Bab 23

Aplikasi - PKI

Hampir semua aplikasi kriptografi di bidang teknologi informasi menggunakan kriptografi *public key*. Aplikasi dasar kriptografi *public key* memang untuk *key management* dan *digital signature*. Namun bermula dari aplikasi dasar, kebutuhan berkembang dan *certificate management* kini menjadi bagian penting dari kriptografi *public key*. Suatu *public key infrastructure* (PKI) adalah suatu infrastruktur yang mendukung

- *key management* termasuk *key generation*, *key exchange*, dan *key agreement*,
- *digital signing* dan *digital signature checking*, dan
- *certificate management* termasuk *certificate generation*, *certificate publishing*, *certificate checking*, dan *certificate revocation*.

Key generation dalam hal ini adalah proses pembuatan pasangan kunci publik dan privat, baik untuk keperluan *digital signature* maupun keperluan *key management*. Pembuatan kunci simetris tidak masuk disini karena tidak melibatkan unsur kriptografi publik, kecuali jika menggunakan *key agreement*. *Key exchange* adalah proses pengiriman kunci simetris secara aman menggunakan kriptogafi *public key*. *Key agreement* adalah proses pembuatan kunci simetris secara bersama menggunakan kriptografi *public key*, contohnya menggunakan Diffie-Hellman (lihat bagian 16.2). Kunci simetris hasil *key agreement* bisa berupa *seed* yang digunakan untuk membuat kunci simetris lainnya.

Digital signing adalah proses pembuatan *digital signature* contohnya menggunakan RSA atau DSA, sedangkan *digital signature checking* adalah proses pengecekan *digital signature* apakah sesuai dengan naskah yang di*sign* dan kunci publik yang di*claim*.

Certificate generation adalah proses pembuatan *certificate* untuk kunci publik, baik untuk keperluan *digital signature* maupun untuk keperluan *key management*. Secara garis besar, ada dua standard untuk format *certificate* yaitu format X.509 dan format PGP. *Certificate publishing* bisa dilakukan menggunakan fasilitas seperti LDAP (*lightweight directory access protocol*) atau cara yang lebih sederhana. *Certificate checking* adalah proses pengecekan *certificate*, baik dari segi format, maupun dari segi isi. *Certificate revocation* adalah proses pembatalan suatu *certificate* yang telah dibuat. Semua aspek *certificate management* dapat melibatkan *certificate authority* yaitu seorang atau suatu badan yang fungsi utamanya adalah mengesahkan *certificate*.

Solusi PKI biasanya berbasis pada X.509 atau pada PGP. Pada awalnya, solusi berbasis X.509 adalah solusi komersial, sedangkan solusi PGP populer di kalangan *open source* saat RSA masih dilindungi hak paten dan pemerintah Amerika Serikat melakukan kontrol yang ketat[1] terhadap penggunaan kriptografi. Dengan habisnya masa berlaku paten untuk RSA dan mengendurnya kontrol pemerintah Amerika Serikat terhadap penggunaan kriptografi, tren saat ini adalah standardisasi kearah X.509.

23.1 PGP

Solusi berbasis PGP untuk PKI menggunakan format dan "cara" PGP. "Cara" PGP adalah pendekatan gerilya dimana struktur hirarkis formal tidak terlalu diperhatikan. Jadi konsep *certificate authority* tidak digunakan. Setiap pengguna dapat menentukan sendiri apakah suatu kunci publik *valid*:

- Sebagai tanda bahwa pengguna, sebut saja A, menganggap bahwa kunci publik B *valid*, A membuat *certificate* untuk kunci publik B.
- Kunci publik juga *valid* jika memiliki *certificate* yang dibuat oleh pemilik kunci publik *valid* dengan *complete trust*, asalkan rantai *certificate* dari pengguna panjangnya tidak lebih dari 5. Rantai *certificate* contohnya A membuat *certificate* untuk kunci B, B membuat *certificate* untuk kunci C, C membuat *certificate* untuk kunci D. Dengan contoh rantai, jika kunci publik C *valid* dengan *complete trust*, make kunci publik D *valid*.
- Kunci publik dengan tiga *certificate* yang masing-masing dibuat oleh pemilik kunci publik *valid* dengan *marginal trust* juga dianggap *valid*, asalkan rantai *certificate* dari pengguna panjangnya tidak lebih dari 5.

Nilai parameter untuk banyaknya *certificate* dengan *full trust* yang diperlukan, banyaknya *certificate* dengan *marginal trust* yang diperlukan, dan panjang ran-

[1]Ini hanya dalam teori. Dalam prakteknya pemerintah Amerika Serikat tidak dapat mengontrol penggunaan kriptografi.

tai, semua dapat diubah oleh pengguna. Nilai diatas adalah nilai *default*. Pengguna dapat menentukan *trust level* dari suatu kunci publik sebagai pembuat *certificate*. Kunci publik pengguna sendiri dianggap memiliki *complete trust*. Kunci publik *valid* yang lain dapat diberi *trust level*:

- *complete trust*,
- *marginal trust*, atau
- *untrusted*.

Kunci publik yang bukan *valid* otomatis dianggap *untrusted*. *Trust model* yang digunakan PGP dinamakan *web of trust*. Selain *valid*, status kunci publik bisa:

- *marginally valid* jika hanya ada satu *certificate* untuk kunci tersebut dan *certificate* dibuat menggunakan kunci dengan *marginal trust*, atau
- *invalid* jika tidak ada *certificate* untuk kunci tersebut yang dibuat menggunakan kunci dengan *complete trust* atau *marginal trust*.

Certificate berisi antara lain:

- nomor versi PGP,
- kunci publik pemegang *certificate*,
- informasi mengenai pemegang *certificate*, contohnya nama, alamat *email*,
- *self-signature* yaitu kunci publik pemegang *certificate* di*sign* menggunakan kunci privat pemegang *certificate* (ini merupakan bukti kepemilikan kunci privat),
- masa berlaku *certificate*, dan
- *preferred symmetric encryption algorithm*,

tetapi tidak terbatas pada itu. Untuk mendapatkan informasi yang lebih rinci mengenai standard format PGP yang terkini (dinamakan OpenPGP), silahkan membaca [cal07].

GnuPG (Gnu *Privacy Guard*) adalah suatu implementasi *open source* untuk OpenPGP yang banyak digunakan untuk *secure email* dan enkripsi *file* terutama oleh pengguna yang merasa X.509 terlalu rumit. Sesuatu yang menarik dengan GnuPG adalah dukungan terhadap penggunaan *smartcard*.

23.2 X.509

X.509 adalah bagian dari X.500, suatu percobaan yang sangat ambisius dari ITU (International Telecommunication Union) untuk standardisasi *directory services*. Judul dari standard X.509 adalah *The Directory: Public-key and attribute certificate frameworks*. X.500 sendiri dianggap terlalu rumit, namun format *certificate* X.509 dijadikan dasar untuk standard defacto oleh industri, dan kini standard *public key infrastructure* untuk internet yaitu PKIX berdasarkan pada format X.509.

Berbeda dengan "cara" PGP yang cenderung demokratis, "cara" X.509 lebih bersifat struktural hirarkis. Ini adalah warisan dari X.500 yang semula ditujukan untuk membuat direktori pengguna jaringan komputer seperti direktori telpon yang akan dikelola oleh berbagai perusahaan telpon di berbagai negara. Sifat-sifat struktural dalam X.509 antara lain:

- Konsep *relative distinguished name* (RDN) yang cukup rumit dalam membuat *distinguished name* (DN) (identitas).

- Konsep *certificate authority* yang semula dikaitkan dengan RDN.

Konsep *relative distinguished name* (RDN) bertujuan untuk membuat identitas berdasarkan pada posisi dalam suatu hirarki. Hirarki ini diatur berdasarkan negara, propinsi, organisasi, sub-unit organisasi dan seterusnya. Berbagai negara dan organisasi melakukan lokalisasi standard yang kerap disebut *profile* yang menambah rumit konsep RDN. Semula, *certificate authority* (CA) juga dikaitkan dengan RDN, dimana satu CA mempunyai otoritas untuk satu subhirarki. Namun dalam prakteknya ini tidak berjalan.

Karena konsep RDN terlalu rumit dan dapat bertentangan dengan berbagai kepentingan seperti:

- *privacy* (contohnya informasi pribadi), dan

- *confidentiality* (contohnya banyak perusahaan yang tidak menginginkan informasi mengenai struktur organisasinya dipublikasikan),

belum lagi pada kenyataannya dunia tidak sepenuhnya tersusun rapi secara hirarkis tanpa tumpang tindih, akibatnya boleh dikatakan tidak ada yang mengetahui bagaimana cara membuat DN yang "benar." Meskipun demikian, DN tetap merupakan bagian tak terpisahkan dari *certificate* X.509, dan X.509 sudah merupakan *defacto standard*. Berikut adalah berbagai komponen DN yang kerap digunakan:

Komponen	**Penjelasan**
CountryName(C)	Kode negara 2 huruf berdasarkan ISO 3166 (contohnya ID).
Organization(O)	Nama organisasi (contohnya PT Sendiri).
OrganizationalUnit(OU)	Nama unit organisasi (contohnya Sales).
StateOrProvince(SP)	Nama negara bagian atau propinsi (contohnya DKI).
Locality(L)	Nama daerah (contohnya Kebayoran Baru).
CommonName(CN)	Nama pemegang *certificate*, bisa perorangan atau bagian dari organisasi, bahkan nama komputer (contohnya Budi Santoso).

Setiap komponen kecuali CountryName besarnya maksimum 64 karakter, sedangkan CountryName besarnya 2 karakter. Masih banyak komponen DN lainnya yang dapat digunakan, namun komponen yang terpenting sudah masuk dalam daftar diatas. Minimal suatu DN harus mempunyai komponen C dan komponen CN. DN dapat mengidentifikasi orang, unit organisasi atau perangkat, dan digunakan baik untuk identifikasi pemegang *certificate* maupun untuk identifikasi pembuat *certificate*.

Sejak penggunaan internet menjadi dominan, ada alternatif dari DN yang disebut *general name* (GN) yang lebih berorientasi pada internet dibandingkan dengan DN. Komponen GN termasuk antara lain DNS *name*, IP *address*, *email address* dan *uniform resource identifier*. Untuk identifikasi perangkat, penggunaan GN mungkin lebih cocok dibandingkan DN.

Sekarang mari kita lihat isi dari *certificate* X.509. Sebetulnya ada bermacam jenis struktur untuk *certificate* X.509, termasuk

- *certificate* biasa,
- *attribute certificate*,
- *certification request* dan
- *certificate revocation list* (CRL).

Untuk *certificate* biasa, berikut adalah daftar komponen beserta penjelasannya:

Komponen	Penjelasan
Version	Versi X.509 yang digunakan.
SerialNumber	Nomor seri. Setiap *certificate* yang dibuat oleh suatu CA harus mempunyai nomor seri yang berbeda.
SignatureAlgorithm	Algoritma kriptografi yang digunakan untuk menanda-tangan *certificate*.
IssuerName	DN/GN untuk pembuat *certificate* (biasanya CA).
Validity	Masa berlaku *certificate*.
SubjectName	DN/GN untuk pemegang *certificate*.
SubjectPublicKeyInfo	Kunci publik pemegang *certificate*.
Extensions	Untuk berbagai macam informasi tambahan.

Attribute certificate fungsinya adalah sebagai *certificate* untuk berbagai atribut, jadi komponen SubjectPublicKeyInfo diganti oleh Attributes. *Certificate request* fungsinya adalah permintaan *certificate* untuk suatu kunci publik, jadi hanya berisi komponen berikut:

Komponen	Penjelasan
Version	Versi X.509 yang digunakan.
SubjectName	DN/GN untuk pemegang kunci.
SubjectPublicKeyInfo	Kunci publik yang diminta dibuatkan *certificate*.
Extensions	Untuk berbagai macam informasi tambahan.

Certificate revocation list (CRL) dimaksudkan sebagai daftar *certificate* yang dibatalkan oleh suatu CA. Untuk berbagai macam alasan, termasuk kunci yang dicuri, suatu *certificate* dapat dibatalkan oleh CA yang membuatnya. Secara berkala, suatu CA mempublikasikan daftar *certificate* yang dibatalkannya dalam bentuk CRL. CRL dapat digunakan untuk menentukan validitas suatu

certificate. Isi dari CRL adalah sebagai berikut:

Komponen	**Penjelasan**
Version	Versi X.509 yang digunakan.
SignatureAlgorithm	Algoritma kriptografi yang digunakan untuk menanda-tangan *certificate*.
IssuerName	DN/GN untuk CA.
ThisUpdate	Waktu CRL ini dibuat.
NextUpdate	Waktu CRL berikutnya akan dibuat.
UserCertificate	Daftar *certificate* yang dibatalkan.
RevocationDate	Tanggal *certificate* dibatalkan.

Meskipun dalam X.500 *certificate authority* (CA) dikaitkan dengan RDN, kini tidak ada lagi kaitan formal antara CA dengan RDN. *Certificate* bahkan tidak harus dibuat oleh CA. Namun konsep CA masih berguna, meskipun siapa yang patut menjadi CA tergantung pada aplikasi dan/atau situasi. Validitas dari suatu kunci publik ditentukan oleh *certificate* untuk kunci tersebut: jika *certificate* dibuat oleh orang, badan atau CA yang dipercaya oleh seorang pengguna, maka kunci publik dianggap valid oleh pengguna tersebut. Perbedaan utama antara "cara" X.509 dan "cara" PGP dalam menentukan validitas *certificate* adalah dengan X.509, CA biasanya tidak ditentukan oleh pengguna, sedangkan dengan PGP pengguna berperan lebih besar dalam menentukan siapa yang dipercaya sebagai pembuat *certificate* menggunakan mekanisme *web of trust*.

Dari segi teknis, masalah yang dihadapi oleh X.509 dalam manajemen validitas *certificate* jauh lebih rumit dibandingkan masalah yang dihadapi PGP. Ini antara lain karena:

- Berbagai CA dan aktor lainnya berperan aktif dalam manajemen validitas *certificate*.
- Bermacam skenario penggunaan *certificate* X.509 dapat melibatkan aktor yang beragam dan dalam skala yang besar.
- Konsep *certificate revocation* sukar untuk dipraktekkan secara efektif.

Sebagai contoh banyaknya aktor yang berperan dalam manajemen validitas *certificate*, kita bisa lihat begitu banyaknya *certificate authority* yang ada, baik

yang bersifat komersial seperti Verisign, maupun yang non-komersial. Contoh penggunaan *certificate* X.509 yang beragam termasuk untuk keperluan *website authentication* (lihat bagian 20.1), untuk *authentication* kunci publik suatu perangkat dalam jaringan yang menggunakan IPsec untuk VPN (lihat bagian 20.3), dan untuk *secure email* menggunakan S/MIME (lihat bab 21). Sukarnya menggunakan konsep *certificate revocation* secara efektif pernah dijelaskan oleh Peter Gutmann (arsitek dari Cryptlib, lihat bagian 24.3) dengan membuat analogi dengan *relational database* dimana mempublikasikan *certificate* ibarat operasi *prepare* dalam suatu *relational database*, sedangkan *certificate revocation* ibarat operasi *commit* dalam *relational database*:

- Dalam *relational database*, status setelah *prepare* tetapi sebelum *commit* tidak bisa dijamin konsisten.
- Analoginya untuk *certificate*, jika *certificate* telah dipublikasi tetapi *revocation* untuk *certificate* tersebut belum terlihat, maka status dari *certificate* tidak pasti, karena bisa saja *certificate* telah dibatalkan.

Akibat dari berbagai masalah tersebut, tidak ada satu standard protokol untuk manajemen *certificate*, tetapi ada beberapa protokol yang fungsinya beragam.

Salah satu standard protokol untuk manajemen *certificate* adalah standard *Lightweight Directory Access Protocol* (LDAP, lihat [ser06]). Protokol ini digunakan untuk mengakses *certificate store* (tempat penyimpanan *certificate*) yang menggunakan direktori LDAP. Suatu LDAP *server* melayani *client* dengan operasi penyimpanan *certificate* di direktori, penghapusan *certificate* dari direktori, dan pencarian *certificate* dalam direktori. Karena LDAP berorientasi pada X.500 yang terkenal rumit, implementasi LDAP tidak bisa efisien sehingga tidak bisa digunakan dalam skala besar. Untuk skala besar, *relational database* banyak digunakan untuk *certificate store*.

Selain LDAP, berbagai protokol berjenis *request/response* yang digunakan untuk manajemen *certificate* antara lain:

- CMP (*certificate management protocol*).
- OCSP (*online certificate status protocol*).
- TSP (*time-stamp protocol*).

CMP melayani *certificate requests* yaitu pembuatan *certificate* dan pembatalan (*revocation*) *certificate*. Komunikasi dengan CMP *server* dapat melalui http, *email*, atau cara lain. Informasi rinci mengenai CMP dapat dibaca di [ada05]. Karena CMP cukup rumit, sukar untuk menggunakannya dalam skala besar. Akibatnya, beberapa pembuat perangkat *networking* seperti Cisco menggunakan protokol yang lebih sederhana untuk digunakan dalam skala besar yaitu SCEP (*simple certificate enrollment protocol*).

OCSP adalah protokol untuk melayani pemeriksaan *certificate* yang berdasarkan pada concept *revocation* (informasi rinci mengenai OCSP bisa didapat di [mye99]). Jadi OCSP *server* hanya dapat menjawab bahwa *certificate* telah dibatalkan (*revoked*) atau tidak, dan informasi yang digunakan biasanya tidak *up-to-date*. Seperti telah dibahas diatas, Peter Gutmann menganggap solusi ini tidak memuaskan. Peter Gutmann menyarankan penggunaan RTCS (*real-time certificate status protocol*) sebagai *extension* dari OCSP yang bisa menjawab apakah *certificate valid* atau tidak secara *real-time*. Buku ini merekomendasikan penggunaan RTCS seperti yang disarankan oleh Peter Gutmann.

TSP adalah protokol yang digunakan untuk melayani *time-stamping requests*. Suatu *time-stamping authority* (TSA) dapat diimplementasi menggunakan TSP *server* yang melayani *time-stamping requests* dengan membuat *time-stamps* yang ditanda-tangan menggunakan kunci TSA. Informasi rinci mengenai TSP dapat dibaca di [ada01].

Kecuali untuk TSP, *server* untuk berbagai protokol *request/response* perlu menggunakan suatu *certificate store*. Tren saat ini adalah menggunakan *relational database* sebagai *back end* untuk *certificate store* agar dapat digunakan dalam skala besar.

Kita ahiri bagian ini dengan pembahasan penggunaan PKI yang berbasis X.509. PKI untuk komunitas terbuka (dimana tidak ada ikatan formal untuk anggota) biasanya tidak dapat berfungsi efektif. Ini antara lain karena:

- Keperluan dan karakteristik anggota komunitas terbuka beragam.
- Sukar untuk melakukan kontrol terhadap setiap anggota komunitas terbuka dan membuatnya bertanggung jawab atas penggunaan PKI.
- Sukar untuk mendapatkan CA yang dapat dipercaya oleh semua pihak dalam komunitas terbuka.

PKI biasanya hanya dapat berfungsi efektif jika digunakan oleh komunitas tertutup dimana ada ikatan formal untuk anggota, misalnya

- PKI untuk suatu perusahaan.
- PKI untuk suatu pemerintahan atau badan pemerintahan.
- PKI untuk komunitas jaringan bisnis tertentu misalnya jaringan bisnis suku cadang otomotif, dimana setiap anggota terdaftar secara formal dan mempunyai tanggung jawab yang jelas.
- PKI untuk VPN.

23.3 Ringkasan

Di bab ini telah dibahas *public key infrastructure* (PKI) yang pada dasarnya menambahkan fungsi *certificate management* ke fasilitas kriptografi *public key*. Dua standard PKI yang berbeda orientasi yaitu PGP dan X.509 juga dibahas. PKI yang berorientasi pada X.509 sangat sukar untuk dimanfaatkan pada komunitas yang terbuka, tetapi cocok untuk komunitas "tertutup." Sebaliknya, PGP lebih cocok untuk komunitas terbuka.