Bahan Kuliah ke-22

IF5054 Kriptografi

Protokol Kriptografi

Disusun oleh:

Ir. Rinaldi Munir, M.T.

Departemen Teknik Informatika
Institut Teknologi Bandung
2004

22. Protokol Kriptografi

22.1 Protokol

- Protokol: aturan yang berisi rangkaian langkah-langkah, yang melibatkan dua atau lebih orang, yang dibuat untuk menyelesaikan suatu kegiatan.

- Protokol kriptografi: protokol yang menggunakan kriptografi. Orang yang berpartisipasi dalam protokol kriptografi memerlukan protokol tersebut misalnya untuk:
  - berbagi komponen rahasia untuk menghitung sebuah nilai,
  - membangkitkan rangkaian bilangan acak,
  - meyakinkan identitas orang lainnya (otentikasi),
  - dll

- Protokol kriptografi dibangun dengan melibatkan beberapa algoritma kriptografi.

- Sebagian besar protokol kriptografi dirancang untuk dipakai oleh kelompok yang terdiri dari 2 orang pemakai, tetapi ada juga beberapa protokol yang dirancang untuk dipakai oleh kelompok yang terdiri dari lebih dari dua orang pemanaki (misalnya pada aplikasi *teleconferencing*)

- Untuk mendemonstrasikan protokol kriptografi, kita menggunakan nama-nama pemain sebagai berikut:

  Alice : orang pertama (dalam semua protokol)
  Bob : orang kedua (dalam semua protokol)
  Carol : orang ketiga dalam protokol tiga- atau empat-orang
  Dave : orang keempat dalam protokol empat-orang

Eve : penyadap (*eavesdropper*)
Trent : juru penengah (*arbitrator*) yang dipercaya

22.2 Protokol Komunikasi dengan Sistem Kriptografi Simetri.

Protokol 1:

(1) Alice dan Bob menyepakati algoritma kriptografi simetri yang akan digunakan.

(2) Alice dan Bob menyepakati kunci yang akan digunakan.

(3) Alice menulis pesan plainteks dan mengenkripsinya dengan kunci menjadi cipherteks.

(4) Alice mengirim pesan cipherteks kepada Bob.

(5) Bob mendekripsi pesan cipherteks dengan kunci yang sama dan membaca plainteksnya.

- Eve mendengar semua percakapan antara Alice dan Bob pada protokol ini.
  - jika Eve menyadap transmisi pesan pada langkah (4), ia harus mencoba mengkriptanalisis cipherteks untuk memperoleh plainteks tanpa mengetahui kunci.
  - jika ia mendengar pembicaraan pada langkah (1)dan (2), maka ia mengetahui algoritma dan kunci yang digunakan, sehingga ia dapat mendekripsi cipherteks dengan kunci tsb.

- Protokol kriptografi di atas tidak bagus karena kunci harus tetap rahasia sebelum, sepanjang, dan setelah protokol. Langkah (1) dapat dilakukan dalam mode publik, namun langkah (2) harus dilakukan dalam mode rahasia. Sistem kriptografi kunci-publik dapat memecahkan masalah distribusi kunci ini.

22.3 Protokol Komunikasi dengan Sistem Kriptografi Kunci-Publik.

Protokol 2:

(1) Alice dan Bob menyepakati algoritma kriptografi kunci-publik yang akan digunakan.

(2) Bob mengirimi Alice kunci publiknya (kunci publik Bob).

(3) Alice mengenkripsi pesannya dengan kunci publik Bob kemudian mengirimkannya ke Bob

(4) Bob mendekripsi pesan dari Alice dengan kunci privat miliknya (kunci privat Bob).

- Pada umumnya, pengguna di jaringan menyepakati algoritma kriptografi kunci-publik yang digunakan. Setiap pengguna jaringan mempunyai kunci publik dan kunci privat, yang dalam hal ini kunci publik dipublikasikan melalui basisdata yang dapat diakses bersama. Dengan demikian, protokol kriptografi kunci-publik menjadi lebih sederhana sebagai berikut:

Protokol 3:

(1) Alice mengambil kunci publik Bob dari basisdata kunci-publik.

(2) Alice mengenkripsi pesannya dengan kunci publik Bob kemudian mengirimkannya kepada Bob.

(3) Bob mendekripsi pesan dari Alice dengan kunci privat miliknya (kunci privat Bob).

- Eve yang mendengar pembicaraan selama protokol ini akan mendapatkan kunci publik Bob, tetapi Eve tidak dapat mendekripsi cipherteks karena ia tidak mengetahui kunci privat Bob.

- Dalam dunia nyata, sistem kriptografi kunci-publik bukanlah pengganti sistem kriptografi sismetri. Sistem kriptografi kunci-publik tidak digunakan untuk mengenkripsi pesan, melainkan untuk mengenkripsi kunci pada sistem kriptografi simetri.

- Dengan sistem kriptografi kunci-publik, maka pertukaran kunci pada sistem kriptografi simetri dapat dilakukan dengan protokol kriptografi kunci-publik sebagai berikut:

  Protokol 4:

  (1) Bob mengirimi Alice kunci publiknya.

  (2) Alice membangkitkan kunci simetri *K*, mengenkripsikannya dengan kunci publik Bob, lalu mengirimkannya ke Bob,
  $E_B(K)$

  (3) Bob mendekripsi pesan dari Alice dengan menggunakan kunci privatnya untuk mendapatkan kembali kunci simetri *K*,
  $D_B(E_B(K)) = K$

  (4) Baik Alice dan Bob dapat saling berkirim pesan dengan sistem kriptografi simetri dengan menggunakan kunci *K*.

- Dua gabungan sistem kriptografi yang digunakan pada protokol 4 di atas disebut *hybrid cryptosystem* dan kunci sismetri yang dipertukarkan disebut *session key*.

- Dengan protokol 4 di atas, kita katakan bahwa sistem kriptografi kunci-publik berhasil memecahkan masalah manajemen kunci yang sangat penting, yaitu pertukaran kunci.

22.4 Protokol untuk Tanda-tangan Digital (*Digital Signature*)

a. Menandatangani Dokumen dengan Sistem Kriptografi Simetri dan Seorang Juru Penengah.

Alice ingin menandatangani dokumen digital (pesan atau arsip) dan mengirimkannya ke Bob. Ia meminta Trent sebagai juru penengah (misalnya pengacara) antara Alice dan Bob (diperlukan jika sewaktu-waktu ada pertengkaran antara Alice dan Bob). Trent akan memberikan tanda-tangan berupa sertifikasi terhadap dokumen yang dikirim oleh Alice. Sistem kriptografi yang digunakan adalah simetri. Trent memberikan kunci rahasia K_A kepada Alice dan kunci rahasia K_B kepada Bob (K_A dan K_B berbeda).

Protokol 5:

(1) Alice mengenkripsi dokumen dengan K_A dan mengirimkannya kepada Trent.

(2) Trent mendekripsi dokumen dari Alice dengan K_A.

(3) Trent menambahkan pada dokumen yang sudah didekripsi sebuah pernyataan sertifikasi bahwa dia telah menerima dokumen itu dari Alice, kemudian mengenkripsi keseluruhannya dengan K_B.

(4) Trent mengirim cipherteks yang dihasilkan kepada Bob.

(5) Bob mendekripsi cipherteks dengan K_B. Ia membaca dokumen dan sertifikasi dari Trent bahwa Alice yang mengirimkan dokumen tersebut.

- Karakteristik pemberian tanda tangan dengan protokol 5 adalah sbb:
  1. Tanda-tangan (*signature*) pasti otentik, karena Trent adalah juru penegah yang dipercaya, Trent mengetahui bahwa dokumen dari Alice. Sertifikasi dari Trent berlaku sebagai bukti bagi Bob.

  2. Tanda-tangan tidak dapat digunakan lagi untuk dokumen yang lain. Jika Bob menggunakan sertifikasi dari Trent untuk dokumen yang lain, maka kecurangan Bob ini dapat diketahui oleh Trent sbb:
     - Trent meminta dokumen tersebut dari Bob.
     - Trent mengenkripsi dokumen tersebut dengan K_A dan membandingkannya dengan cipherteks dari Alice.
     - Jika hasil enkripsi dokumen dari Bob tidak sama dengan cipherteks dari Alice, maka Bob telah mekakukan kecurangan.

  3. Dokumen yang sudah ditandatangani tidak dapat diubah. Trent dapat membuktikan bahwa dokumen sudah berubah dengan cara yang sama seperti 2 di atas.

  4. Tanda-tangan tidak dapat disangkal. Jika Alice menyangkal bahwa dia yang mengirim dokumen, sertifikasi dari Trent dapat menyanggah sangkalan Alice.

- Protokol 5 di atas tidak praktis karena membutuhkan pihak ketiga (Trent) untuk memberikan sertifikasi keabsahan dokumen dan prosesnya memakan waktu.

b. Menandatangani Dokumen dengan Sistem Kriptografi Kunci-Publik.

Protokol 6:

(1) Alice mengenkripsi dokumen dengan kunci privatnya. Ini sekaligus juga berarti Alice telah memberikan tanda-tangan (*signature*) pada dokumennya.

(2) Alice mengirim dokumen yang terenkripsi kepada Bob.

(3) Bob mendekripsi dokumen dengan kunci publik Alice. Ini sekaligus juga berarti Bob telah memverifikasi tanda-tangan pada dokumen.

- Protokol 6 tidak membutuhkan pihak ketiga (Trent) untuk memberikan tandatangan (Trent hanya diperlukan untuk mensertifikasi bahwa kunci publik Alice memang benar milik Alice).

- Protokol 6 memiliki karakteristik yang sama seperti pada protokol 5.

c. Menandatangani Dokumen dengan Sistem Kriptografi Kunci-Publik dan Fungsi Hash Satu-Arah

Protokol 7:

(1) Alice meringkas dokumennya menjadi *message digest* dengan fungsi *hash* satu-arah.

(2) Alice mengenkripsi *message digest* dengan kunci privatnya. Hasil enkripsinya disertakan (*embedded*) pada dokumen. Ini berarti Alice telah memberi tanda-tangan digital pada dokumennya.

(3) Alice mengirim dokumen yang sudah diberi tanda-tangan digital kepada Bob.

(4) Bob meringkas dokumen dari Alice menjadi *mesaage digest* dengan fungsi *hash* yang sama. Bob mendekripsi tanda-tangan digital yang disertakan pada dokumen Alice. Jika hasil dekripsinya sama dengan *message digest* yang dihasilkan, maka tanda-tangan digital tersebut sah.

- Jika dokumen yang sama ingin ditandatangani oleh dua orang (Alice dan Bob), maka orang ketiga, Carol, dibutuhkan pada proses verifikasi. Protokolnya adalah sebagai berikut:

  Protokol 8:

  (1) Alice memberi tanda-tangan digital pada *message digest* dari dokumen.

  (2) Bob memberi tanda-tangan digital pada *message digest* dari dokumen.

  (3) Bob mengirimkan tanda-tangan digitalnya kepada Alice.

  (4) Alice mengirim dokumen yang sudah diberi tanda-tangan digitalnya dan tanda-tangan digital dari Bob kepada Carol.

  (5) Carol memverifikasi tanda-tangan digital Alice dan tanda-tangan digital Bob (Carol mengetahui kunci publik Alice dan kunci publik Bob).

22.5 Protokol untuk Tanda-tangan Digital dengan Enkripsi

- Protokol ini dapat dianalogikan seperti pengiriman surat yang menggunakan amplop tertutup. Tanda tangan pada surat memberikan bukti kempemilikan, hal ini sama dengan fungsi tanda-tangan digital pada pada dokumen elektrinis. Sedangkan amplop memberikan perlindungan keamanan (*privacy*), hal ini sama dengan fungsi enkripsi pada dokumen.

- Tanda-tangan digital diberikan dengan menggunakan kunci privat pengirim (lihat protokol 6) dan dokumen dienkripsi dengan kunci publik penerima.

Protokol 9:

(1) Alice menandatangi dokumen atau pesan (M) dengan menggunakan kunci privat (A).

$$S_A(M)$$

(2) Alice mengenkripsi dokumen yang sudah ditandatangi dengan kunci publik Bob (B) dan mengirimkannya kepada Bob

$$E_B(S_A(M))$$

(3) Bob mendekripsi cipherteks yang diterima dengan kunci privatnya.

$$D_B(E_B(S_A(M))) = S_A(M))$$

(4) Bob melakukan verifikasi dengan mendekripsi hasil pada langkah 3 dengan menggunakan kunci publik Alice dan sekaligus mendapatkan kembali dokumen yang belum dienkripsi.

$$V_A(\ S_A(M)) = M$$

- Menandatangani dokumen sebelum mengenkripsikannya adalah cara yang alamiah. Dalam kehidupan sehari-hari, kita menulis surat, menandatanganinya, dan memasukkannya ke dalam amplop. Bila Alice memasukkan surat ke dalam amplop, kemudian menandatangani amplop, maka keabsahannya diragukan. Jika Bob memperlihatkan surat Alice tersebut kepada Carol, maka Carol mungkin menuduh Bob berbohong tentang isi surat tersebut.

- Alice tidak harus menggunakan menggunakan kunci publik/kunci privat yang sama untuk enkripsi dan tanda tangan. Alice dapat menggunakan dua pasang kunci: sepasang untuk enkripsi dan sepasang untuk pemberian tanda tangan.

- Misalkan Bob ingin mengkonfirmasi bahwa dia telah menerima dokumen dari Alice. Maka, Bob mengirimkan konfirmasi "tanda terima" kepada Alice. Protokol pengiriman pesan tanda terima adalah sebagai berikut:

Protokol 10:

(1) Alice menandatangi dokumen atau pesan (M) dengan menggunakan kunci privatnya, mengenkripsikannya dengan kunci publik Bob dan mengirimkannya kepada Bob

$$E_B(S_A(M))$$

(2) Bob mendekripsi cipherteks yang diterima dengan kunci privatnya (B), memverifikasi tanda-tangan digital dengan kunci publik Alice dan sekaligus mendapatkan kembali dokumen yang belum dienkripsi.

$$V_A(D_B(E_B(S_A(M)))) = M$$

(3) Bob menandatangani dokumen (M) dengan kunci pribvatnya, mengenkripsikannya dengan kunci publik Alice, dan mengirimkannya ke Alice.

$$E_A(S_B(M))$$

(4) Alice mendekripsi dokumen dengan kunci privatnya dan memverifikasi tanda-tangan digital dengan kunci publik Bob.

$$V_B(D_A(E_A(S_B(M)))) = M'$$

Jika M' yang dihasilkan sama dengan dokumen yang dikirim oleh Alice (M), maka Alice tahu bahwa Bob menerima dokumennya dengan benar.

22.6 Pertukaran Kunci

- Seperti yang sudah disebutkan pada bagian sebelum ini, *session key* adalah kunci simetri yang digunakan untuk mengenkripsi pesan selama berkomunikasi saja.

- Protokol 4 menyebutkan bahwa Alice (atau Bob) mengirimkan kunci publiknya terlebih dahulu sebelum mengenkripsi *session key*. Dalam praktek, kunci publik disimpan di dalam basisdata. Hal ini membuat pertukaran kunci menjadi lebih mudah dengan protokol berikut:

Protokol 11:

(1) Alice mengambil kunci publik Bob dari basisdata.

(2) Alice membangkitkan *session key K*, mengenkripsikannya dengan kunci publik (*PK*) Bob, dan mengirimkannya ke Bob,

$$E_{PK}(K)$$

(5) Bob mendekripsi pesan dari Alice dengan menggunakan kunci rahasianya (*SK*) untuk mendapatkan kembali *session key K*,

$$D_{SK}(E_{PK}(K)) = K$$

(6) Baik Alice dan Bob dapat saling berkirim pesan dengan sistem kriptografi simetri dengan menggunakan kunci *K*.

- Pertukaran kunci dan pengiriman pesan dapat dilakukan bersamaan. Jadi, Alice dan Bob tidak perlu menyelesaikan protokol pertukaran kunci sebelum bertukar pesan.

Protokol 12:

(1) Alice membangkitkan *session key K*, dan mengenkripsi pesan *M* dengan menggunakan *K*,

$$E_K(M)$$

(2) Alice mengambil kunci publik Bob dari basisdata.

(3) Alice mengenkripsi *K* dengan dengan kunci publik Bob,

$$E_B(K)$$

(4) Alice mengirim pesan terenkripsi bersama-sama dengan kunci terbenkripsi kepada Bob,

$E_K(M)$, $E_B(K)$

(5) Bob mendekripsi menggunakan kunci privatnya untuk mendapatkan kembali *session key K*,

$D_B(E_B(K)) = K$

(6) Bob mendekripsi pesan dengan menggunakan kunci *K*,

$D_K(E_K(M)) = M$

- Jika Alice ingin mengirim pesannya tidak hanya kepada Bob, tetapi juga kepada Carol dan Dave, maka protokol pertukaran kunci dan pengiriman pesan dilakukan secara broadcast dengan protokol berikut:

Protokol 12:

(1) Alice membangkitkan *session key K*, dan mengenkripsi pesan *M* dengan menggunakan *K*,

$E_K(M)$

(2) Alice mengambil kunci publik Bob, Carol, dan Dave dari basisdata.

(3) Alice mengenkripsi *K* masing-masing dengan dengan kunci publik Bob, Carol, dan Dave,

$E_B(K)$, $E_C(K)$, $E_D(K)$

(4) Alice mengirim pesan terenkripsi bersama-sama dengan kunci terbenkripsi masing-masing kepada Bob, Carol, dan Dave,

$E_B(K)$, $E_C(K)$, $E_D(K)$, $E_K(M)$,

(5) Hanya Bob, Carol, dan Dave yang dapat mendekripsi kunci *K* dengan menggunakan kunci privatnya masing-masing.

(6) Hanya Bob, Carol, dan Dave yang dapat mendekripsi pesan dengan menggunakan kunci *K*.

- Protokol 12 di atas dapat diimplementasikan pada jaringan *store-and-forward*. Dalam hal ini, *server* memforwardkan pesan terenkripsi dan kunci terenkripsi dari Alice kepada Bob, Carol, dan Dave.

- Diffie-Hellman membuat algoritma pertukaran kunci yang keamanannya didasarkan pada fakta bahwa menghitung logaritma diskrit sangat sulit.
  Mula-mula Alice dan Bob menyepakati bilangan prima yang besar, *n* dan *g*, sedemikian sehingga $g < n$. Bilangan *n* dan *g* tidak perlu rahasia. Bahkan, Alice dan Bob dapat membicarakannya melalui saluran yang tidak aman sekalipun.

  Protokol pertukaran kunci Diffie-Hellman dinyatakan dalam protokol 13 berikut:

  Protokol 13:

  (1) Alice memilih bilangan bulat acak yang besar *x* dan mengirim hasil perhitungan berikut kepada Bob:
  $$X = g^x \bmod n$$

  (2) Bob memilih bilangan bulat acak yang besar *y* dan mengirim hasil perhitungan berikut kepada Alice:
  $$Y = g^y \bmod n$$

  (3) Alice menghitung
  $$K = Y^x \bmod n$$

(4) Bob menghitung

$$K' = X^y \bmod n$$

- Jika perhitungan dilakukan dengan benar, maka $K = K'$. Baik K dan K' sama dengan $g^{xy} \bmod n$. Eve yang mendengarkan semua hal selama protokol berlangsung tidak dapat menghitung kunci K. Ia hanya memiliki informasi n, g, X dan Y, tetapi ia tidak mempunyai informasi nilai x dan y. Untuk mengetahui x atau y, ia perlu melakukan perhitungan logaritma diskrit, yang mana sangat sulit dikerjakan.

- Varian dari algoritma Diffie-Hellman dikemukakan oleh Hughes sebagai berikut:

Protokol 14:

(1) Alice memilih bilangan bulat acak yang besar x dan menghitung:

$$K = g^x \bmod n$$

(2) Bob memilih bilangan bulat acak yang besar y dan mengirim hasil perhitungan berikut kepada Alice:

$$Y = g^y \bmod n$$

(3) Alice mengirim hasil perhitungan berikut kepada Bob

$$X = Y^x \bmod n$$

(4) Bob menghitung

$$z = y^{-1} \quad \text{(balikan } y \text{ dalam modulo } n\text{)}$$
$$K' = X^z \bmod n$$

- Jika perhitungan dilakukan dengan benar, maka $K = K'$. Keuntungan dari protokol ini, Alice dapat langsung mendapatkan kunci rahasia K sebelum interaksi dengan Bob.

Alice dapat mengenkripsi pesannya kepada Bob sebelum protokol pertukaran kunci selesai.

22.7 Otentikasi

1. Otentikasi dengan menggunakan sandi-lewat dan fungsi hash satu-arah.

- Misalkan Alice *log on* ke komputer *host* (misalnya *automatic teller machine*). Bagaimana *host* tahu bahwa yang masuk adalah Alice? Secara tradisionil, sandi-lewat (*password*) digunakan untuk otentikasi.

  Host tidak perlu menyimpan sandi-lewat, ia hanya perlu menyimpan nilai *hash* dari sandi-lewat dengan fungsi *hash* satu-arah. Protokol otentikasinya adalah sebagai berikut:

  Protokol 15

  (1) Alice mengirim sandi-lewatnya ke *host*.

  (2) *Host* mengkompresi sandi-lewat dengan fungsi *hash* satu-arah.

  (3) Host membandingkan hasil dari fungsi *hash* dengan nilai *hash* yang disimpan sebelumnya di dalam tabel (basisdata).

- Kelemahan otentikasi dengan protokol 15 ini adalah rentan terhadap serangan *dictionary attack*. Misalkan Mallory (seorang penyerang aktif yang sangat dengki) berhasil meng-*hack* komputer *host* dan mencuri tabel data sandi-lewat yang sudah dikompres dengan fungsi hash satu-arah. Selanjutnya Mallory menggunakan kamus yang berisi 1.000.000 sandi-

lewat yang sangat umum dipakai orang (nama jalan, tanggal kelahiran, nama anak, dsb). Ia mengkompres seluruh *entry* di dalam kamus dengan fungsi *hash* satu-arah dan menyimpan hasilnya. Kemudian ia membandingkan tabel data sandi-lewat yang dicuri dari *host* dengan hasil *hash* terhadap isi kamus, dan melihat kecocokannya.

- Untuk membuat *dictionary attack* lebih sulit, sistem keamanan komputer biasanya menambahkan garam (*salt*). *Salt* adalah rangkaian bit yang dibangkitkan secara acak dan disambungkan dengan sandi-lewat. Kemudian sandi-lewat yang sudah disambung dengan *salt* dikompres dengan fungsi *hash* dan hasilnya disimpan di dalam tabel. Semakin panjang *salt* semakin bagus. Sistem UNIX menggunakan *salt* 12-bit.

2. Otentikasi dengan menggunakan sistem kriptografi kunci-publik.

- *Host* menyimpan tabel yang berisi kunci publik semua pengguna. Setiap pengguna memiliki kunci rahasia yang bersesuaian dengan kunci publiknya. Protokol otentikasinya adalah sebagai berikut:

Protokol 16

(1) *Host* mengirimi Alice sebuah *string* acak.

(2) Alice mengenkripsi string dengan kunci rahasianya dan mengirimkannya kembali ke *host* beserta *user-id*-nya.

(3) *Host* mencari kunci publik Alice berdasarkan *user-id* yang diberikan dan mendekripsi cipherteks dari Alice dengan kunci publik tersebut.

(4) Jika hasil dekripsi sama dengan string yang semula dikirim oleh *host*, maka *host* mengizinkan Alice mengakses sistem.