Analisis Manajemen Kunci
Pada Sistem Kriptografi Kunci Publik

Vicky Fathurrahman[1], Anindya Chandra Astri[2] dan Renni Kusumowardani[3]

Program Studi Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Jalan Ganesha 10 Bandung 40132

E-mail : *if12064@students.if.itb.ac.id*[1], *if12070@students.if.itb.ac.id*[2], *if12082@students.if.itb.ac.id*[3]

Abstrak

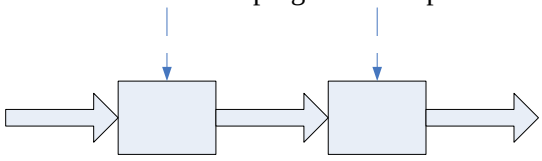
Sistem kriptografi kunci publik adalah sistem kriptografi dengan menggunakan dua buah kunci yaitu kunci privat dan kunci publik. Sistem kriptografi kunci publik memiliki banyak kelebihan. Namun, banyak juga kelemahan yang ada di balik itu. Manajemen pasangan kunci menjadi salah satu topik yang menarik untuk dibahas, karena banyak hal yang menjadikan sistem kriptografi kunci publik tidak sempurna terdapat pada manajemen pasangan kuncinya.

Makalah ini dibuat untuk membahas hal-hal yang menyebabkan ketdaksempurnaan pada sistem kriptografi kunci publik yang muncul selama daur hidup kunci. Dilakukan analisis solusi-solusi yang muncul sebagai kontribusi untuk ikut menyempurnakan sistem kriptografi kunci publik.

***Kata kunci**: Ketidaksempurnaan, Sistem Kriptografi Kunci Publik, Manajemen Kunci*

1. Pendahuluan

Sistem kriptografi kunci publik (kriptografi asimetri) adalah sistem kriptografi dengan menggunakan dua buah kunci yaitu kunci privat dan kunci publik. Kunci publik digunakan untuk melakukan enkripsi sedangkan kunci privat untuk melakukan dekripsi. Gambar 1 memperlihatkan deskripsi umum dari sistem kriptografi kunci publik.



Gambar 1 Sistem Kriptografi Kunci Publik

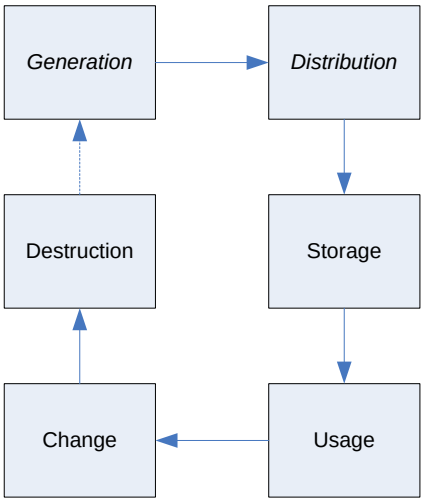
Beberapa kelebihan sistem kriptografi kunci publik diantaranya:

a) Hanya kunci privat yang perlu dijaga kerahasiaannya[1)]
b) Pasangan kunci tidak perlu diubah dalam kurun waktu yang lama[1)]
c) Dapat digunakan untuk mengamankan pengiriman kunci simetri dalam sistem kriptografi simetri[1)]
d) Lebih aman dan mudah untuk dikelola[2)]
e) Memiliki lingkup geografis yang luas[2)]

Di sisi lain, beberapa kelemahan sistem kriptografi kunci publik adalah[1)]:

a) Enkripsi dan dekripsi data umumnya lebih lambat daripada sistem kriptografi simetri
b) Ukuran cipherteks lebih panjang daripada plainteks
c) Ukuran kunci relatif lebih besar daripada kunci simetri
d) Cipherteks tidak memberikan informasi mengenai otentikasi pengirim.
e) Tidak ada algoritma kunci publik yang terbukti aman

Secara umum, tujuan manajemen kunci adalah menjaga keamanaan dan integritas kunci pada seluruh fase daur hidup kunci tersebut. Gambar 2 memperlihatkan daur hidup kunci secara umum.



Gambar 2 Daur Hidup Kunci

Selain beberapa kelemahan dari sistem kriptografi kunci publik yang telah disebutkan di atas, terdapat kelemahan lain yaitu pada bagian pengelolaan kunci publik (*public key management*).

Pengelolaan kunci publik dibebankan pada *client* (pengguna atau pemilik) kunci publik. *Client* harus selalu berhati-hati dalam penggunaan setiap kunci publik, dengan cara memvalidasinya setiap saat. Caranya adalah dengan membuat sertifikat kunci publik pada CA (*Certification Authority*), yaitu sebuah badan yang memiliki otoritas untuk memberikan seritifikat yang menyatakan keabsahan kunci publik. Lalu secara periodik melihat *directory*, yaitu basis data yang bisa diakses oleh publik, berisi sertifikat-sertifikat yang valid. Lalu memeriksa masa keberlakuan kunci publik pada *Certificate Revocation List* (CRL), yaitu basis data yang bisa diakses oleh publik, berisi sertifikat-sertifikat yang sudah tidak valid lagi.

Selain memvalidasi kunci publik setiap saat, *client* juga harus menjaga kerahasiaan kunci privatnya. Kedua hal tersebut sangat bergantung pada kedisiplinan *client* itu sendiri, karena sistem kriptografi kunci publik tidak mampu memaksa pemilik kunci untuk melakukan validasi kunci publiknya.

2. Manajemen Kunci Publik

Terdapat beberapa permasalahan pada manajemen kunci publik, yaitu pada fase-fase tertentu pada daur hidup kunci publik berikut:

a) Pembuatan-kunci
   Pengguna terlebih dahulu membuat pasangan kunci. Kemudian pengguna tersebut mendaftarkan identitasnya ke CA (tidak secara otomatis). CA akan menandatangani sertifikat yang menyatakan bahwa pengguna tersebut memiliki kunci publik tertentu. Pengguna tersebut juga menerima kunci publik milik *Root* CA. Selain itu pengguna akan memilih kata sandi rahasia yang digunakan untuk mengenkripsi kunci privatnya.

b) *Single-sign-on*
   Ketika *login*, pengguna harus mengetikkan kata sandi rahasianya untuk

mendekripsi kunci privat. Dengan kunci privat tersebut, pengguna sudah berpartisipasi dalam protokol kunci publik.

c) Mengotentikasi kunci publik lain
Untuk berkomunikasi secara aman dengan pengguna lain melalui jaringan, pengguna akan menggunakan kunci publik pihak lawan bicara, tentu saja yang sudah bersertifikat. Pengguna dapat memperoleh kunci publik tersebut secara langsung dari pengguna lain, atau melalui layanan *directory*. Sebelum menggunakan sertifikat tersebut, pengguna harus mengecek apakah sertifikat tersebut masih berlaku. Pengecekan dilakukan dengan melihat ke CRL. Setelah meyakini bahwa sertifikat tersebut valid, maka selanjutnya pengguna harus memvalidasi tanda tangan CA. Tahap ini rekursif sampai ke validasi tanda tangan Root CA.

d) Perubahan *password*
Pengguna secara rutin harus mengubah kata sandi rahasia yang digunakan untuk mendekripsi kunci privat.

e) Penghancuran kunci
Setiap sertifikat memiliki masa berlaku sampai beberapa bulan atau tahun. Jika pengguna mengubah kunci privatnya, maka dia harus memberi tahu kepada administrator CRL. CRL akan menyatakan bahwa sertifikat kunci publik milik pengguna tersebut sudah tidak berlaku lagi. Selain itu pengguna harus mengecek CRL setiap dia akan menggunakan suatu sertifikat, karena CRL dapat diubah kapan saja.

Dari beberapa fase dari daur hidup tersebut terlihat bahwa *client* kunci publik seringkali diharuskan untuk memproteksi dan memvalidasi sejumlah kunci. Setiap fase tersebut memiliki kelemahan sebagai berikut:

a) Pembuatan kunci → otentikasi pengguna.
Pada saat pembuatan sertifikat awal (inisialisasi), Bagaimana CA mengotentikasi pengguna jarak jauh?

b) *Single-sign-on* → manajemen kunci privat.
Pengguna harus menjaga kunci privatnya tetap berada dalam memori, selama *login-session*.

c) Mengotentikasi kunci publik lain → otentikasi CA.
Kriptografi kunci public tidak dapat mengamankan distribusi dan validasi dari kunci publik Root CA

d) Perubahan *password* → Pengguna tidak bisa dipaksa untuk memilih kata sandi rahasia yang bagus, dalam artian tidak mudah ditebak.

e) Penarikan kunci
Penarikan (*revocation*) kunci masih belum efektif, sehingga penyebaran kunci publik masih dilakukan tanpa adanya infrastruktur penghancuran kunci.

Kelemahan pada pembuatan dan penghancuran kunci memang dapat ditangani dengan infrastruktur terpusat, namun jika penggunaannya masih belum efisien, kemungkinan besar pengguna akan mengorbankan alasan keamanan demi alasan performansi.

3. Pembahasan

Dari kelemahan – kelemahan yang diungkapkan pada upabab 2, berikut

pembahasan untuk masing-masing kelemahan.

3.1. Pembuatan Kunci

Secara umum, CA dapat melayani permintaan pembuatan sertifikat kunci publik dalam jumlah yang banyak per satuan waktu. Sertifikat kunci publik adalah sebuah asuransi yang menunjukkan identitas dari pemegang kunci privat yang berkorespondensi.

Namun permasalahannya adalah CA tidak dapat dengan mudah mempercayai, atau dalam kata lain, tidak memiliki asuransi kebenaran dari identitas *client* baru yang akan divalidasi kunci publiknya. Untuk mendapatkan asuransi kebenaran identitas tersebut, idealnya CA harus melakukan pertemuan fisik dengan *client* baru tersebut dan melakukan pengecekan identitas *client* secara langsung. Namun hal tersebut akan memakan usaha yang sangat besar apabila *client* yang dihadapi berjumlah sangat banyak.

Salah satu solusi untuk menyelesaikan permasalahan ini adalah dengan cara menggunakan keamanan kunci simetri dalam melakukan otentikasi permintaan sertifikasi kunci publik. Dalam hal ini, skema yang dilakukan adalah dengan melakukan pengikatan ID *client* terhadap kunci publik yang akan disertifikasi seperti yang dilakukan oleh sistem otentikasi Kerberos yang dikembangkan oleh MIT. Namun tetap saja verifikasi terhadap pengikatan ID *client* dan kunci publik harus dilakukan melalui pertemuan langsung.

3.2. *Single-Sign-On*

Selain untuk menandatangani dan mendekripsi sebuah surat elektronik, kunci privat juga digunakan untuk menginisiasi hubungan dengan server. Dengan begitu, kunci privat akan berada pada memori komputer sepanjang klien melakukan hubungan sesi loginnya. Hal ini merupakan salah satu masalah keamanan penting bagi kerahasiaan kunci publik. Ancaman tidak hanya datang dari penyerang dijital seperti virus atau *worm* yang dibuat untuk mengambil mencuri kunci ini. Ancaman juga datang dari penyerang secara fisik yaitu pencuri. Apabila klien meninggalkan komputernya dalam keadaan masih terbuka dan komputer atau laptopnya dicuri orang maka akan sangat besar kemungkinan kunci privatnya bocor ke pihak yang tidak berhak.

Solusi untuk masalah ini tidak hanya satu. Solusi pertama adalah dengan meletakkan kunci privat ini di sebuah *smart-card* sehingga kunci privat tidak akan berada di memori. Solusi ini memiliki kelemahan karena walaupun kunci privat tidak lagi berada pada memori, kartu ini tetap harus berada pada pembaca-kartu selama klien *login*. Hal ini tentu saja tidak memberikan solusi apabila terjadi pencurian barang.

Solusi kedua adalah solusi pengamanan secara dijital. Kunci privat yang berumur sepanjang sesi *login* ini dapat digantikan dengan kunci sesi yang berumur relatif lebih pendek. Hal ini hanya menjadi solusi apabila pasangan kunci hanya digunakan untuk menandatangani pesan. Dalam penggunannya untuk mendekripsi pesan, alternatif solusi ini tidak dapat digunakan karena Agar dapat mendekripsikan pesan, klien harus meletakkan kunci privatnya pada memori dan dalam bentuk teks sepanjangn sesi *login*-nya.

Solusi yang terakhir adalah jelas bahwa pengamanan harus dilakukan secara fisik dan konvensional,, yaitu dengan menjaga komputer atau laptopnya dari pemakaian oleh pihak yang tidak berhak dan melakukan pelarangan untuk seluruh akses *remote* ke komputer miliknya. Hal ini sebenarnya sangat penting untuk dilakukan, karena apabila kunci-privat sampai jatuh ke tangan yang tidak berhak akibatnya akan sangat berbahaya sekali. Namun, biasanya klien masih belum sadar pentingnya keamanan kunci privat ini, seihingga perlu dilakukan pendekatan-pendekatan persuasif kepada klien untuk menyadarkan pentingnya menjaga kerahasiaan kunci privat dan satu-satunya cara untuk itu adalah dengan keamanan secara fisik dan konvensional seperti yang telah disebutkan.

3.3. Mengotentikasi Kunci Publik Lain

Client harus selalu melakukan otentikasi kunci publik dalam rantai otoritas CA sebelum sertifikat kunci publik *client* tersebut digunakan. Kegiatan tersebut tidak efisien apabila kegiatan tersebut harus dilakukan untuk setiap sertifikat kunci publik akan digunakan. Namun sistem kriptografi kunci publik tidak memiliki layanan untuk melakukan kegiatan validasi rantai CA secara otomatis.

Dalam melakukan validasi rantai CA, *client* dapat memilih salah satu dari tiga pilihan di bawah:

a) Melakukan protocol *hand-checking* terhadap salinan tanda tangan CA yang otentik.
b) Menjamin bahwa salinan tanda tangan CA pada CPU tidak dapat di rusak atau diubah oleh *attacker*.
c) Menggunakan sistem keamanan yang terpisah untuk memberikan salinan tanda tangan CA yang otentik ke CPU.

Solusi yang lebih praktis adalah dengan menjamin kunci *root* CA yang telah tevalidasi tidak dapat diubah atau dirusak oleh *attacker*. Beberapa cara untuk menjamin validitas kunci *root* adalah dengan menyimpannya dalam *smartcard*, menyimpannya dalam bentuk cipherteks atau menyimpan salinan kunci *root* itu sendiri.

3.4. Perubahan Sandi Kunci

Sistem kriptografi kunci publik tidak memiliki mekanisme pengontrolan kualitas sandi kunci yang diipilih klien karena sandi kunci ini hanya diketahui oleh klien. Padahal pemberian tanggal tenggat untuk setiap sandi kunci yang ada atau pemilihan sandi kunci yang tepat dapat membuat kunci privat terjaga kerahasiaanya. Sandi kunci yang tidak tepat hanya membuat kunci privat klien seaman sistem arsip. Ini tentu saja membuat kunci privat sangat rapuh untuk dienkripsi oleh pihak yang tidak berhak.

Alternatif solusi pertama adalah dengan menambahkan sebuah modul pada saat klien memilih sandi kunci. Modul ini secara otomatis melakukan pemeriksaan terhadap sandi kunci yang dipilihnya. Sandi kunci dicocokkan dengan kamus-kamus apakah merupakan kata-kata umum yang mudah ditebak, atau dapat juga digunakan mekanisme dan aturan-aturan yang lain yang dapat melakukan pemeriksaan terhadap sandi kunci.

Solusi untuk pemberian tanggal tenggat untuk sandi kunci adalah penanganan hal ini pada saat klien *login*.Sistem akan memeriksa tanggak tenggat sandi kunci. Apabila sudah

masuk masa tenggat, klien harus mengganti sandi kuncinya dengan yang baru. Pada saat penggantian modul pada solusi pertama otomatis dijalankan.

Dua solusi ini terlihat kurang mungkin untuk dilakukan. Pemeriksaan sandi kunci dan tanggat tenggat dilakukan oleh sistem yang berada pada server sehingga sandi kunci harus terlebih dahulu dikirm ke server, padahal sandi kunci pada sistem kriptografi kunci publik tidak membiarkan klien untuk membagi sandi kuncinya dengan pihak manapun.

Alternatif lainnya adalah dengan membuat aplikasi berbasis *desktop* untuk melakukan pemeriksaan terhadap calon pilihan sandi kunci sebelum klien mendaftarkannya. Solusi ini menuntut aplikasi yang mudah digunakan oleh klien. Karena aplikasi yang tidak nyaman akan ditinggalkan klien dan mungkin saja klien memilih aplikasi yang nyaman digunakan walaupun aplikasi baru ini tidak melakukan pemeriksaan calon pilihan sandi kunci secara maksimal misalnya. Tantangan lainnya adalah menyadarkan klien pentingnya menggunakan aplikasi pemeriksaan calon pilihan sandi kunci untuk memastikanm pilihan sandi kuncinya sudah tepat.

3.5. Penarikan Kunci

Sebelum kunci publik dapat digunakan, sertifikat kunci tersebut harus divalidasi dengan menggunakan dua cara:

a) Mengecek tanda tangan CA
b) Melihat CRL untuk mengecek apakah kunci publik tersebut masih aktif

Pada sistem kunci simetri, biaya untuk pembuatan dan penghancuran kunci hampir sama. Namun pada sistem kunci publik, biaya penarikan kunci jauh lebih besar daripada pembuatan kunci. Satu-satunya cara untuk menarik sebuah kunci publik adalah mengecek CRL sebelum menggunakan kunci tersebut. Jadi sebenarnya, infrastruktur kunci publik membutuhkan CRL hierarki, seperti pada CA. Pada akhirnya pengecekan CRL akan sama seperti validasi tanda tangan, namun dengan waktu yang 10 kali lebih lama. Perbedaan performansi tersebut akan menyebabkan pengguna untuk menghindari pengecekan penarikan kunci (CRL). Hal ini tentu saja menjadi kelemahan pada keamanan aplikasi yang menggunakan kriptografi kunci publik.

Manajemen waktu menjadi hal yang sangat penting pada CRL. Kebutuhan penarikan akan sangat penting jika tanda tangan digital melibatkan aspek ekonomi dan kontraktual. Ukuran CRL dapat diminimalisasi dengan menggunakan sistem kontrol akses untuk mengendalikan akses ke CRL, namun hal ini tidak dapat diterapkan pada jaringan yang besar.

4. Kesimpulan

Masih ada beberapa ketidaksempurnaan pada sistem kriptografi kunci publik. Beragam alternatif solusi muncul untuk mengatasi ketidaksempurnaan ini. Setelah dilakukan analisis untuk setiap solusi yang ditawarkan, ternyata solusi yang berorientasi kepada kesadaran klien yang paling memungkinkan untuk dilaksanakan.

Muncul masalah baru, yaitu menyadarkan klien untuk melakukan aktivitas-aktivitas yang bersifat menjaga pasangan kunci miliknya sendiri. Hal ini bukanlah hal yang mudah, namun setidaknya hal itu yang dapat kita lakukan. karena seperti telah dijelaskan pada bagian pembahasan solusi-solusi lain

yang ditawarkan kurang dapat menyelesaikan masalah.

[1] D. Davis, *Compliance Defect in Public-Key Cryptography*, MIT, 1997.
[2] R. Munir, *Diktat Kuliah IF5054 Kriptografi*, ITB, 2005