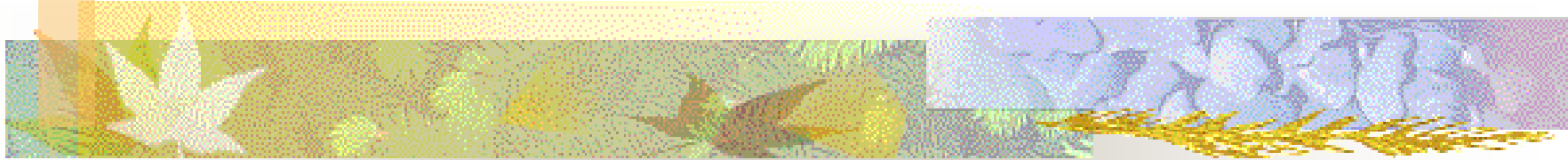


Tantangan Digital



Bahan Kuliah
IF4020 Kriptografi



Review materi awal

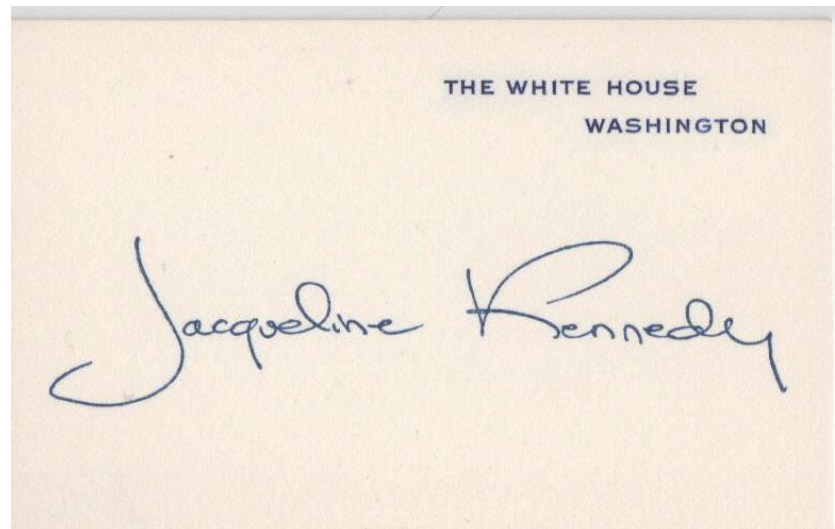
- Aspek keamanan yang disediakan oleh kriptografi:
 1. Kerahasiaan pesan (*confidentiality/secretcy*)
 2. Otentikasi (*authentication*).
 3. Keaslian pesan (*data integrity*).
 4. Anti-penyangkalan (*nonrepudiation*).
- Aspek 1 diselesaikan dengan enkripsi/dekripsi
- Aspek 2 s/d 4 diselesaikan dengan tanda-tangan digital (*digital signature*).




Tanda-tangan Digital

- Sejak zaman dahulu, tanda-tangan sudah digunakan untuk otentikasi dokumen cetak.
- Tanda-tangan mempunyai karakteristik sebagai berikut:
 - Tanda-tangan adalah bukti yang otentik.
 - Tanda tangan tidak dapat dilupakan.
 - Tanda-tangan tidak dapat dipindah untuk digunakan ulang.
 - Dokumen yang telah ditandatangani tidak dapat diubah.
 - Tanda-tangan tidak dapat disangkal(*repudiation*).

- Fungsi tanda tangan pada dokumen kertas juga diterapkan untuk otentikasi pada data digital (pesan, dokumen elektronik).
- Tanda-tangan untuk data digital dinamakan **tanda-tangan digital**.
- Tanda-tangan digital bukanlah tulisan tanda-tangan yang di-digitisasi (*di-scan*).



- 
- Tanda-tangan digital adalah nilai kriptografis yang bergantung pada isi pesan dan kunci.
 - Tanda-tangan pada dokumen cetak selalu sama, apa pun isi dokumennya.
 - Tanda-tangan digital selalu berbeda-beda antara satu isi dokumen dengan dokumen lain.

■ Contoh:

Kepada Yth.
Bapak Dekan
Di Tempat

Dengan hormat.

Bersama surat ini saya ingin mengabarkan bahwa nilai skripsi mahasiswa yang bernama Faisal Saleh dengan NIM 13902021 adalah 86,5 atau dalam nilai indeks A. Sidang skripsi sudah dilakukan pada Hari Rabu Tanggal 21 Januari 20 Juli 2005.

Atas perhatian Bapak saya ucapkan terima kasih.

Bandung, 25 Juli 2005

Dosen Pembimbing Skripsi

Ir. Ahmad Agus

-----BEGIN PGP SIGNATURE-----

**iQA/AwUAQnibsbPbxejK4Bb3EQJXvQCg8zN6UL0xnwBTPR5
FfWNt4uxh3AEAn2NC/G2VTUrLpcSyo2l/S/D/+rUI=pZeh**

-----END PGP SIGNATURE-----

Tanda-tangan digital



Dua cara menandatangani pesan:


1. Enkripsi pesan
2. Menggunakan kombinasi fungsi *hash* (*hash function*) dan kriptografi kunci-publik



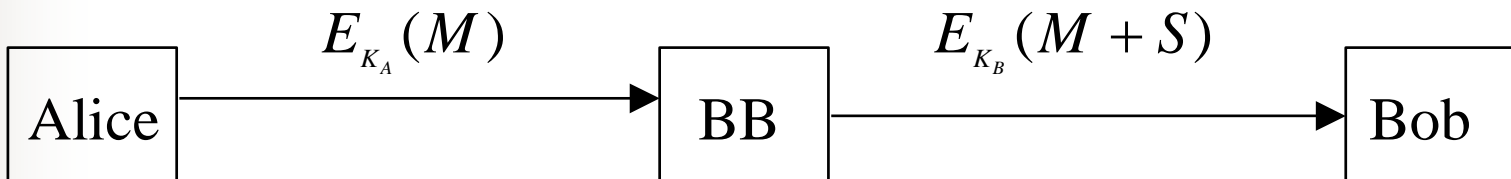
Penandatanganan dengan Cara Mengenkripsi Pesan


a. Menggunakan kriptografi simetri

- Pesan yang dienkripsi dengan algoritma simetri sudah memberikan solusi untuk otentikasi pengirim dan keaslian pesan,
- karena kunci simetri hanya diketahui oleh pengirim dan penerima.
- namun cara ini tidak menyediakan mekanisme untuk anti-penyangkalan.

- 
- Agar dapat mengatasi masalah penyangkalan, maka diperlukan pihak ketiga yang dipercaya oleh pengirim/penerima.
 - Pihak ketiga ini disebut penengah (arbitrase).
 - Misalkan BB (*Big Brothers*) adalah otoritas arbitrase yang dipercaya oleh Alice dan Bob.
 - BB memberikan kunci rahasia K_A kepada Alice dan kunci rahasia K_B kepada Bob.
 - Hanya Alice dan BB yang mengetahui K_A , begitu juga hanya Bob dan BB yang mengetahui K_B .

- Jika Alice bekirim pesan P kepada Bob, maka langkah-langkahnya adalah sebagai berikut:
 1. Alice mengenkripsi pesan M untuk Bob dengan K_A , lalu mengirim cipherteksnya ke BB.
 2. BB melihat bahwa pesan dari Alice, lalu mendekripsi pesan dari Alice dengan K_A .
 3. BB membuat pernyataan S bahwa ia menerima pesan dari Alice, lalu menambahkan pernyataan tersebut pada plainteks dari Alice.
 4. BB mengenkripsi bundel pesan ($M + S$) dengan K_B , lalu mengirimkannya kepada Bob.
 5. Bob mendekripsi bundel pesan dengan K_B . Ia dapat membaca pesan dari Alice (M) dan pernyataan (S) dari BB bahwa Alice yang mengirim pesan tersebut.

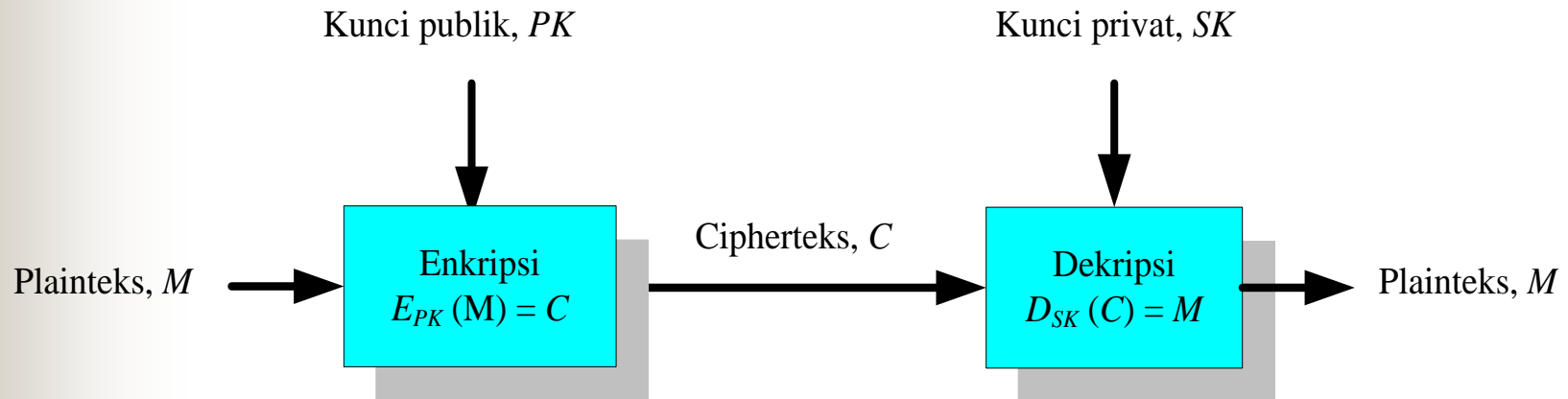


- 
- Jika Alice menyangkal telah mengirim pesan tersebut, maka pernyataan dari BB pada pesan yang diterima oleh Bob digunakan untuk menolak penyangkalan Alice.
 - Bagaimana BB tahu bahwa pesan tersebut dari Alice dan bukan dari Charlie?
 - Karena hanya BB dan Alice yang mengetahui kunci rahasia, maka hanya Alice yang dapat mengenkripsi pesan dengan kunci tersebut.

b. Menggunakan kriptografi kunci-publik

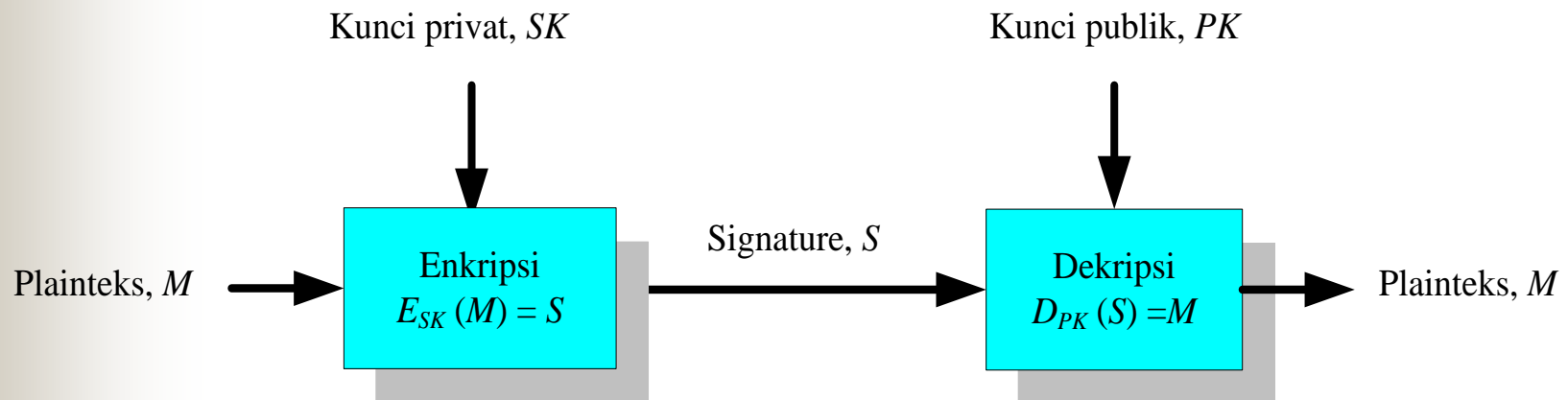
Enkripsi biasa (hanya untuk *secrecy*):

- pesan dienkripsi dengan kunci publik penerima.
- pesan didekripsi dengan kunci privat penerima.
- cara ini tidak memberikan sarana otentikasi karena kunci publik diketahui oleh banyak orang



Enkripsi sebagai tanda-tangan:

- pesan dienkripsi kunci privat pengirim.
- pesan didekripsi pesan dengan kunci publik pengirim.
- dengan cara ini, maka kerahasiaan pesan dan otentikasi keduanya dicapai sekaligus.
- ide ini ditemukan oleh Diffie dan Hellman.



- Proses menandatangani pesan (oleh pengirim):

$$S = E_{SK}(M)$$

- Proses membuktikan otentikasi pesan (oleh penerima):

$$M = D_{PK}(S)$$

Keterangan:

$SK = secret\ key =$ kunci privat pengirim

$PK = public\ key =$ kunci publik pengirim

$E =$ fungsi enkripsi $D =$ fungsi dekripsi

$M =$ pesan semula

$S = signature =$ hasil enkripsi pesan

- Dengan algoritma kunci-publik, penandatanganan pesan tidak membutuhkan lagi pihak penengah (arbitrase).

- 
- Beberapa algoritma kunci-publik dapat digunakan untuk menandatangani pesan dengan cara mengenkripsinya, asalkan algoritma tersebut memenuhi sifat:

$$D_{SK}(E_{PK}(M)) = M \text{ dan } D_{PK}(E_{SK}(M)) = M ,$$

Keterangan:


PK = kunci publik

SK = kunci privat (*secret key*).

E = fungsi enkripsi

D = fungsi dekripsi

M = pesan

- 
- Misalkan M adalah pesan yang akan dikirim. Pesan M ditandatangani menjadi pesan terenkripsi S dengan menggunakan kunci privat (SK) si pengirim,

$$S = E_{SK}(M)$$

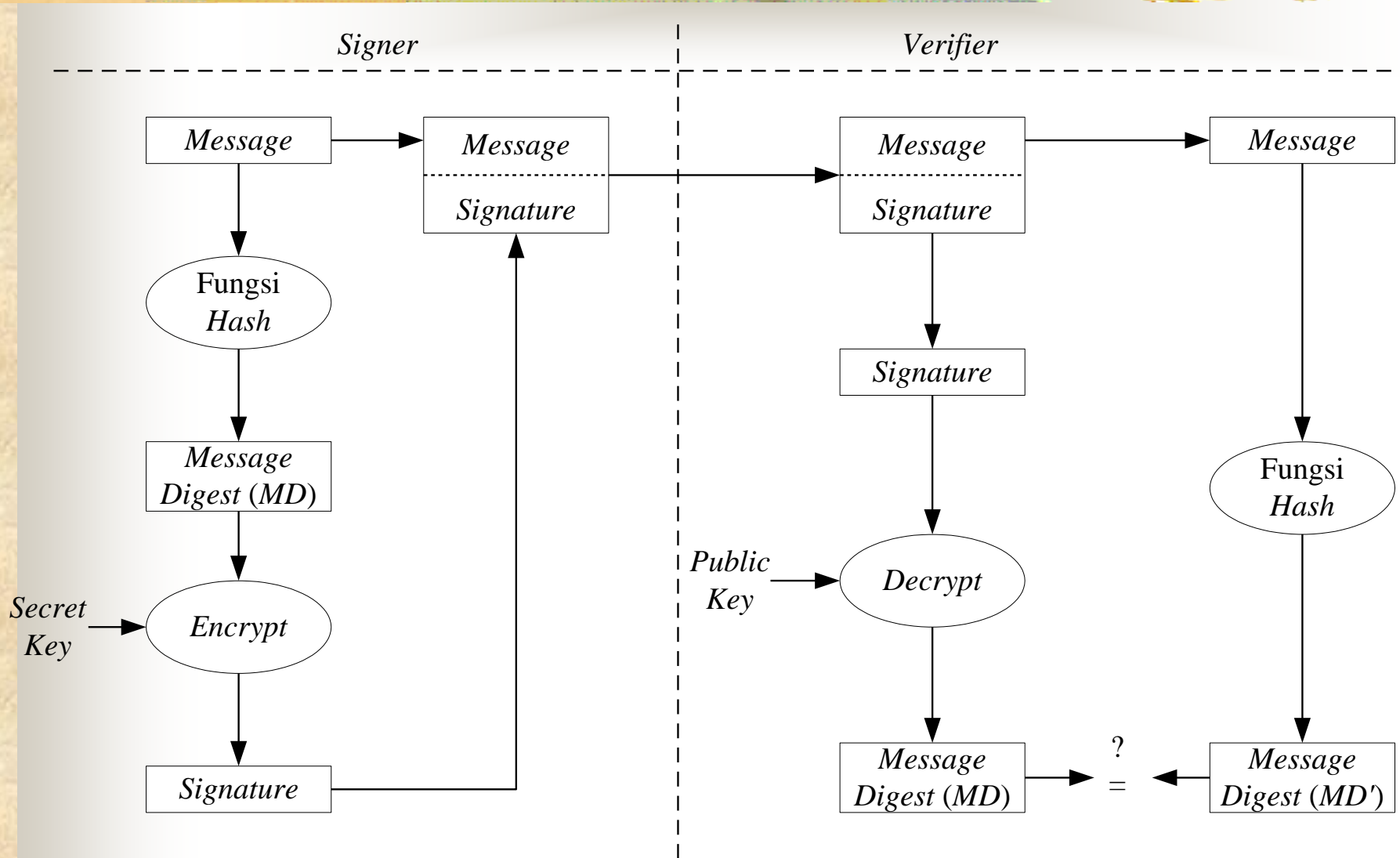
yang dalam hal ini, E adalah fungsi enkripsi dari algoritma kunci-publik.

- Selanjutnya, S dikirim melalui saluran komunikasi.

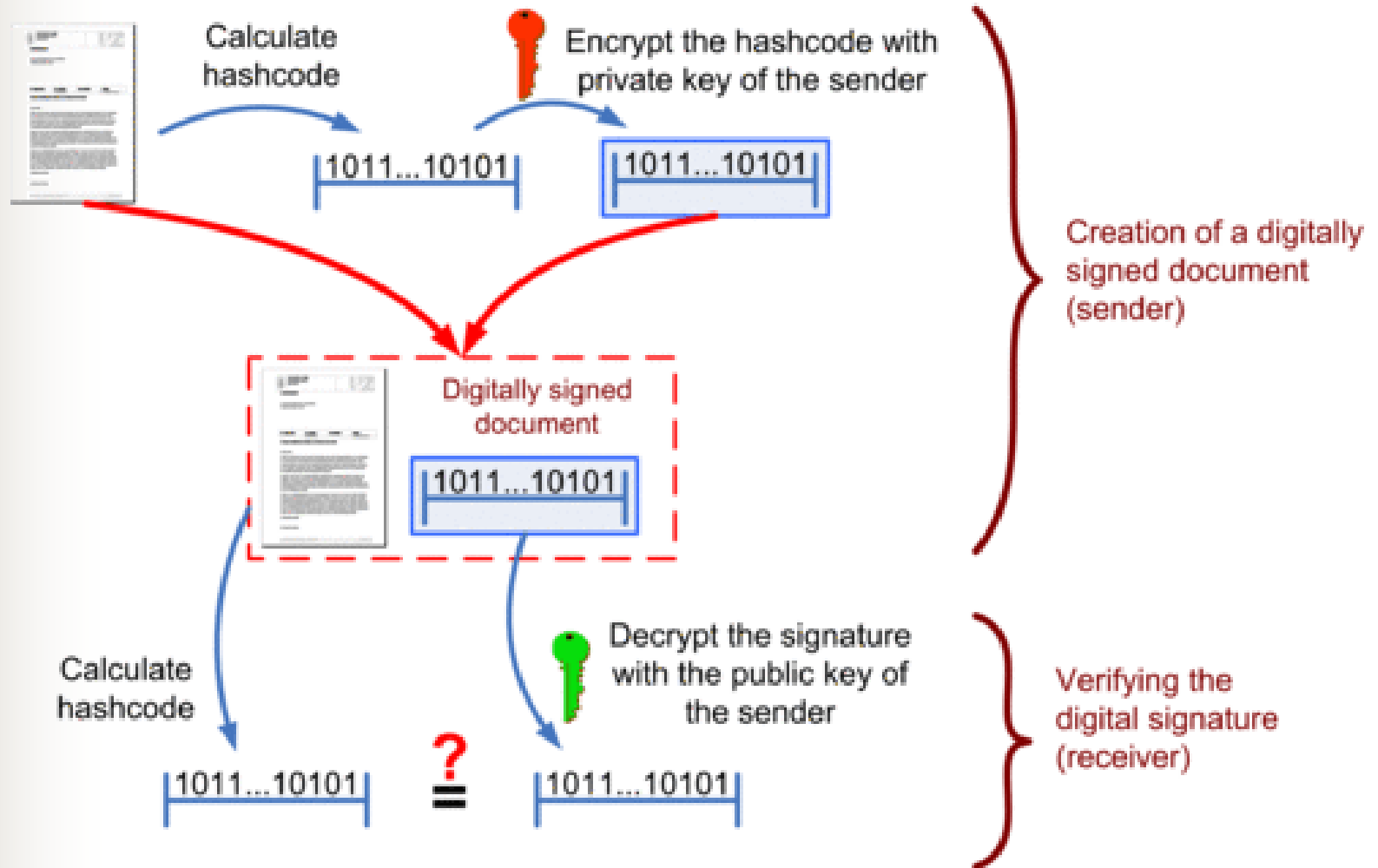


Penandatanganan dengan Menggunakan Kriptografi kunci-publik dan Fungsi *Hash*

- Penandatanganan pesan dengan cara mengenkripsinya selalu memberikan dua fungsi berbeda: kerahasiaan pesan dan otentikasi pesan.
- Pada beberapa kasus, seringkali otentikasi yang diperlukan, tetapi kerahasiaan pesan tidak. Maksudnya, pesan tidak perlu dienkripsikan, sebab yang dibutuhkan hanya keotentikan pesan saja.
- Algoritma kunci-publik dan fungsi hash dapat digunakan untuk kasus seperti ini.



Creating and verifying a digital signature




If the calculated hashcode does not match the result of the decrypted signature, either the document was changed after signing, or the signature was not generated with the private key of the alleged sender.



Keotentikan ini dijelaskan sebagai berikut:

- a. Apabila pesan M yang diterima sudah berubah, maka MD' yang dihasilkan dari fungsi *hash* berbeda dengan MD semula. Ini berarti pesan tidak asli lagi.
- b. Apabila pesan M tidak berasal dari orang yang sebenarnya, maka *message digest* MD yang dihasilkan dari persamaan 3 berbeda dengan *message digest* MD' yang dihasilkan pada proses verifikasi (hal ini karena kunci publik yang digunakan oleh penerima pesan tidak berkoresponden dengan kunci privat pengirim).
- c. Bila $MD = MD'$, ini berarti pesan yang diterima adalah pesan yang asli (*message authentication*) dan orang yang mengirim adalah orang yang sebenarnya (*user authentication*).

- 
- Dua algoritma *signature* yang digunakan secara luas adalah *RSA* dan *ElGamal*.
 - Pada *RSA*, algoritma enkripsi dan dekripsi identik, sehingga proses *signature* dan verifikasi juga identik.
 - Selain *RSA*, terdapat algoritma yang dikhususkan untuk tanda-tangan digital, yaitu *Digital Signature Algorithm (DSA)*, yang merupakan bakuan (*standard*) untuk *Digital Dignature Standard (DSS)*.
 - Pada *DSA*, algoritma *signature* dan verifikasi berbeda

Tanda-tangan dengan algoritma RSA

■ Langkah-langkah pemberian tanda-tangan

1. Pengirim menghitung nilai *hash* dari pesan M yang akan dikirim, misalkan nilai *hash* dari M adalah h .
2. Pengirim mengenkripsi h dengan kunci privatnya menggunakan persamaan enkripsi *RSA*:

$$S = h^{SK} \text{ mod } n$$

yang dalam hal ini SK adalah kunci privat pengirim dan n adalah modulus ($n = pq$, p dan q adalah dua buah bilangan prima).

3. Pengirim mentransmisikan $M + S$ ke penerima

■ Langkah-langkah verifikasi tanda-tangan

1. Penerima menghitung nilai *hash* dari pesan *M* yang akan dikirim, misalkan nilai *hash* dari *M* adalah *h'*.
2. Penerima melakukan dekripsi terhadap tanda-tangan *S* dengan kunci publik si pengirim menggunakan persamaan dekripsi *RSA*:

$$h = S^{PK} \text{ mod } n$$

yang dalam hal ini *PK* adalah kunci privat pengirim dan *n* adalah modulus ($n = pq$, *p* dan *q* adalah dua buah bilangan prima).

3. Penerima membandingkan *h* dengan *h'*. Jika $h = h'$ maka tanda-tangan digital adalah otentik. Jika tidak sama, maka tanda-tangan tidak otentik sehingga pesan dianggap tidak asli lagi atau pengirimnya