Bab 21

Aplikasi - Pengamanan Email

Meskipun banyak orang yang tidak menyadari, pengamanan *email* merupakan sesuatu yang penting untuk berbagai situasi. *Email* biasanya tidak dienkripsi jadi rentan terhadap penyadapan. Ini terutama jika menggunakan fasilitas seperti Yahoo atau Gmail, jelas pihak Yahoo atau Google dapat membaca *email* yang dikirimkan dari atau ke fasilitas mereka. *Email* juga bisa dipalsukan, seorang dapat saja mengaku sebagai orang lain dalam mengirim email, bahkan menggunakan *email address* orang lain sebagai *address* pengirim. Jadi seperti halnya dengan pengamanan sesi (lihat bab 20), diperlukan dua macam pengamanan untuk *email* yaitu:

- *authentication* dan
- enkripsi.

Untuk memastikan bahwa *email* bukan sesuatu yang dipalsukan, mekanisme *authentication* biasanya menggunakan *digital signature*. *Email* ditanda-tangan secara *digital* menggunakan kunci privat pengirim, dan diperiksa penerima menggunakan kunci publik pengirim. Untuk memastikan bahwa hanya penerima yang diinginkan yang dapat membacanya, *email* dienkripsi menggunakan kunci publik penerima, jadi hanya bisa didekripsi oleh pemegang kunci privat penerima. (Sebetulnya yang dienkripsi menggunakan kunci publik penerima adalah kunci sesi, sedangkan *email* dienkripsi menggunakan kunci sesi. Penerima mendekripsi kunci sesi menggunakan kunci privat, lalu mendekripsi *email* menggunakan kunci sesi.) Jadi cukup jelas bahwa pengamanan *email* memerlukan kriptografi *public key*.

Ada dua standard pengamanan *email* yang populer yaitu:

- S/MIME (*Secure/Multipurpose Internet Mail Extensions*) dan
- OpenPGP.

S/MIME berbasis pada format CMS (*cryptographic message syntax*), yang berawal pada PKCS#7, jadi berorientasi pada X.509 (lihat bagian 23.2), sedangkan OpenPGP berbasis pada format PGP (lihat bagian 23.1). Kedua standard juga menggunakan format *multipurpose internet mail extensions* (MIME) untuk integrasi dengan *email*. Cara berfungsi kedua standard sebenarnya serupa, hanya format saja yang berbeda. Tabel 21.1 memperlihatkan beberapa perbedaan format antara S/MIME dan OpenPGP.

Feature	S/MIME	OpenPGP
Message Format	CMS	PGP
Certificate Format	X.509	PGP
Symmetric Encryption	3DES, AES, IDEA, CAST	3DES, AES, CAST, Blowfish
Key Exchange	Diffie-Hellman, RSA	ElGamal, RSA
Digital Signature	RSA, DSA/DSS	DSA/DSS, RSA
Hash	SHA	SHA
MIME Encapsulation Signed Data	multipart/signed or CMS	multipart/signed with ASCII armor
MIME Encapsulation Encrypted Data	application/pkcs7-mime	multipart/encrypted

Tabel 21.1: Perbedaan S/MIME dengan OpenPGP

Tabel 21.1 menunjukkan bahwa S/MIME dan OpenPGP menggunakan kumpulan algoritma kriptografi yang serupa. Perbedaan terletak pada berbagai format yang digunakan. (Untuk mendapatkan informasi yang rinci mengenai format S/MIME versi 3.1, lihat [ram04]. Untuk mendapatkan informasi yang rinci mengenai format OpenPGP, lihat [cal07].) S/MIME bersumber pada "*the establishment*" yang awalnya berorientasi pada RSA dan berbagai standard PKCS yang dikembangkan oleh RSA. Meskipun tidak bisa disebut "*anti-establishment*," PGP dikembangkan oleh Philip Zimmerman sebagai alternatif untuk RSA yang saat itu (tahun 1991) masih dilindungi hak paten, dan sebagai taktik gerilya melawan kontrol pemerintahan Amerika Serikat terhadap kriptografi saat itu. Dengan mengendurnya kontrol pemerintahan Amerika Serikat terhadap kriptografi dan habisnya masa berlaku paten untuk RSA, pembagian pengguna kriptografi menjadi "*the establishment*" versus "*outsider*" sudah tidak berlaku lagi. Tren saat ini mengarah pada standardisasi dengan S/MIME sebagai standard yang dipilih. Bahkan GnuPG, suatu program *open*

source untuk OpenPGP, kini juga mendukung standard S/MIME. Oleh sebab itu selanjutnya kita akan lebih fokus pada S/MIME.

Secara garis besar, pengamanan *email* dilakukan dengan apa yang disebut *data enveloping* atau memasukkan data kedalam amplop. Berbagai jenis *data enveloping* antara lain:

- *Basic data enveloping.*
- *Compressed data enveloping.*
- *Encryption enveloping.*
- *Authenticated enveloping.*

Basic enveloping hanya sekedar membungkus data menjadi suatu unit tanpa memproses data. *Compressed data enveloping*, selain membungkus juga melakukan kompresi data (jadi yang dibungkus adalah data yang sudah dikompresi). Demikian juga *encryption enveloping* melakukan enkripsi terhadap data sebelum dibungkus. Sedangkan *authenticated enveloping* melakukan *authentication* terhadap data yang dibungkus. Yang membuat konsep *enveloping* sangat berguna adalah kita dapat mengkombinasi beberapa jenis *enveloping* secara berlapis. Sebagai contoh, kita dapat melakukan *compressed enveloping* untuk mengkompres data, kemudian melakukan *encrypted enveloping* terhadap *compressed envelope* untuk mengenkripsinya, kemudian melakukan *authenticated enveloping* terhadap *encrypted envelope* dengan melakukan *digital signing.*

Sesuatu yang sangat penting dalam S/MIME adalah *certificate* untuk kunci publik, baik kunci publik untuk mengecek *digital signature* maupun kunci publik untuk enkripsi. *Certificate* merupakan sesuatu yang seharusnya dibuat oleh seorang atau badan yang dapat dipercaya. Biasanya pembuat adalah suatu *certificate authority*, namun seorang dapat juga membuat *self-signed certificate*. Tentunya siapa yang dapat dipercaya adalah sesuatu yang relatif. Sebagai contoh, pemerintah Perancis tentunya tidak akan mempercayai *certificate authority* komersial dari Amerika Serikat seperti Verisign untuk keperluan resmi negara Perancis. Jika agak ragu dengan kebenaran *certificate*, kita dapat mengecek *fingerprint* dari kunci publik melalui jalur lain (misalnya telpon) dengan pemilik kunci publik. Manajemen *certificate* akan dibahas lebih lanjut di bab 23.

Berbagai program *email* terkemuka seperti Outlook dan Mozilla Thunderbird sudah mendukung S/MIME. Menggunakan program *email* yang sudah mendukung S/MIME, untuk mengenkripsi atau menanda-tangan secara *digital email* yang akan dikirim, tinggal memilih opsi sewaktu membuat *email.* Menerima *email* yang dienkripsi juga seolah menerima *email* biasa, asalkan *email* dienkripsi menggunakan kunci publik penerima. Akan tetapi ada resiko dengan penerimaan *email* yang dienkripsi, karena *virus scanner* biasanya

tidak bisa mendeteksi virus yang telah dienkripsi. Untuk mengurangi resiko sebaiknya ikuti petunjuk sebagai berikut:

- Jangan publikasikan kunci publik kita kecuali kepada orang-orang yang kita kehendaki.
- Jangan buka *email* yang dienkripsi kecuali dari orang-orang yang telah kita berikan kunci publik.

Verifikasi *digital signature* juga bisa otomatis, mirip dengan verifikasi *secure web page* (lihat bagian 20.1).

Untuk program *email* yang belum mendukung S/MIME, biasanya kita dapat menambahkan S/MIME, misalnya menggunakan cryptlib (lihat bagian 24.3). Untuk *web-based email* ceritanya agak berbeda. Mayoritas *web-based email* tidak memberikan fasilitas untuk S/MIME dan agak lebih sukar untuk mengintegrasi S/MIME dengan *web-based email*. Jika mendapatkan *email* yang diamankan menggunakan S/MIME, *envelope* akan berupa *attachment* yang dapat di*save* ke *file* contohnya `smime.p7m`. *File* `smime.p7m` dapat dibuka menggunakan program seperti p7mviewer atau menggunakan cryptlib (lihat bagian 24.3). Sebaliknya, jika ingin mengirimkan *email* yang diamankan menggunakan S/MIME, *file attachment* dapat dibuat menggunakan cryptlib. Untuk yang menggunakan Gmail, *browser* Mozilla Firefox kini mendukung penggunaan S/MIME dengan Gmail, dengan meng*install* Gmail S/MIME *extension*.

21.1 Ringkasan

Di bab ini kita telah bahas keperluan pengamanan *email* dan dua standard pengamanan *email* yaitu S/MIME dan OpenPGP. Beberapa program *email* terkemuka sudah mendukung S/MIME dan penggunaannya sangat mudah (istilahnya *seemless*), namun penerimaan *email* yang dienkripsi harus dengan hati-hati.