Reti di Calcolatori:
Internet, Intranet e Mobile Computing
a.a. 2007/2008

http://www.di.uniba.it/~lisi/courses/reti/reti0708.htm

dott.ssa Francesca A. Lisi
lisi@di.uniba.it

Orario di ricevimento: mercoledì ore 10-12

dib

Sommario della lezione di oggi: La sicurezza nelle reti

Dai fondamenti ...

- che cosa è la sicurezza?
- crittografia
- autenticazione
- integrità e non ripudiabilità del messaggio
- distribuzione delle chiavi e certificazione

... in pratica

- a livello di applicazione: PGP
- a livello di trasporto: SSL, SET
- a livello di rete: IPsec

Cos'è la sicurezza delle reti?

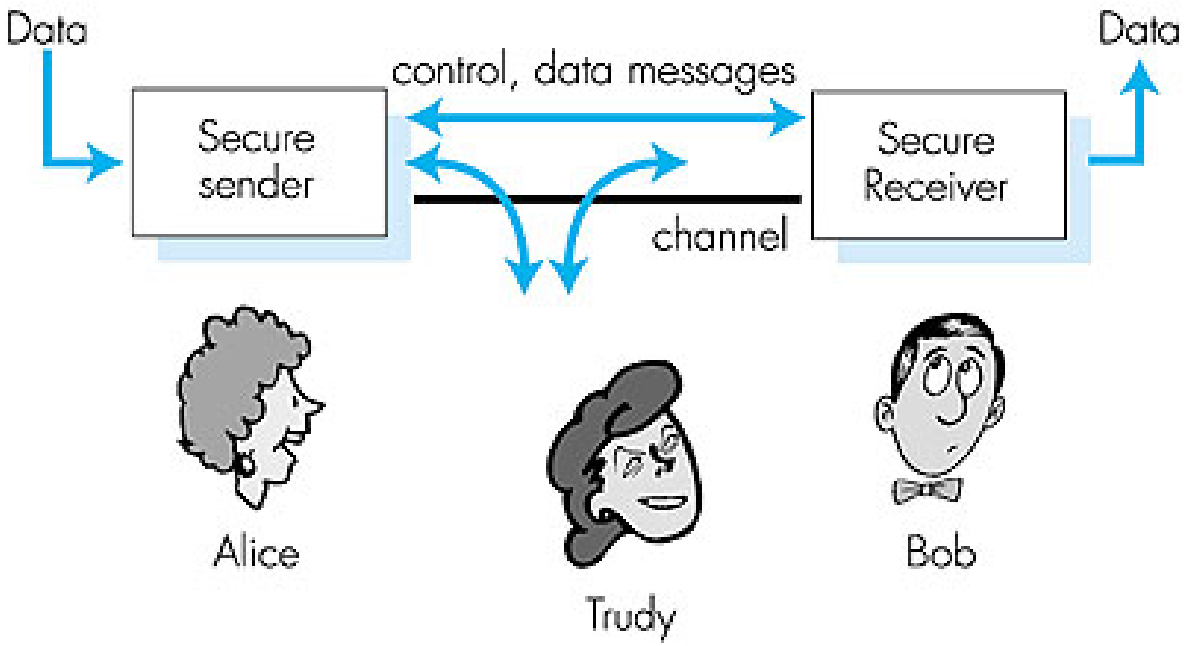
Segretezza: solo il mittente e il destinatario inteso dovrebbero "comprendere" il contenuto di un messaggio

- il mittente cripta il messaggio
- il destinatorio decripta il messaggio

Autenticazione: mittente e destinatario vogliono conferme su identità reciproca

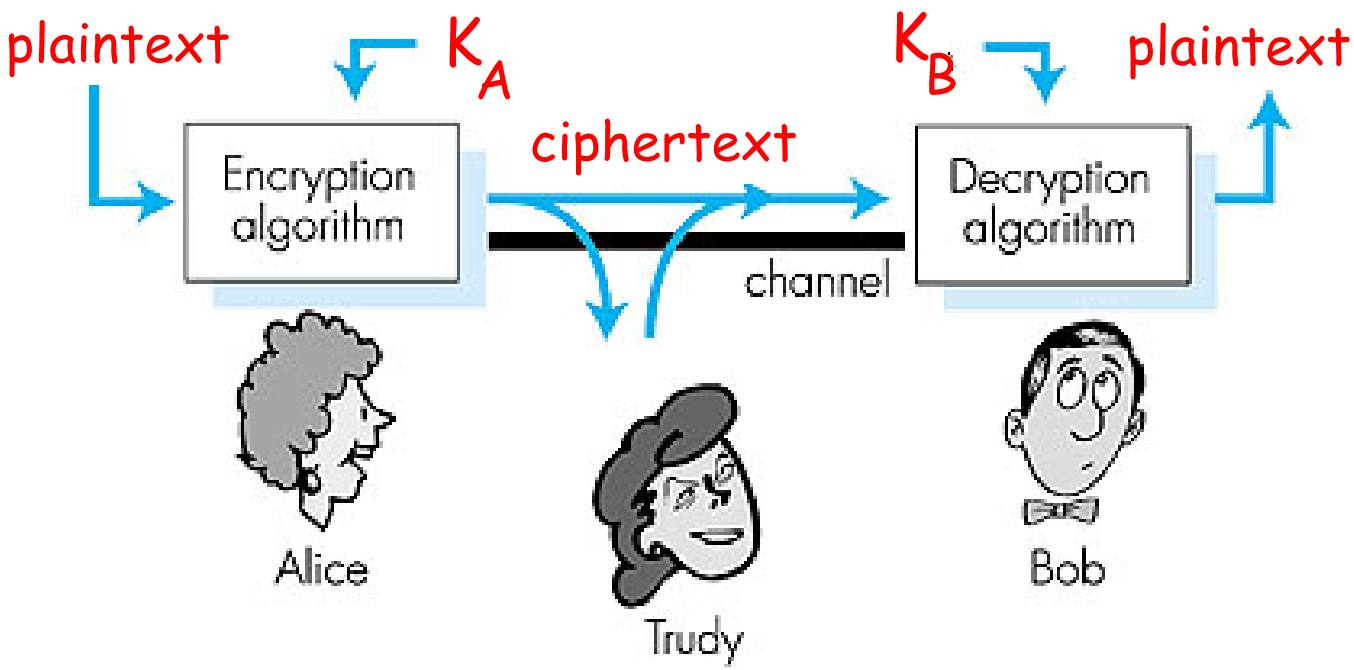
Integrità del messaggio: mittente e destinatorio vogliono assicurare che il messaggio sia inalterato (in transito, o in seguito)

Un esempio di sicurezza nel mondo reale



- Bob e Alice (amanti!) vogliono comunicare "in modo sicuro"
- Trudy (l'intrusa) potrebbe intercettare, cancellare, aggiungere messaggi

Fondamenti di crittografia



- Algoritmi di cifratura e decifratura
- Chiavi di cifratura e decifratura (parametri degli algoritmi)
- Testo viaggia cifrato lungo il canale di comunicazione

Fondamenti di crittografia

Crittografia a chiave simmetrica: mittente e ricevente hanno la medesima chiave, utilizzata sia per la cifratura che per la decifratura

Crittografia a chiave pubblica: mittente e ricevente hanno ciascuno una coppia di chiavi, di cui una pubblica (nota) per la cifratura e una privata (segreta) per la decifratura

[Diffie-Hellman76]

Crittografia a chiave simmetrica

Cifra di sostituzione: sostituire una cosa per un'altra

- cifra monoalfabetica: sostituisci una lettera per un'altra

Testo in chiaro: **abcdefghijklmnopqrstuvwxyz**

Testo cifrato: **mnbvcxzasdfghjklpoiuytrewq**

E.g.: **In chiaro: bob. i love you. alice**
cifrato: nkn. s gktc wky. mgsbc

D: Quanto è difficile infrangere questo semplice codice?:

- brute force
- altro?

Crittografia a chiave simmetrica: DES (Data Encryption Standard)

- US encryption standard [NIST 1993]
- chiave simmetrica a 56 bit, input di testo in chiaro a 64 bit
- Quanto è sicuro?
  - DES Challenge: frase cifrata con chiave a 56 bit ("Strong cryptography makes the world a safer place") decifrata (brute force) in 4 mesi
  - nessun approccio noto di decodica "backdoor"
- rendere DES più sicuro
  - usare tre chiavi sequenzialmente (3-DES) su ogni dato
  - usare concatenamento dei blocchi cifrati

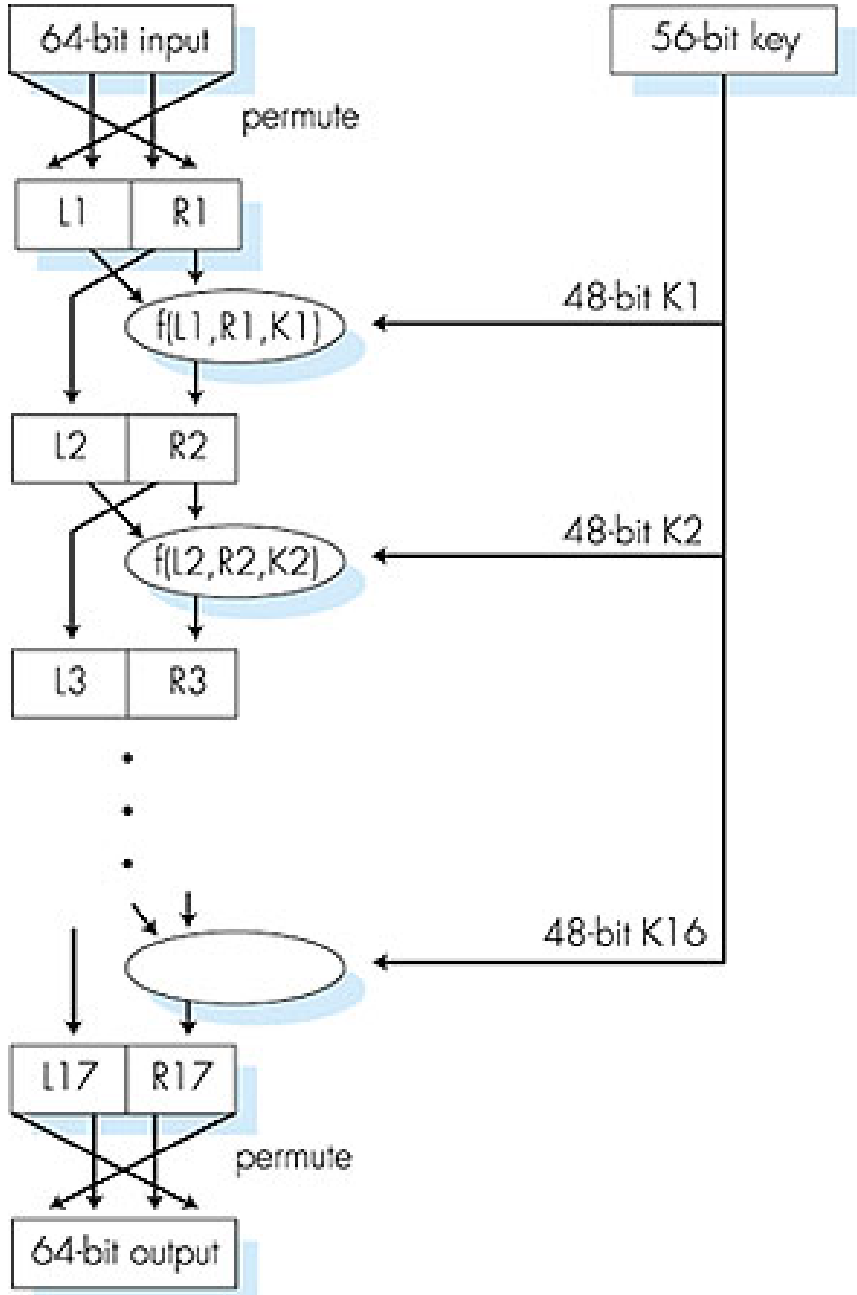
Crittografia a chiave simmetrica: DES

DES operation

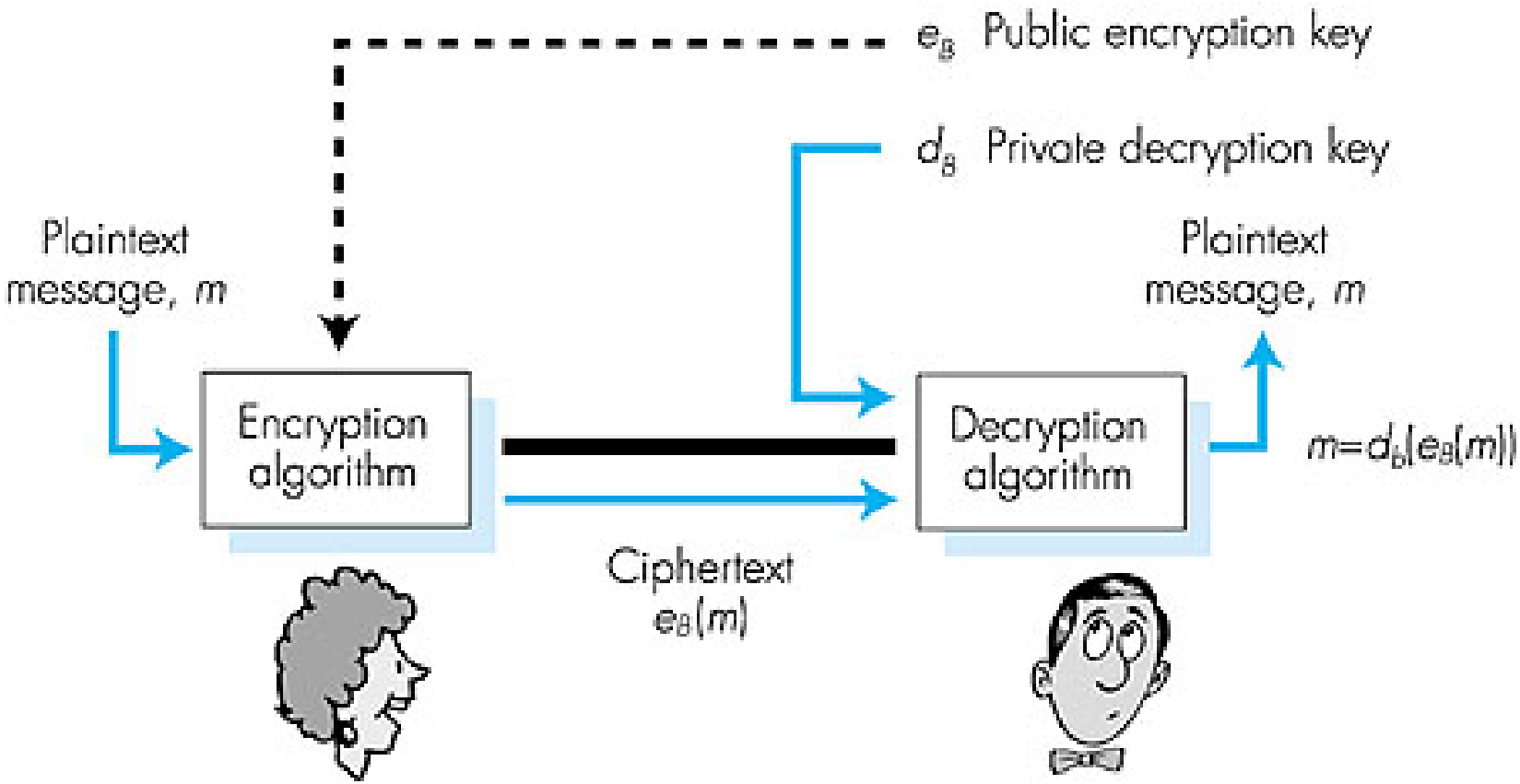
Permutazione iniziale

16 cicli identici di applicazione di funzione, ciascuno dei quali usa 48 bit diversi della chiave

permutazione finale



Crittografia a chiave pubblica



Crittografia a chiave pubblica: proprietà delle chiavi

Due requisiti interrelati:

1. need $d_B(\cdot)$ and $e_B(\cdot)$ such that
   $d_B(e_B(m)) = m$

2. need public and private keys for $d_B(\cdot)$ and $e_B(\cdot)$

RSA: Rivest, Shamir, Adelson algorithm (1978)

Crittografia a chiave pubblica: scelta delle chiavi in RSA

1. Scegliere due grandi numeri primi *p, q.* (e.g., ciascuno di 1024 bit)
2. Calcolare *n = pq, z = (p-1)(q-1)*
3. Scegliere *e (con e<n)* che non abbia fattori comuni con z. (*e, z* sono "relativamente primi").
4. Scegliere *d* tale che *ed-1* sia esattamente divisibile per *z*. (in altri termini: *ed* mod *z = 1*).
5. La chiave *pubblica* è *(n,e)*. La chiave *privata* è *(n,d)*.

Crittografia a chiave pubblica: cifratura e decifratura in RSA

0. Dati (n,e) e (n,d) calcolati come visto prima

1. Per codificare il pattern di bit, m, calcola

   $c = m^e \bmod n$ (i.e., resto se m^e è diviso per n)

2. Per decifrare il pattern di bit ricevuto, c, calcola

   $m = c^d \bmod n$ (i.e., resto se c^d è diviso per n)

Magia! $m = (m^e \bmod n)^d \bmod n$

Crittografia a chiave pubblica: un esempio di RSA

Bob sceglie *p=5, q=7*. Quindi *n=35, z=24*.

e=5 (così *e, z* relativamente primi).
d=29 (così *ed-1* esattamente divisibile per z).

encrypt:

letter	m	m^e	$c = m^e \bmod n$
l	12	1524832	17

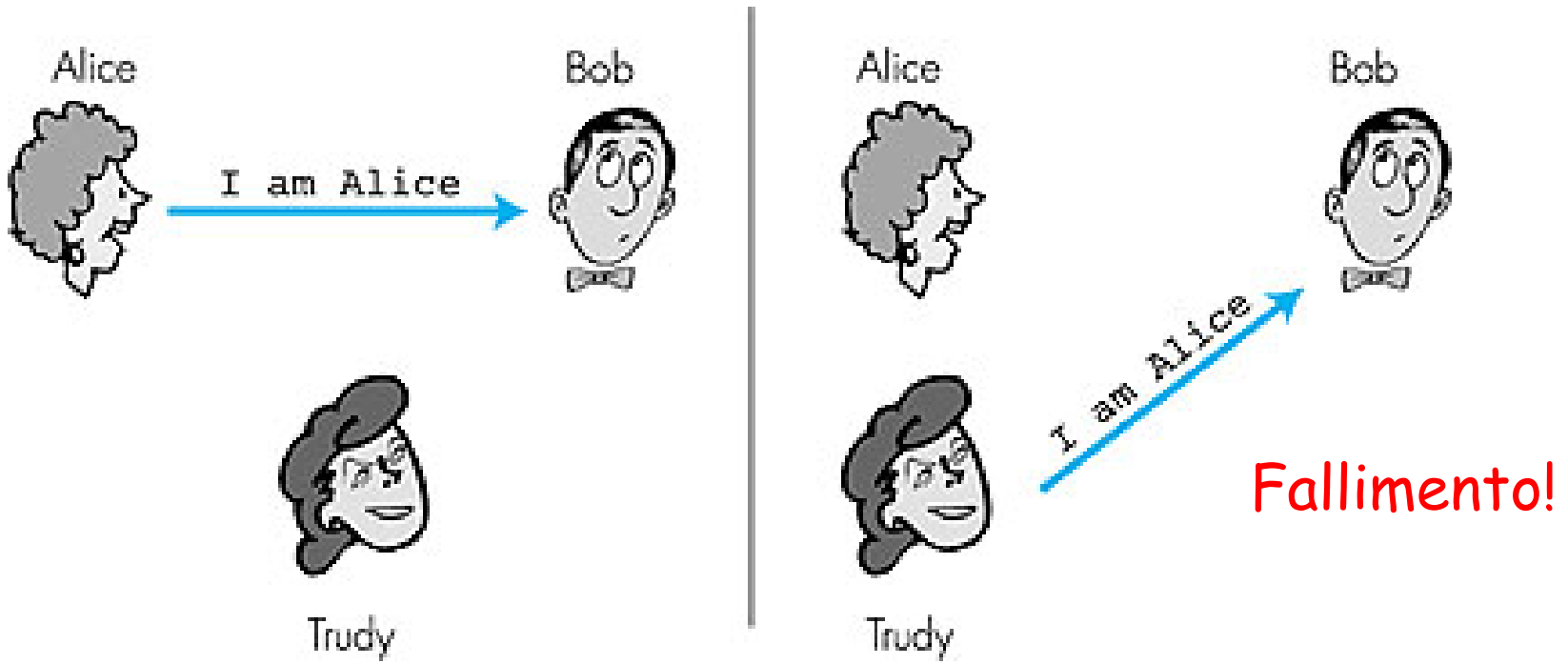
decrypt:

c	c^d	$m = c^d \bmod n$	letter
17	481968572106750915091411825223072000	12	l

Autenticazione: definizione del problema

Bob vuole che Alice gli "provi" la sua identità

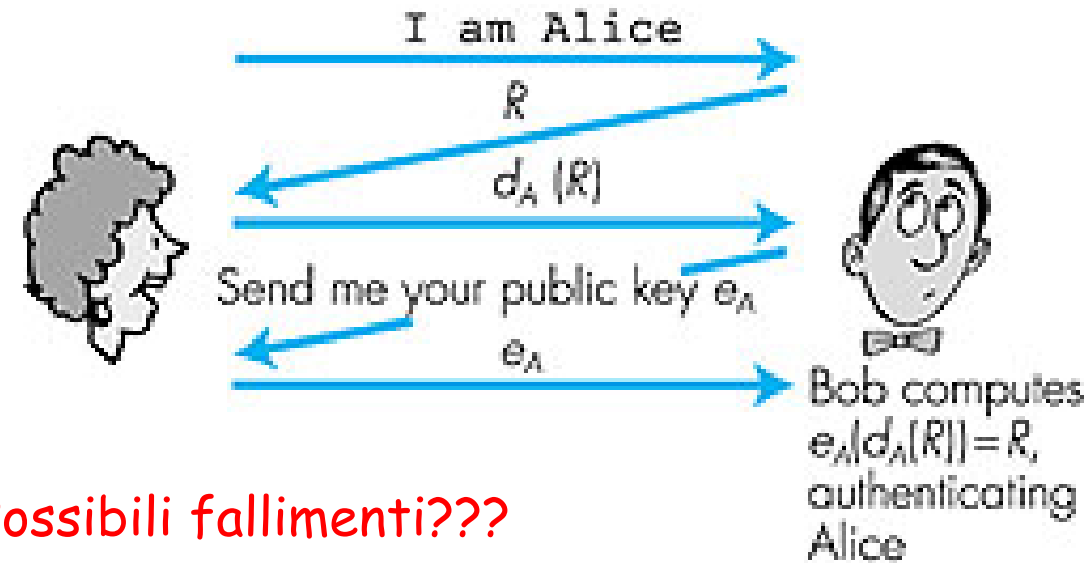
Soluzione non sicura : Alice dice "I am Alice"



Obiettivo: evitare attacchi di playback

Autenticazione: soluzione con crittografia a chiave pubblica

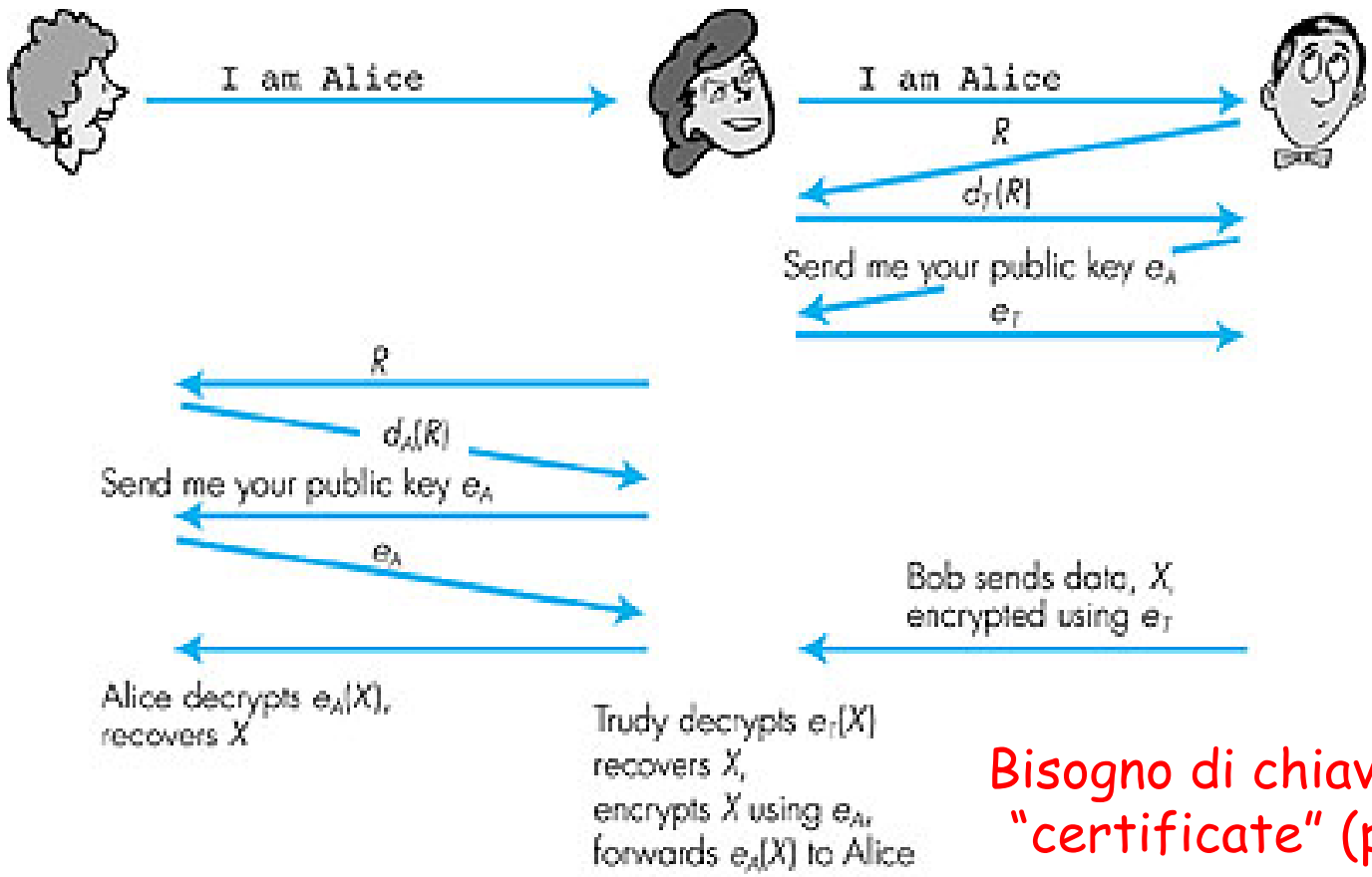
- Per verificare se Alice è "viva", Bob invia ad Alice un numero R, detto nonce perchè usato solo una volta.
- Alice deve restitutire a Bob il numero R criptato con la propria chiave privata ($d_A(R)$)
- Bob autentica Alice applicando la chiave pubblica di Alice al numero $d_A(R)$



Possibili fallimenti???

Autenticazione: attacco man-in-the-middle

Trudy si pone come Alice (nei confronti di Bob) e come Bob (nei confronti di Alice)



Bisogno di chiavi pubbliche "certificate" (più tardi…)!

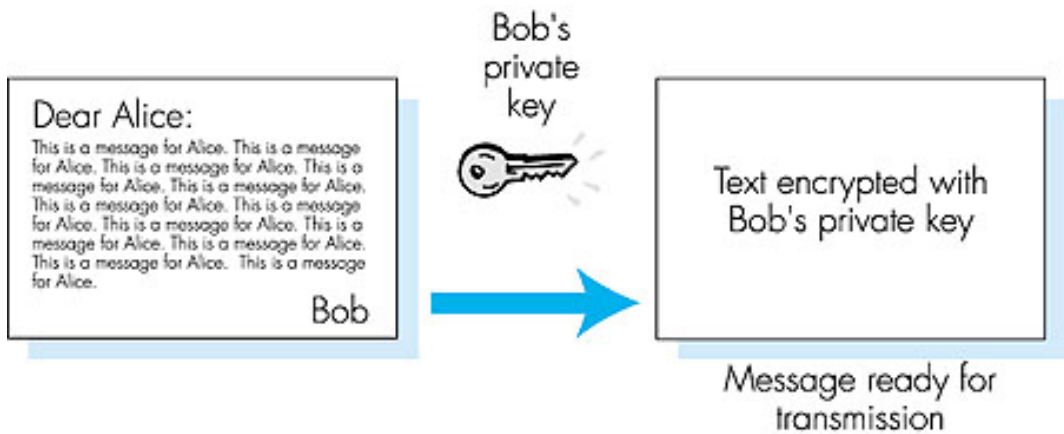
Integrità: firme digitali

Tecnica crittografica analoga alle firme scritte a mano.

- Non ripudiabile: Il mittente (Bob) firma elettronicamente il documento, stabilendo che lui è il possessore/creatore del documento.
- Verificabile, non falsificabile: il ricevente (Alice) può verificare che Bob, e nessun altro, ha firmato il documento.

Semplice firma digitale per il messaggio m:

- Bob codifica m con la sua chiave privata d_B, creando il messaggio firmato, $d_B(m)$.
- Bob invia m e $d_B(m)$ ad Alice.



Integrità: firme digitali (cont.)

- Supponiamo che Alice riceve il msg *m* e la firma digitale $d_B(m)$
- Alice verifica che *m* sia firmato da Bob applicando la chiave pubblica di Bob e_B a $d_B(m)$ *quindi* controlla se $e_B(d_B(m)) = m$.
- Se $e_B(d_B(m)) = m$, chiunque abbia firmato *m* deve aver usato la chiave privata di Bob.

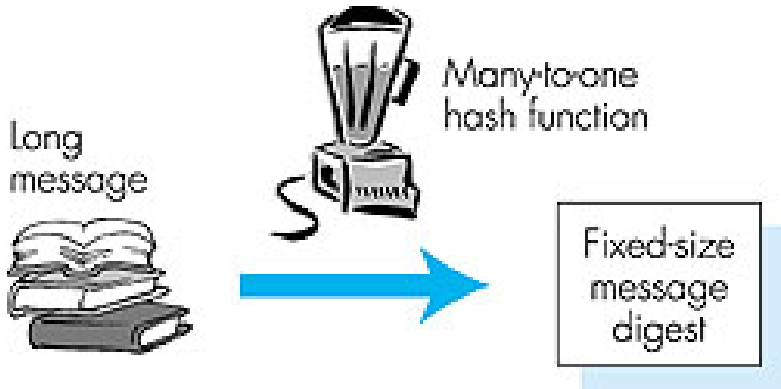
Alice pertanto verifica che:

- Bob ha firmato *m*.
- Nessun altro ha firmato *m*.
- Bob ha firmato *m* ma non *m'*.

Non ripudio:

- Alice può prendere *m*, e la firma $d_B(m)$ per corteggiare e provare che Bob ha firmato *m*.

Integrità: digest del messaggio



E' computazionalmente costoso codificare con chiave pubblica i messaggi lunghi!

Obiettivo: lunghezza fissa, facile da calcolare una firma digitale, "fingerprint"

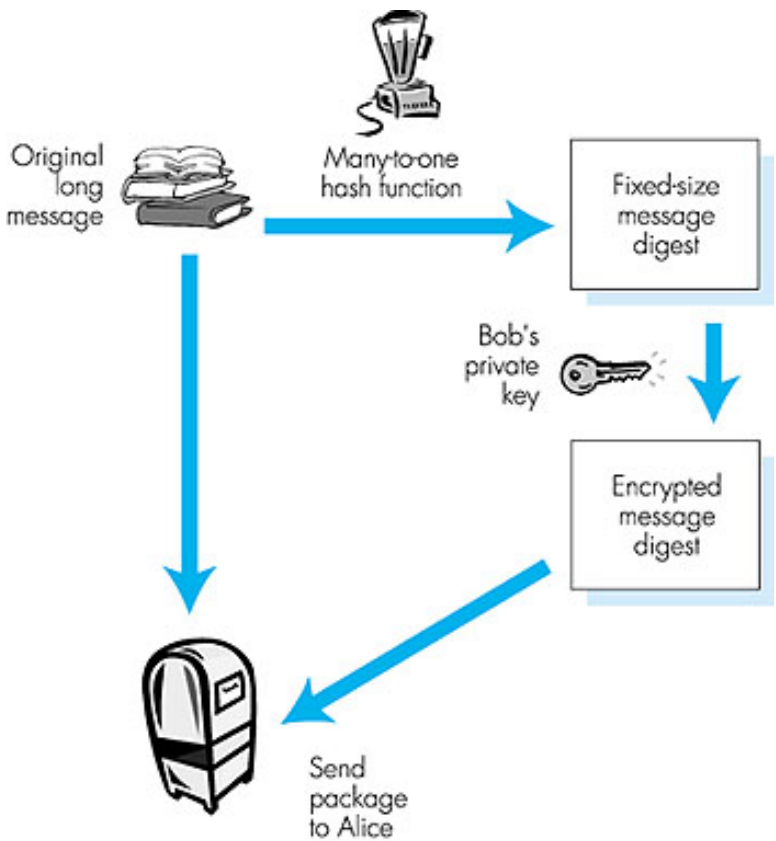
- applicare funzione hash H ad *m* per ottenere digest di msg a lungh. fissa, *H(m)*.

Proprietà delle funzioni Hash:

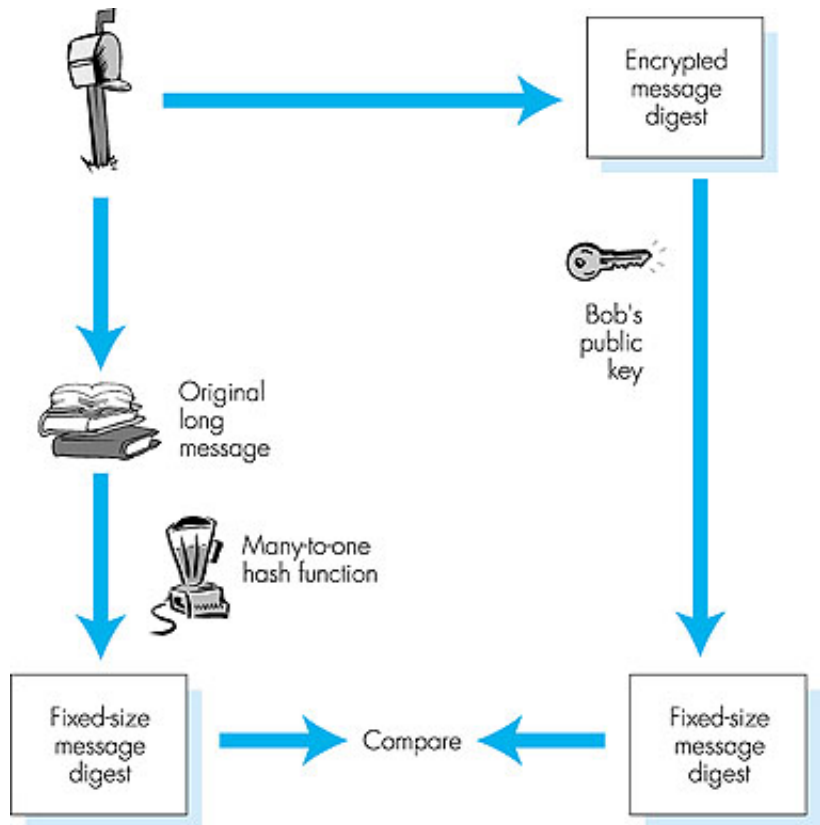
- Molti-ad-1
- Produce digest di msg a lunghezza fissa (fingerprint)
- Dato un digest di msg x, è computazionalmente inpossibile
  - trovare *m* t.c. *x = H(m)*
  - trovare due qualsiasi messaggi *m* ed *m'* t.c. *H(m) = H(m')*.

Integrità:
firma digitale = digest firmato del msg

Bob invia elettronicamente un messaggio firmato:



Alice verifica la firma e l'integrità del messaggio firmato elettronicamente:



Integrità: Algoritmi per la funzione hash

- Il checksum di Internet produrrebbe un digest di messaggio inaffidabile.
  - Troppo facile trovare due messaggi con stesso checksum!
- Funzione hash MD5 ampiamente utilizzata.
  - Calcola digest a 128-bit in 4 passi.
  - Presa una stringa arbitraria x a 128 bit, appare difficile costruire msg m il cui hash MD5 è uguale a x.
- SHA-1 è anche usata.
  - US standard
  - 160-bit message digest

Distribuzione delle chiavi: intermediari di fiducia

Problema 1):

- Come fanno due entità a stabilire una chiave segreta condivisa sulla rete?

Soluzione:

- Intermediario di fiducia: *Key Distribution Center* (KDC)

Problema 2):

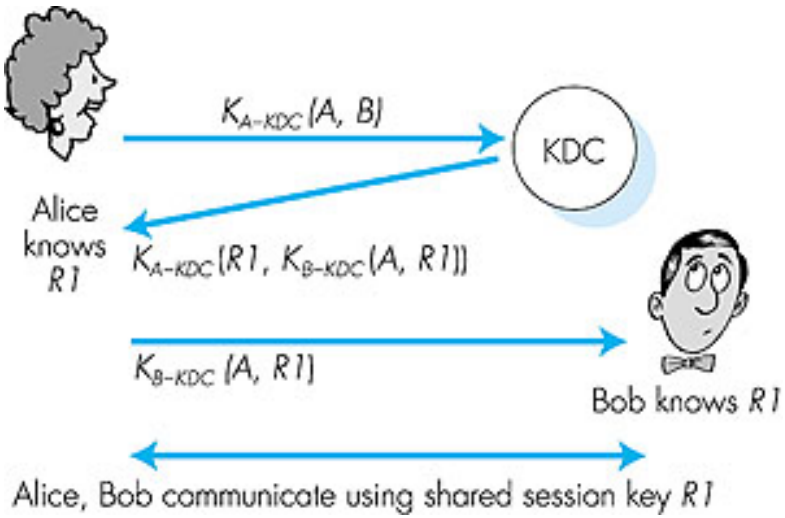
- Quando Alice ottiene la chiave pubblica di Bob (tramite un sito web, o e-mail o floppy), come fa a sapere che è proprio la chiave pubblica di Bob e non di Trudy?

Soluzione:

- Intermediario di fiducia: *Certification Authority* (CA)

Distribuzione delle chiavi: Key Distribution Center (KDC)

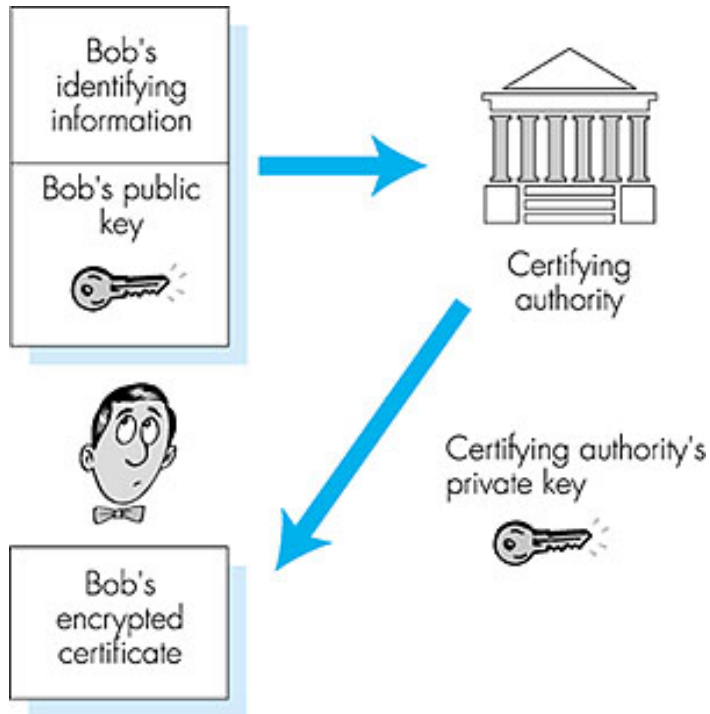
- Alice e Bob hanno bisogno di condividere una chiave simmetrica.
- il server KDC condivide una diversa chiave segreta per ogni utente registrato.
- Alice e Bob conoscono le proprie chiavi simmetriche, $K_{A\text{-}KDC}$ e $K_{B\text{-}KDC}$, per comunicare con il KDC.



- Alice comunica con il KDC, ottiene la chiave di sessione R1, e $K_{B\text{-}KDC}(A,R1)$
- Alice invia $K_{B\text{-}KDC}(A,R1)$ a Bob, Bob estrae R1
- Alice e Bob ora condividono la chiave simmetrica R1.

Distribuzione delle chiavi: Certification Authorities (CA)

- I CA legano le chiavi pubbliche ad una particulare entità.
- Un'entità (persona, router, etc.) può registrare la propria chiave pubblica con un CA.
  - Entità fornisce "prova di identità" al CA.
  - CA crea certificato che lega entità ad una chiave pubblica.
  - Certificato elettronicamente firmato dal CA.

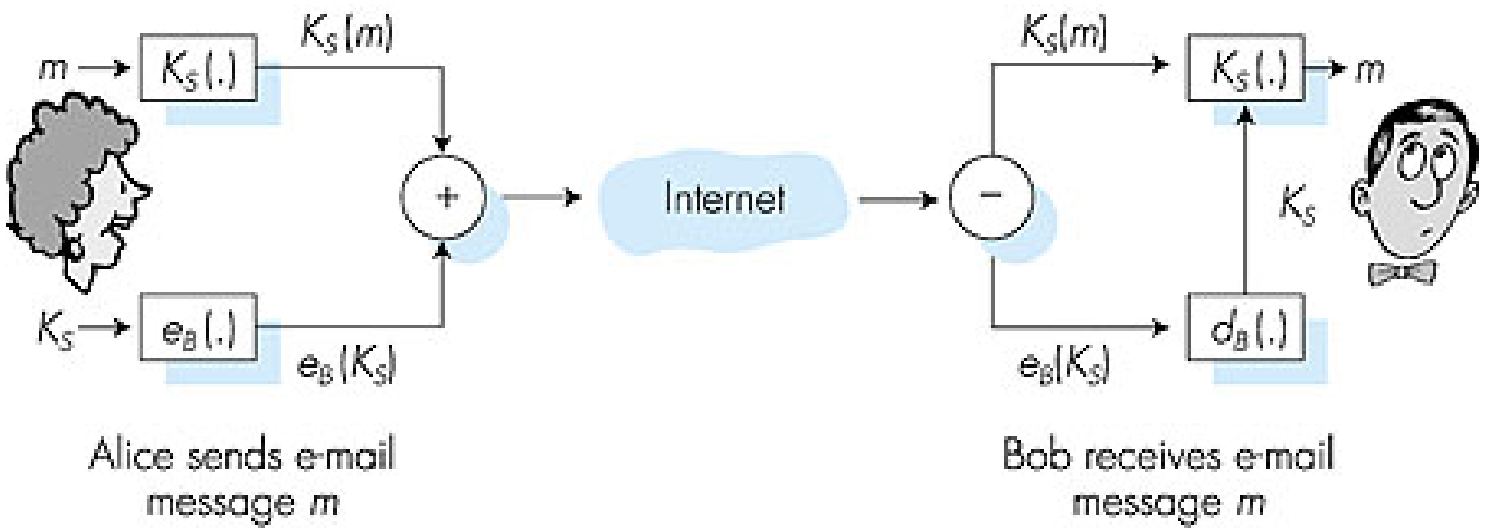


Qundo Alice vuole sapere la chiave pubblica di Bob:

- ottiene il certificato di Bob (da Bob o da qualche altra parte).
- Applica la chiave pubblica del CA al certificato di Bob ed ottiene la chiave pubblica di Bob

Sicurezza a livello di applicazione: posta elettronica sicura

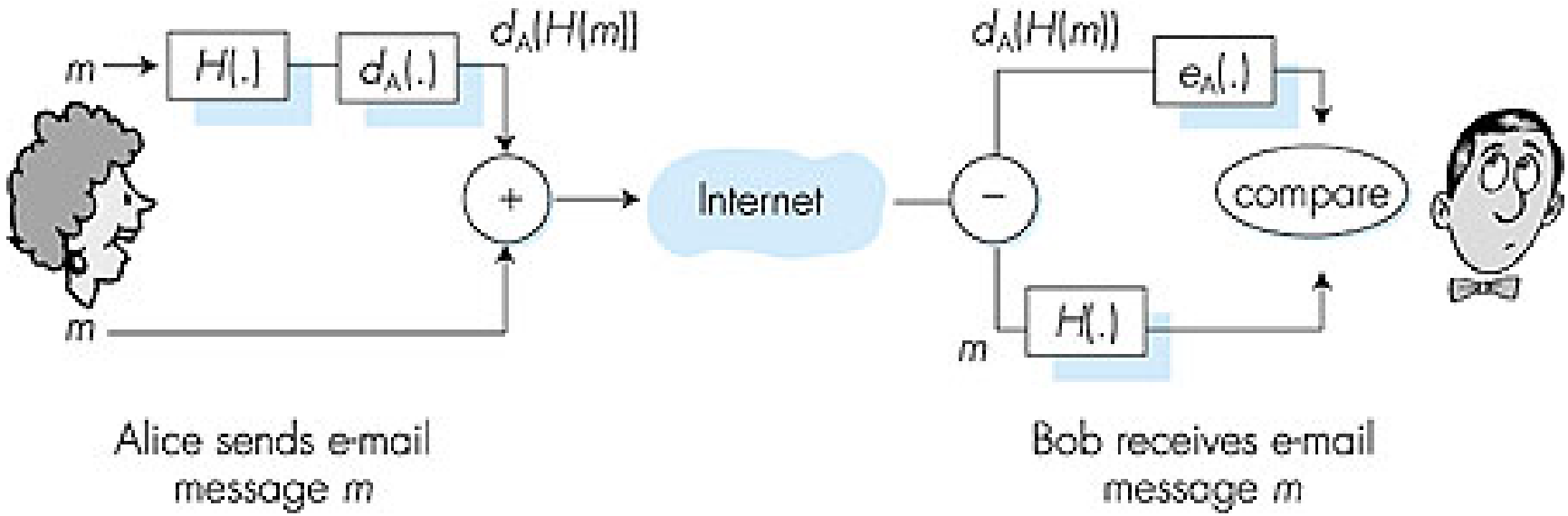
Alice vuole inviare un messaggio e-mail segreto, m, a Bob.



- genera una chiave simmetrica random, K_S.
- codifica il messaggio con K_S
- codifica anche K_S con la chiave pubblica di Bob
- invia sia $K_S(m)$ sia $e_B(K_S)$ a Bob.

Sicurezza a livello di applicazione: posta elettronica sicura (cont.)

Alice vuole fornire autenticazione del mittente ed integrità del messaggio.

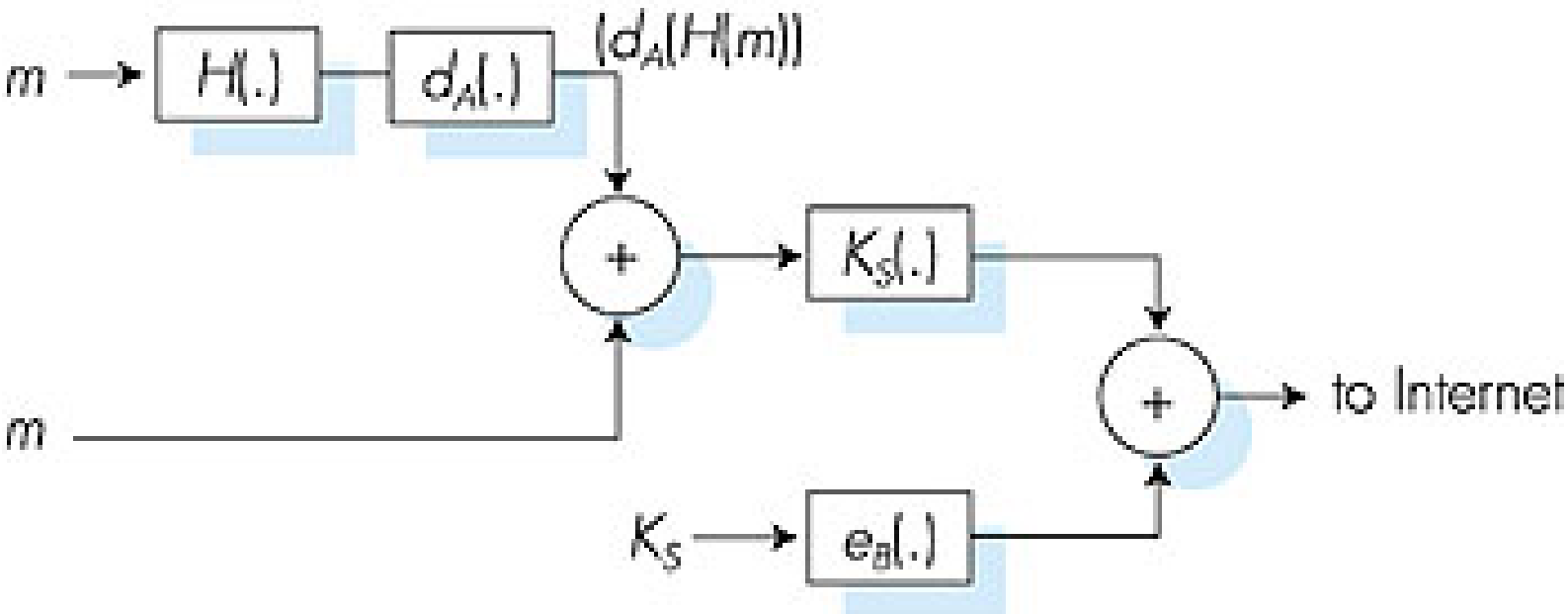


- Alice appone firma digitale al digest del messaggio

e invia sia il messaggio (in chiaro) sia la firma digitale

Sicurezza a livello di applicazione: posta elettronica sicura (cont.)

Alice vuole fornire segretezza, autenticazione del mittente, ed integrità del messaggio.



Nota: Alice usa sia la propria chiave privata sia la chiave pubblica di Bob.

Sicurezza a livello di applicazione: Pretty Good Privacy (PGP)

- Schema di cifratura della posta elettronica su Internet, uno standard de facto.
- Usa crittografia a chiave simmetrica, crittografia a chiave pubblica, funzione hash, e firma digitale come descritto.
- Fornisce segretezza, autenticazione del mittente, integrità.
- Inventore: Phil Zimmerman, soggetto ad investigazione federale per tre anni.

Un messaggio firmato PGP:

```
---BEGIN PGP SIGNED MESSAGE---
Hash: SHA1

Bob:My husband is out of town
   tonight.Passionately yours,
   Alice

---BEGIN PGP SIGNATURE---
Version: PGP 5.0
Charset: noconv
yhHJRHhGJGhgg/12EpJ+lo8gE4vB3mqJ
   hFEvZP9t6n7G6m5Gw2
---END PGP SIGNATURE---
```

Sicurezza a livello di trasporto: Secure Sockets Layer (SSL)

- SSL fornisce sicurezza a qualsiasi applicazione TCP-based che usa i servizi SSL
  - https viene usato fra browser WWW e server per E-commerce.
- Servizi SSL:
  - autenticazione del server
  - criptaggio dati
  - autenticazione del client (opzionale)

- Autenticazione del server:
  - browser SSL-enabled include chiavi pubbliche per CA di fiducia.
  - Browser richiede al server il certificato, rilasciato da CA di fiducia.
  - Browser usa chiave pubblica di CA per estrarre la chiave pubblica del server dal certificato.
- Visita il menu di sicurezza del tuo browser per vedere i suoi CA di fiducia.

Sicurezza a livello di trasporto: SSL (cont.)

Sessione SSL criptata:

- Browser genera una chiave simmetrica di sessione, la cripta con la chiave pubblica del server, invia la chiave criptata al server.
- Usando la sua chiave privata, il server decifra la chiave di sessione.
- Browser e server concordano che tutti i dati inviati nel socket TCP sono criptati con chiave di sessione.

- SSL: alla base di IETF Transport Layer Security (TLS).
- SSL può essere usato per applicazioni non-Web, p.e. IMAP.
- Autenticazione del client può essere fatta con i certificati del client.

Sicurezza a livello di trasporto: Secure Electronic Transactions (SET)

- Progettato per transazioni di carta di pagamento su Internet.
- Fornisce sicurezza fra 3 entità in gioco, tutti muniti di certificati:
  - cliente
  - venditore
  - banca del venditore
- SET specifica significati legali dei certificati.
  - Distribuzione di responsabilità per le transazioni
- Numero di carta del cliente viene passato alla banca del venditore senza che quest'ultimo veda mai il numero in chiaro.
  - Previene i venditori dal rubare o far trapelare i numeri di carta di pagamento.
- Tre componenti sw:
  - il portafogli del browser
  - il server del venditore
  - il gateway dell'acquirente

Sicurezza a livello di rete: IPsec

- Segretezza:
  - host mittente cripta i dati in IP datagram
  - segmenti TCP e UDP; messaggi ICMP e SNMP.
- Autenticazione
  - host destinatario può autenticare indirizzo IP del sorgente
- Due protocolli principali:
  - Authentication Header (AH) protocol
  - Encapsulation Security Payload (ESP) protocol

- Per entrambi AH e ESP, handshake fra sorgente e destinazione:
  - crea canale logico a livello di rete detto *Security Association* (SA)
- Ogni SA è unidirezionale.
- Unicamente determinato da:
  - protocollo di sicurezza (AH o ESP)
  - indirizzo IP del sorgente
  - ID di connessione a 32 bit