan dengan menggunakan kunci privat pengirim. Namun berbeda dengan S/MIME tanda tangan digital ini dimasukkan sebagai body dari email dan dienkripsi bersama body email.

4. ANALISIS MASALAH

Pada implementasi kriptografi kunci publik kurva eliptik pada email ada beberapa hal yang berbeda dengan kriptografi pada *file* biasa. Misalnya pada pengiriman ke milis bagaimana penanganan kunci publik dan kunci privatnya yang digunakan apakah akan mendaftar seluruh kunci publik anggota milis atau cukup dibuat satu key saja atau implementasi pembagian kunci seperti *secret sharing*, begitu pula pada penanganan yang dilakukan jika email direply atau diforward, bagaimana dengan tanda tangan digital yang sebelumnya apakah perlu dihapus atau tidak, demikian juga dengan enkripsi yang dilakukan sebelumnya apakah perlu dihapus atau tidak. Selain itu melakukan metode otomatisasi juga memiliki permasalahan sendiri.

4.1. Penanganan Masalah Otomatisasi Kriptografi Kunci Publik

Untuk penanganan otomatisasi kunci publik kurva eliptik diharapkan program mengurus manajemen kunci sehingga pengguna awam dapat memanfaatkan keamanan tanpa perlu mahir dalam bidang kriptografi. Untuk mencapai hal tersebut program harus menjaga keamanan integritas kunci pada semua fase pada daur hidupnya.

4.2. Pengiriman Satu Pengirim pada Banyak Penerima

Pada kasus pengiriman satu pengirim pada banyak penerima solusi yang digunakan adalah dibuat sebuah blok pesan yang terenkripsi dengan *session key* namun *session key* dienkripsi kunci publik masing-masing penerima disertai dengan komentar alamat email yang kunci publiknya digunakan untuk mengenkripsi blok pesan tersebut. Untuk tanda tangan digital hanya dibubuhkan sebuah tanda tangan digital dari kunci privat pengirim, jadi pada sebuah email terdapat sebuah blok pesan dan sebuah tanda tangan digital yang terenkripsi dengan *session key*, dan beberapa *session key* yang dienkripsi, email yang memiliki banyak blok pesan yang terenkripsi tersebut dikirim pada banyak pengguna.

4.3. Pengiriman pada Mailing List

Untuk pengiriman pada milis (mailing list) seperti yang disebutkan pada analisis permasalahan ada beberapa alternatif solusi :

1. Menggunakan satu kunci untuk email pada milis,
2. Mendaftar keseluruhan kunci dari anggota milis,
3. Menggunakan *secret sharing* untuk berbagi kunci milis,

Dari pertimbangan maka dipilih solusi pertama dimana sebuah kunci digunakan oleh keseluruhan anggota dalam milis

Implementasi kunci publik untuk milis adalah dengan membuat suatu kunci publik yang terikat pada alamat milis tersebut, sementara kunci privat dari milis tersebut dimiliki seluruh anggota milis, hal tersebut dengan asumsi milis bersifat terbuka pada anggota milis tersebut sehingga email yang dikirim pada milis tidak perlu dirahasiakan pada anggota milis yang lain, namun dirahasiakan dari orang diluar milis.

4.4. Penanganan Forward Email

Pada email yang diforward tanda tangan digital oleh pengirim sebelumnya masih diperlukan oleh penerima selanjutnya karena perlu mengetahui pesan tersebut berasal dari sumber yang benar dan tidak diubah oleh perantara (aspek otentikasi dari kriptografi), selain itu perantara yang memforward email pada penerima selanjutnya perlu membubuhkan tanda tangannya sendiri untuk meyakinkan penerima bahwa dia benar-benar mendapat email dari perantara.

Untuk enkripsi pesan jika pada email sebelumnya dienkripsi maka enkripsi yang sebelumya dihapus dan diganti dengan enkripsi baru yang menggunakan kunci publik dari penerima selanjutnya. Dengan kata lain email sebelumnya didekripsi dengan kunci privat perantara dan kemudian dienkripsi dengan kunci publik dari penerima selanjutnya.

4.5. Penanganan Reply Email

Pada email yang direply tanda tangan digital sebelumnya boleh tidak disertakan, alasannya karena penerima email selanjutnya merupakan pengirim email yang direply sehingga dia tidak perlu memverifikasi apakah email itu benar dari dirinya, namun pihak yang mereply harus membubuhkan tanda tangan digitalnya agar pengirim email yang direply yakin bahwa dia berkomunikasi dengan pihak yang benar.

Untuk enkripsi jika pada email sebelumnya dienkripsi maka enkripsi yang sebelumya dihapus dan diganti dengan enkripsi baru yang menggunakan kunci publik dari pengirim email yang direply. Dengan kata lain email sebelumnya didekripsi dengan kunci privat pihak yang mereply dan kemudian dienkripsi dengan kunci publik dari pengirim email yang direply.

Dengan kata lain enkripsi-dekripsi dan tanda tangan digital pada email yang direply sama dengan mengirim email baru pada pengirim email yang direply.

5. ANALISIS DAN PERANCANGAN PERANGKAT LUNAK

Perangkat lunak yang dibangun digunakan untuk melakukan kriptografi kurva eliptik baik pada saat pengiriman dengan enkripsi dan pembubuhan tanda-tangan digital maupun saat menerima dengan dekripsi dan verifikasi tanda tangan digital.

Perangkat lunak yang dibangun merupakan add-on yang harus dipasang pada program Mozila Thunderbird kedua belah pihak. Add-On yang dipasang pada program Mozila Thunderbird dapat membaca dan mengubah body pesan bergantung kebutuhan pengguna jika pengguna mengirim email maka add-on dapat mengenkripsi dan membubuhkan tanda-tangan digital sebaliknya jika pengguna menerima email yang terenkripsi dan terdapat tanda-tangan digital maka program dapat mendekripsi dan memverifikasi tanda-tangan digital dalam email tersebut.

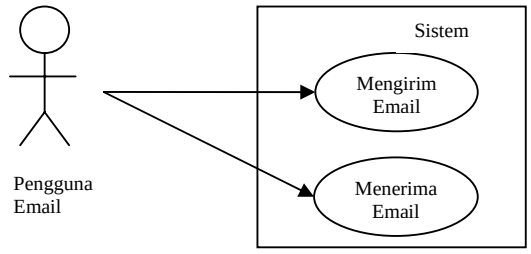
5.1. Diagram *Use Case*

Use case terdiri hanya seorang aktor yaitu seorang pengguna email yang dapat mengirim maupun menerima email. Diagram *use case* ini terdiri dari 2 *use case* yaitu mengirim email dan menerima email

Use case mengirim email menangani aksi yang berhubungan dengan mengirim email. Didalamnya terdapat proses pembubuhan tanda-tangan digital dan enkripsi.

Use case menerima email menangani aksi yang berhubungan dengan menerima email. Didalamnya terdapat proses dekripsi dan verifikasi tanda-tangan digital.

Diagram *use case* perangkat lunak dapat dilihat pada gambar 1.



Gambar 1 Diagram *use case* perangkat lunak

5.2. Perancangan Modul Program

Modul-modul yang terdapat dalam perangkat lunak terdiri dari lima buah modul, yaitu: modul utama, modul kurva eliptik, modul hashing menggunakan SHA, modul kunci simetri menggunakan AES, modul I/O *File*.

5.3. Perancangan Database

Karena *web server* akan diakses oleh banyak pengguna yang akan mencari atau mempublish kunci publik maka digunakan basis data untuk mempercepat akses. Basis data kunci publik yang akan dibuat memerlukan informasi alamat email yang digunakan pengguna lain untuk mencari kunci publik, informasi kurva eliptik yang digunakan, informasi titik G dan titik kunci publik, dan yang terakhir adalah informasi keberlakuan kunci yang digunakan pengguna lain untuk mengetahui apakah kunci publik tersebut masih berlaku. Dari informasi data yang disimpan diatas

dapat dibuat desain basis data seperti pada gambar 2

public_key	
PK	id_key
	email p a b g_x g_y pub_x pub_y expired_date

Gambar 2 Desain Basis Data

6. IMPLEMENTASI

Implementasi dilakukan pada lingkungan sebagai berikut :

a. Prosessor : AMD Athlon XP 2500+ 1.83GHz
b. Memori : 768 MB DDR
c. Harddisk : 200 GB
d. Sistem operasi : *Microsoft Windows XP SP2*
e. Server Package : XAMPP 1.5.0-pl1
f. Server Basis Data : *MySQL 5.0.15*
g. Server Web : Apache PHP 5.0.5
h. Bahasa Pemrograman : *Javascript*, PHP, XUL (*XML User Interface Language*)
i. Kakas Pengembangan : *Notepad++*
j. Email Client : *Mozilla Thunderbird, Portable Edition 2.0.0.4*

6.1. Implementasi Modul Perangkat Lunak

Modul perangkat lunak diimplementasikan dengan menggunakan kakas Notepad++ dengan bahasa pemrograman JavaScript. Implementasi kelas perangkat lunak dapat dilihat pada tabel

Tabel 1 Hasil Implementasi Modul Program

No.	Nama File	Keterangan
1.	`main.js`	Modul utama untuk manajemen kunci, enkripsi, dekripsi dan tanda tangan digital.
2.	`ec.js`	Modul untuk operasi kurva eliptik.
3.	`sha1.js`	Modul untuk operasi *hashing* menggunakan SHA.
4.	`aes.js`	Modul untuk kriptografi kunci simetri menggunakan AES.
5.	`io.js`	Modul untuk operasi *file* dan direktori.
6.	`overlays.js`	Modul untuk mengkondisikan lingkungan *add-on*.

6.2. Implementasi Antarmuka

Pengembangan antarmuka ini menggunakan pendekatan scripting PHP sehingga tidak ada kelas yang dihasilkan melainkan sebuah file PHP untuk masing-masin antarmuka. Daftar file-file fisik yang berisi implementasi antarmuka dapat dilihat pada tabel

Tabel 2 Hasil Implementasi Antarmuka

No	Nama File	Keterangan
1.	`thunderbirdOverlay.xul`	Modul *user interface* pada jendela utama *Mozilla Thunderbird*.
2.	`composeOverlay.xul`	Modul *user interface* pada jendela *Compose Mozilla Thunderbird*.

6.3. Implementasi Web-Server

Aplikasi web dibangun dengan menggunakan bahasa PHP. Hasil implementasi tersebut dapat dilihat pada tabel

Tabel 3 Hasil Implementasi Aplikasi Web

No.	Nama File	Keterangan
1.	`server.php`	Modul untuk koneksi dengan basis data *MySQL*.
2.	`publish.php`	Modul untuk penyebaran kunci publik.
3.	`search.php`	Modul untuk mencari dan mengambil kunci publik dari *server*.

7. PENGUJIAN

Pengujian perangkat lunak merupakan aktivitas menjalankan perangkat lunak dengan berbagai cara yang bertujuan untuk melakukan evaluasi terhadap perangkat lunak yang dibuat serta mendeteksi atau menemukan kesalahan perangkat lunak. Tujuan

pengujian disesuaikan dengan tujuan tugas akhir, yaitu apakah perangkat lunak mampu:

1. Melakukan enkripsi email menggunakan algoritma kriptografi kurva elliptik.
2. Melakukan dekripsi email menggunakan algoritma kriptografi kurva elliptik.
3. Membubuhkan tanda-tangan digital pada email menggunakan algoritma kriptografi kurva elliptik.
4. Melakukan verifikasi email menggunakan algoritma kriptografi kurva elliptik.
5. Melakukan otomatisasi manajemen algoritma kunci publik.

Seluruh pengujian dilakukan dengan asumsi kedua belah pihak telah memiliki Mozilla Thunderbird dengan add-on untuk kriptografi kurva eliptik.

Berdasarkan hasil pengujian, diperoleh hasil bahwa perangkat lunak berhasil memenuhi seluruh kebutuhan awal pengembangan.

8. KESIMPULAN

Kesimpulan yang didapat selama pelaksanaan tugas akhir ini adalah :

1. Metode otomatisasi manajemen kunci publik yang terdiri dari pembuatan kunci, penyebaran kunci, penyimpanan kunci, penggunaan kunci, penggantian kunci, dan penghancuran kunci dapat dibuat dengan cara melakukan otomatisasi di tiap-tiap bagian.
2. Solusi kasus khusus seperti : penanganan email yang terforward atau reply dengan tanda tangan digitalnya, penanganan pengiriman ke dan dari milis dapat ditemukan solusinya.
3. Add-On yang dapat melakukan proses manajemen kunci, enkripsi, dekripsi, dan tanda tangan digital menggunakan algoritma kriptografi kurva eliptik dapat dibangun.

Untuk pengembangan lebih lanjut, saran-saran yang dapat diberikan pada tugas akhir ini adalah sebagai berikut :

1. Pengembangan pada bahasa yang lebih rendah dari javascript agar mampu melakukan perhitungan yang lebih banyak yang pada akhirnya akan meningkatkan keamanan dan mempercepat komputasi.
2. Melakukan penyamaran kode sumber atau cukup dengan kompilasi yang disertai dengan pengecekan keaslian program agar kriptanalis tidak mudah untuk mengubah kode program.
3. Mengubah proses konversi antar format email HTML atau PlainText pada mozila thunderbird agar tidak mengubah data enkripsi dan tanda-tangan digital.
4. Karena kriptografi kurva eliptik memiliki beberapa macam pendekatan, perlu dilakukan studi pendekatan mana yang paling baik dari segi keamanan maupun kecepatan komputasi.

DAFTAR REFERENSI

[MUN06] Munir, Rinaldi. (2006). Diktat Kuliah IF5054 Kriptografi.

[SCH96] Schneir, Bruce. (1996). Applied Cryptography Second Edition. John Willey & Sons.

[TRI05] Triwinarko, Andy. (2005). Studi dan Implementasi Tanda Tangan Digital menggunakan Algoritma Kriptografi Kurva Eliptik.

DAFTAR PUSTAKA

Chris Caldwell. The First 1,000 Primes. http://primes.utm.edu/lists/small/1000.txt diakses pada tanggal 16 Agustus 2008.

David E. Ross. (2006). PGP: Pretty Good Privacy. http://www.rossde.com/PGP/ diakses pada tanggal 15 Mei 2008.

Herbert Hanewinkel. (2005). PGP Encryption Service http://www.hanewin.net/encrypt/PGcrypt.htm diakses pada tanggal 16 Agustus 2008.

John Hanna's Javascript Shopping & Crypto System. (2005). http://shop-js.sourceforge.net/ diakses pada tanggal 16 Agustus 2008.

Jörn Rönnow, (2006). eMail Encryption for the Lazy. http://www.dtek.chalmers.se/%7Ed97jorn/pgp/index.html diakses pada tanggal 15 Mei 2008.

Langenhoven.com. Gmail Encryption. www.langenhoven.com/code/emailencrypt/gmailencrypt.php diakses pada tanggal 28 Agustus 2008.

Mozilla Developer Center. (2007). Building an Extension. http://developer.mozilla.org/en/docs/Building_an_Extension diakses pada tanggal 4 Februari 2008.

MozillaZine.(2007).Extension Development. http://kb.mozillazine.org/Extension_development diakses pada tanggal 30 Agustus 2008.

Munir, Rinaldi. (2006). Diktat Kuliah IF5054 Kriptografi.

Network Associates, Inc. (1990-1999). How PGP works. http://www.pgpi.org/doc/guide/6.5/en/intro/ diakses pada tanggal 29 April 2008.

Network Working Group. (1996). MIME Security with Pretty Good Privacy (PGP). RFC2015

Schneir, Bruce. (1996). Applied Cryptography Second Edition. John Willey & Sons.

Stalling, William. (1998). Cryptography and Network Security : Principles and Practice Second Edition. Prentice Hall.

Triwinarko, Andy. (2005). Studi dan Implementasi Tanda Tangan Digital menggunakan Algoritma Kriptografi Kurva Eliptik.