Bahan Kuliah IF4020 Kriptografi

Protokol Kriptografi

Oleh: Rinaldi Munir

Program Studi Teknik Informatika
STEI-ITB
2023

Protokol

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

- Protokol kriptografi adalah protokol yang menggunakan algoritma kriptografi.

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

Contoh-contoh protokol kriptografi

1. Secure Socket Layer (SSL)
2. IPSec (Internet Protocol Security)
3. Kerberos
4. Protokol pertukaran kunci Diffie-Hellman
5. Transport Layer Security (TLS)

- 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 pemankai (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

Protokol komunikasi pesan dengan sistem kriptografi simetri

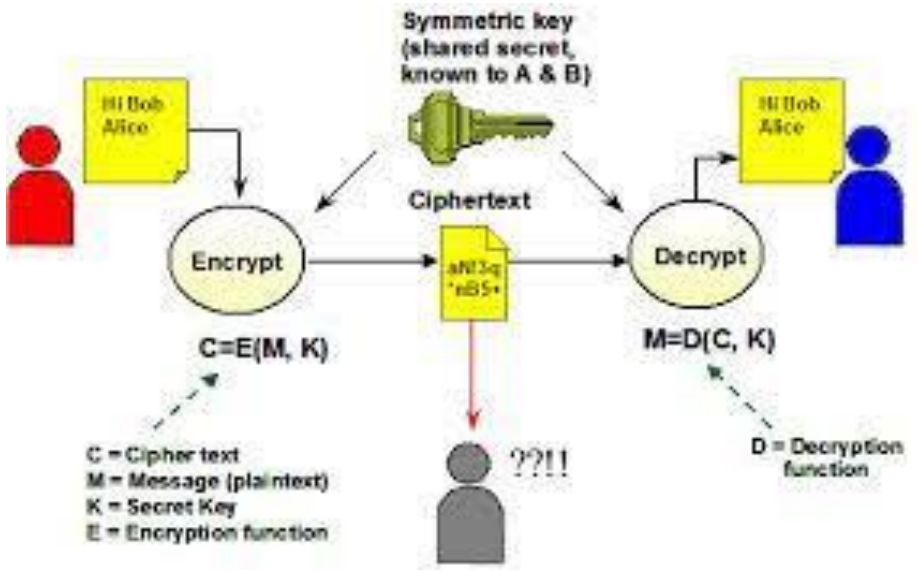
(1) Alice dan Bob menyepakati algoritma kriptografi simetri yang akan digunakan.

(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.



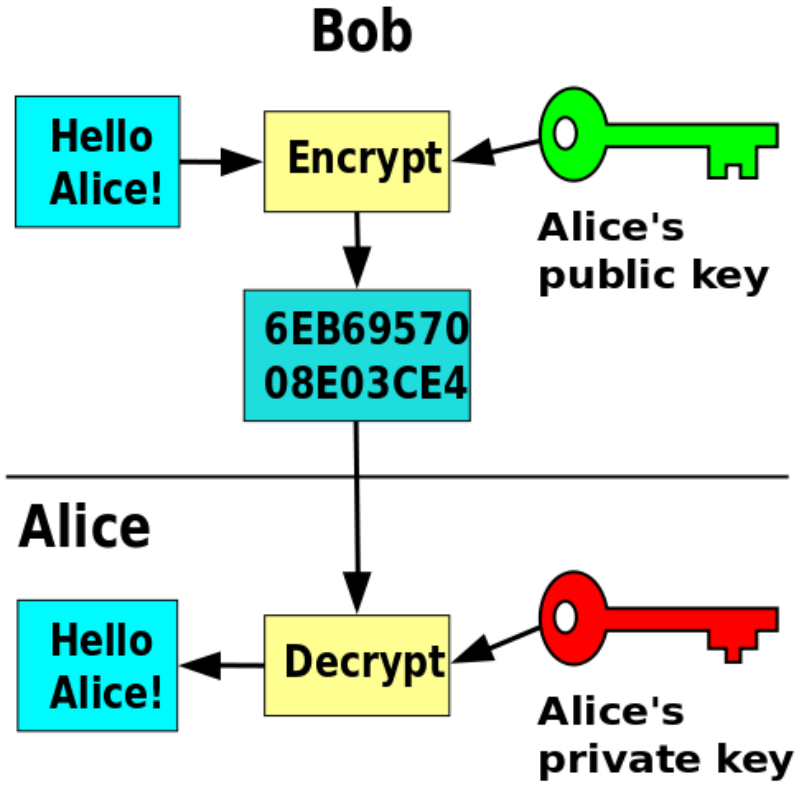
Protokol komunikasi pesan dengan sistem kriptografi kunci-publik

(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).



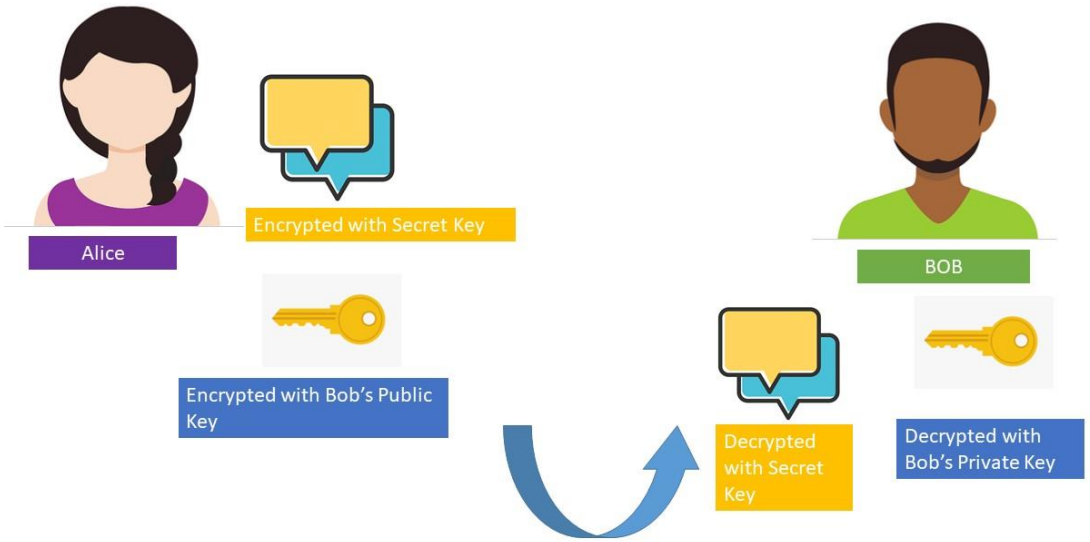
Protokol pertukaran kunci sesi (simetri) - (tanpa basis data)

(1) Bob mengirimi Alice kunci publiknya.

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

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

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



Protokol pertukaran kunci sesi (simetri) (dengan basisdata)

(1) Alice mengambil kunci publik Bob dari basisdata.

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

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

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

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

Protokol pertukaran kunci sesi (simetri) (bersamaan dengan mengirim pesan)

(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*,

$D_K(E_K(M)) = M$

Protokol pertukaran kunci Diffie-Hellman

(1) Alice memilih bilangan bulat acak yang besar *a* dan mengirim hasil perhitungan berikut kepada Bob:

$$A = g^a \bmod n$$

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

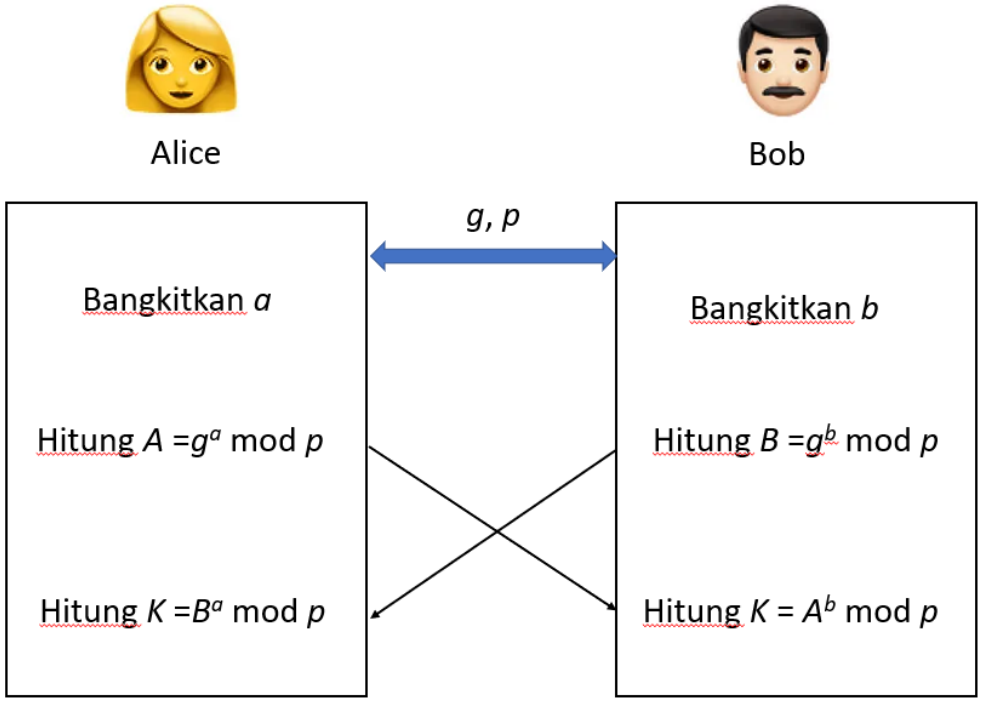
$$B = g^b \bmod n$$

(3) Alice menghitung

$$K = B^a \bmod n$$

(4) Bob menghitung

$$K = A^b \bmod n$$



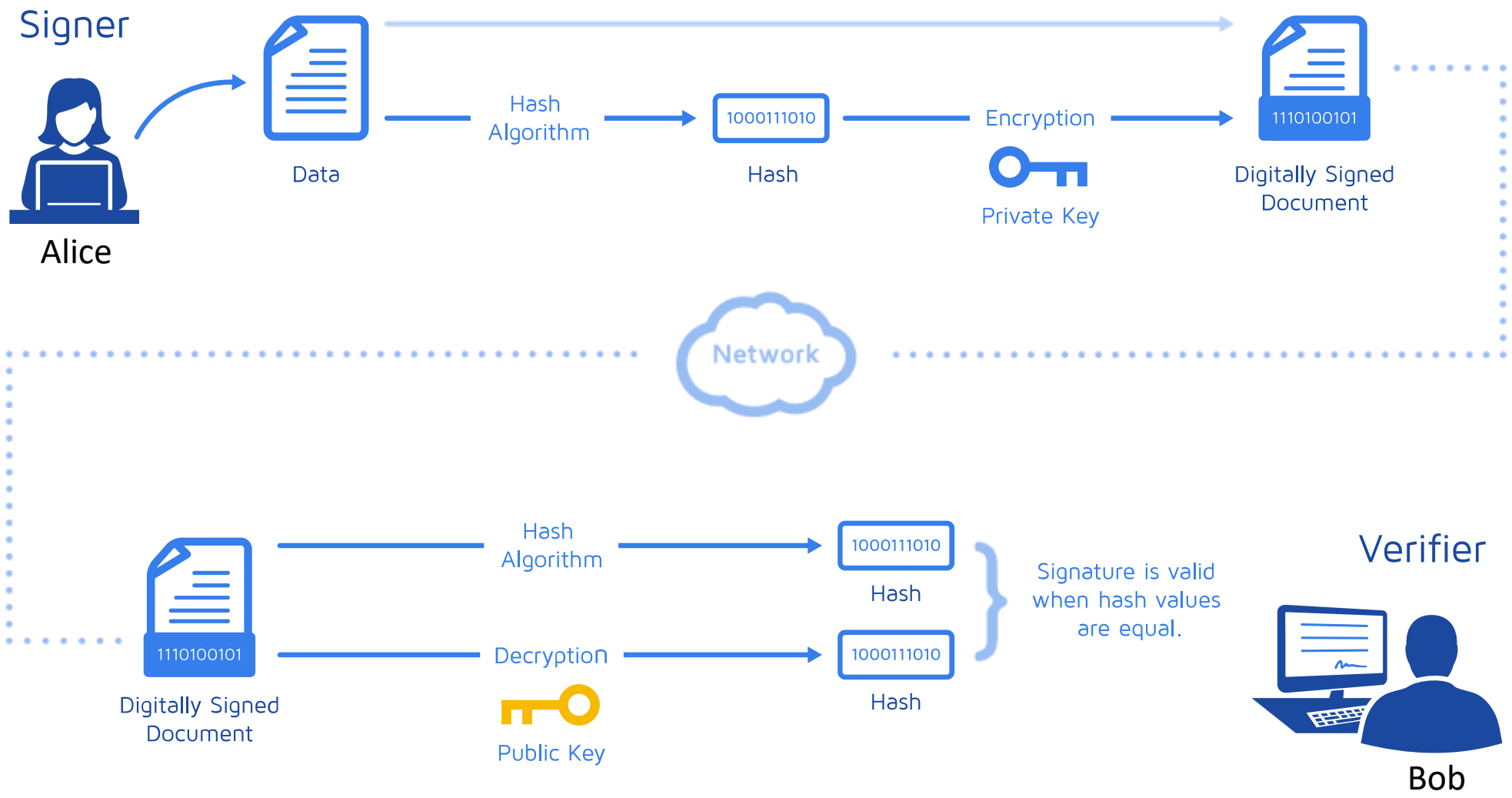
Protokol tanda-tangan digital (2 orang)

(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.



Sumber: https://origin.docusign.com/how-it-works/electronic-signature/digital-signature/digital-signature-faq

Protokol tanda-tangan digital (3 orang)

(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).

Protokol enkripsi plus tanda-tangan

(1) Alice menandatangi dokumen atau pesan (*M*) dengan menggunakan kunci privatnya (*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 (*b*).

$D_b(E_B(S_a(M))) = S_a(M)$

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

$V_A(\ S_a(M)) = M$

Protokol konfirmasi “tanda-terima” pesan

(1) Alice menandatangi dokumen atau pesan (*M*) dengan menggunakan kunci privatnya (*a*), mengenkripsikannya dengan kunci publik Bob (*B*) 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 (*A*) 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 privatnya (*b*), mengenkripsikannya dengan kunci publik Alice (*A*), dan mengirimkan hasilnya kepada Alice.

$$E_A(S_b(M))$$

(3) Alice mendekripsi dokumen dengan kunci privatnya (*a*) dan memverifikasi tanda-tangan digital dengan kunci publik Bob (*B*).

$$V_B(D_a(E_A(S_b(M)))) = M'$$

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

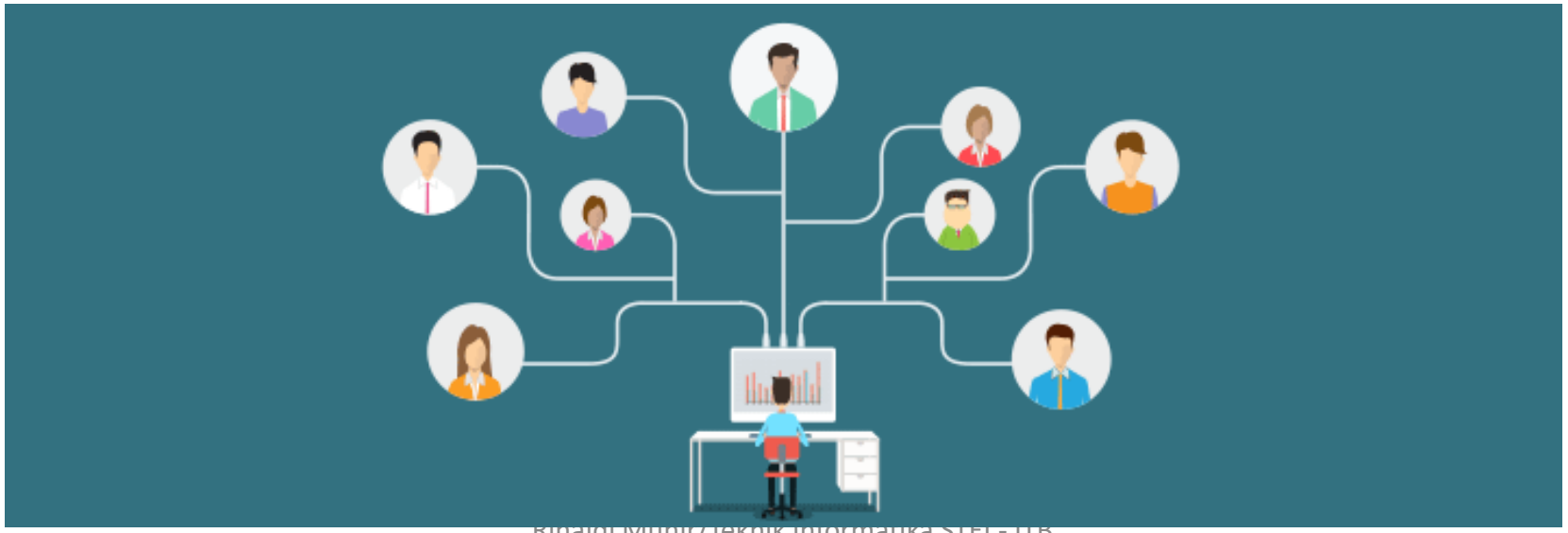
Protokol otentikasi kata-sandi (*password*)

Otentikasi dengan menggunakan kata-sandi dan fungsi hash satu-arah.

(1) Alice mengirim kata-sandi kepada *host*.

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

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



- Kelemahan protocol di atas: rentan terhadap serangan *dictionary attack*

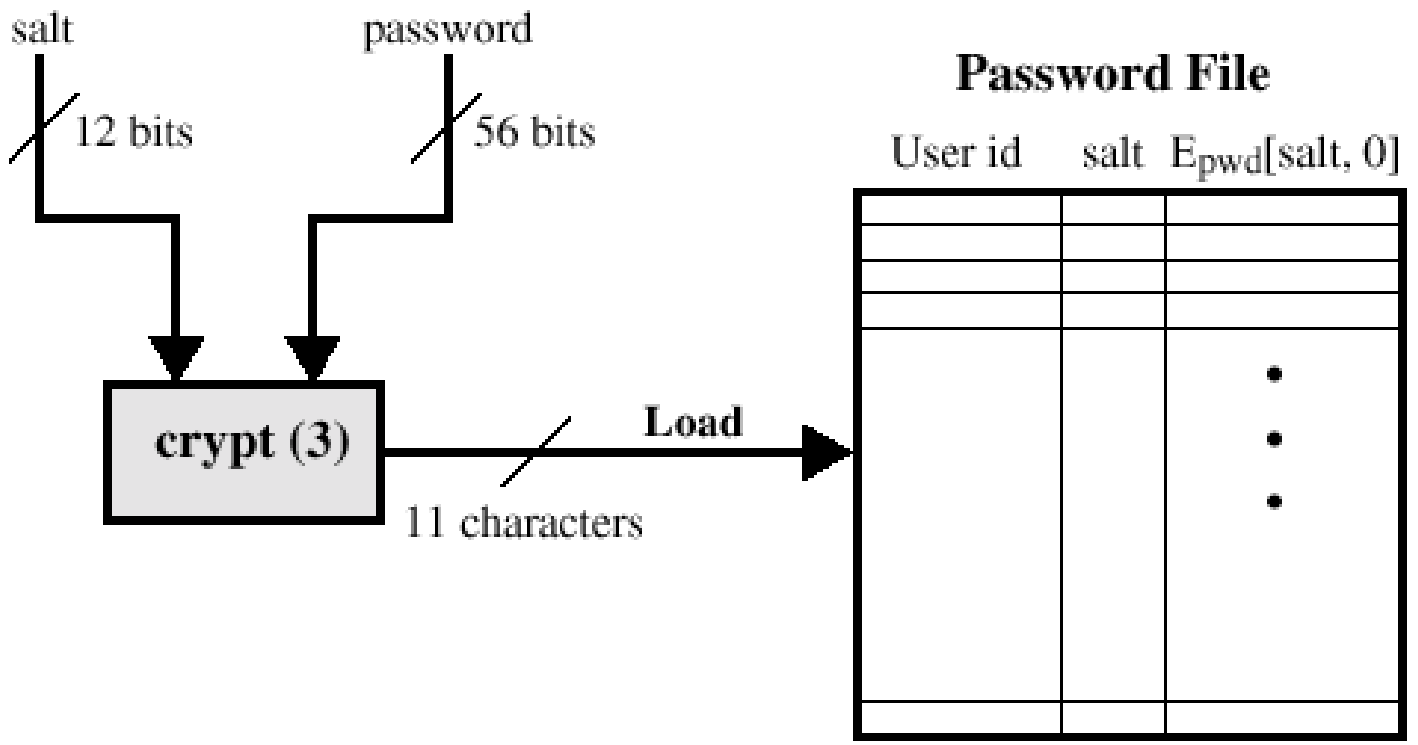
- Untuk membuat *dictionary attack* lebih sulit, sistem keamanan komputer biasanya menambahkan garam (*salt*).

- *Salt* adalah rangkaian bit yang dibangkitkan secara acak dan disambungkan dengan kata-sandi.

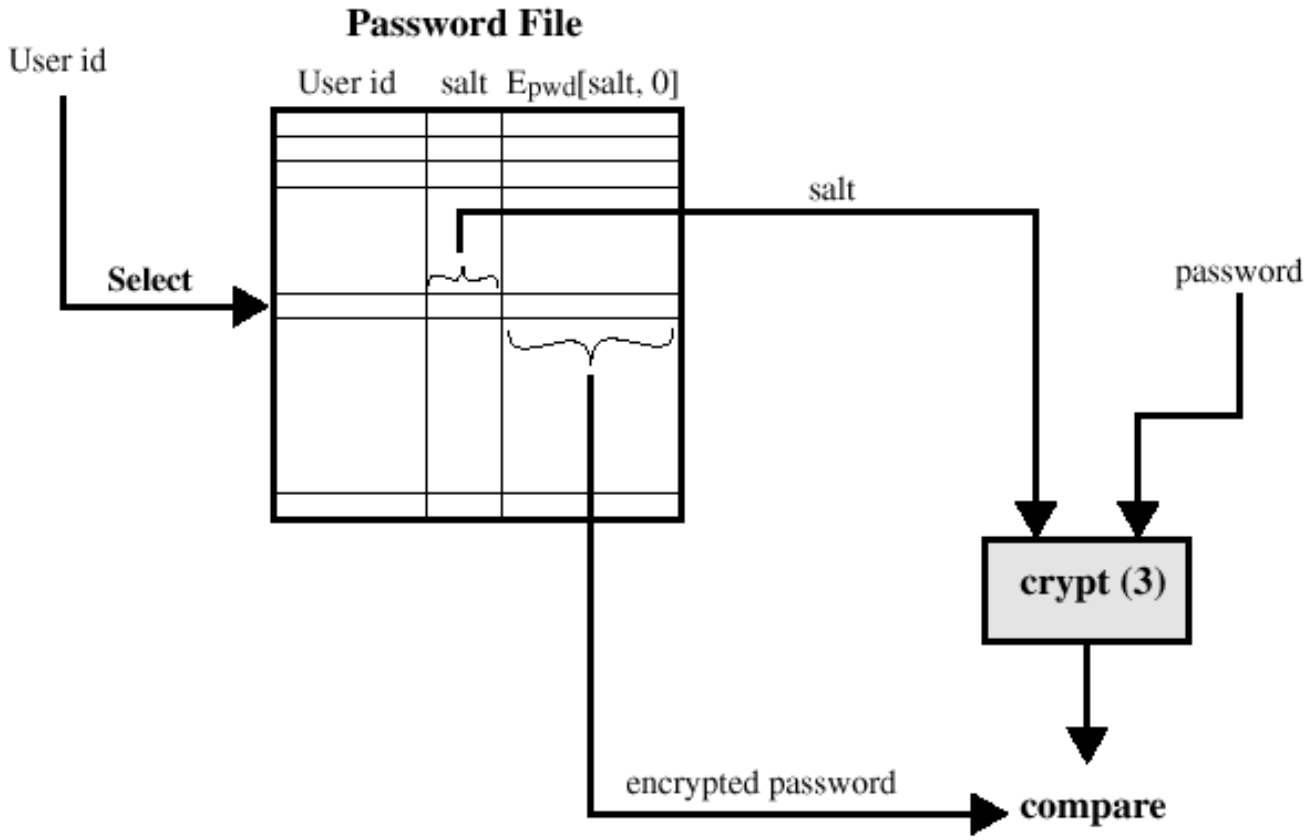
- Kemudian kata-sandi yang sudah disambung dengan *salt* dikompres dengan fungsi *hash* dan hasilnya disimpan di dalam tabel.

- Semakin panjang *salt* semakin bagus.

Skema kata-sandi di dalam UNIX (*salt* = 12 bit)



Loading a new password



Verifying a password file

Otentikasi dengan menggunakan sistem kriptografi kunci-publik .

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

(2) Alice mengenkripsi string dengan kunci privatnya dan mengirimkannya kembali kepada *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.