# Sicurezza nelle reti: protezione della comunicazione

Gaia Maselli

maselli@di.uniroma1.it

# La sicurezza nelle reti

- Principi di crittografia
- Integrità dei messaggi
- Autenticazione end-to-end
- HTTPS

# Sicurezza nella comunicazione

Riservatezza o confidenzialità: solo mittente e destinatario devono comprendere il contenuto del messaggio

- Inviare messaggi cifrati
- Ricevere il codice di decifratura

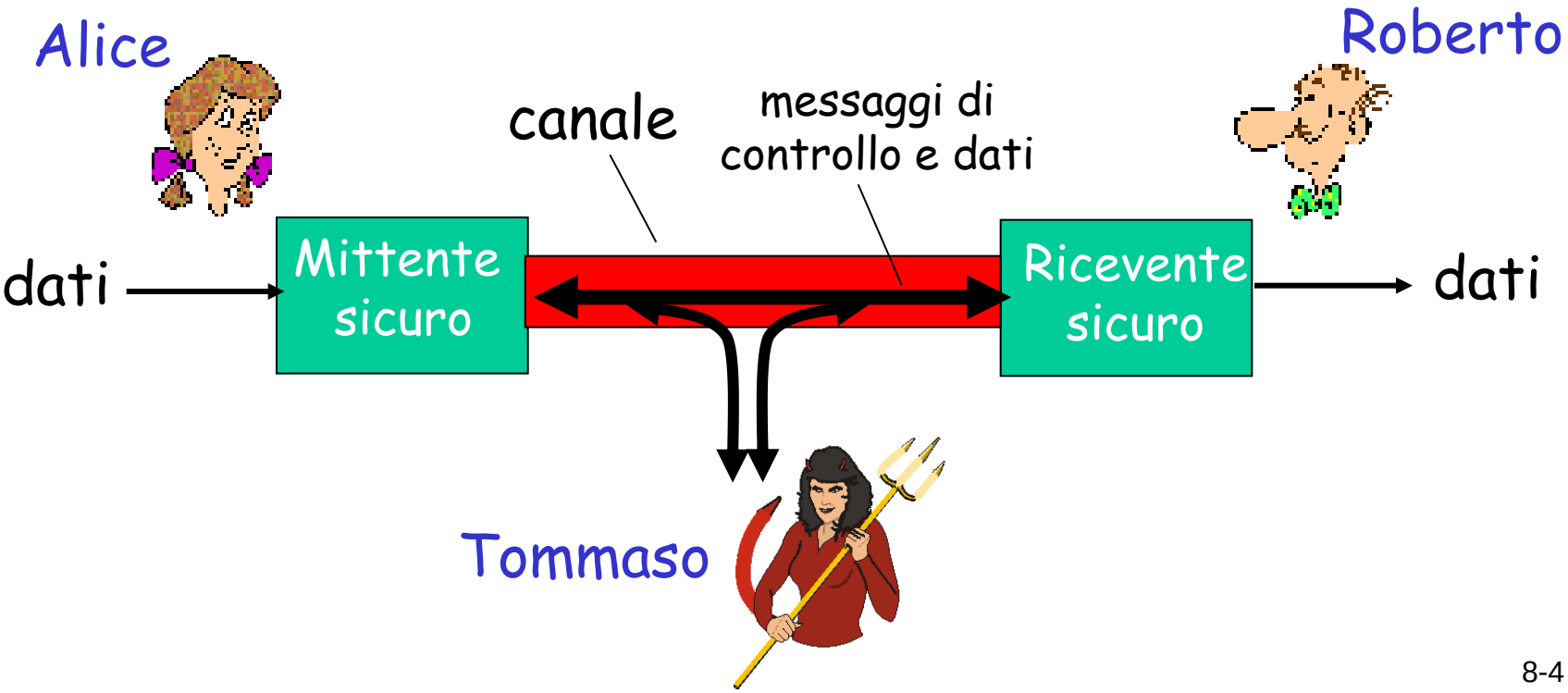
Autenticazione: mittente e destinatario devono essere sicuri della loro identità

Integrità del messaggio: mittente e destinatario devono essere sicuri che il contenuto non subisca alterazioni durante la trasmissione (per cause fortuite o per manipolazioni)

Disponibilità e controllo dell'accesso: un servizio deve essere accessibile a chi è legittimamente autorizzato.

# Mittente, ricevente e intruso: Alice, Roberto e Tommaso

- Scenario ben noto nel mondo della sicurezza di rete
- Roberto e Alice vogliono comunicare in modo sicuro
- Tommaso (l'intruso) può intercettare, rimuovere, aggiungere messaggi o modificare il loro contenuto



# Chi sono Alice e Roberto?

Nella vita reale Alice e Roberto possono essere:

- browser/server Web durante una transazione elettronica (es. un acquisto on-line)
- client/server di banche on-line
- server DNS
- sistemi che si scambiano tabelle d'instradamento
- altro

# Là fuori ci sono "cattivi" ragazzi (e ragazze)

D: Cosa può fare un nemico?

R: Molto!

- *spiare:* intercettare i messaggi
- *aggiungere* messaggi e sovraccaricare il sistema
- *impersonare* un altro soggetto
- *dirottare* una sessione in corso e sostituirsi al mittente o al destinatario
- *negare il servizio*

*E molto altro ancora! ……*

# La sicurezza nelle reti

Sicurezza di rete

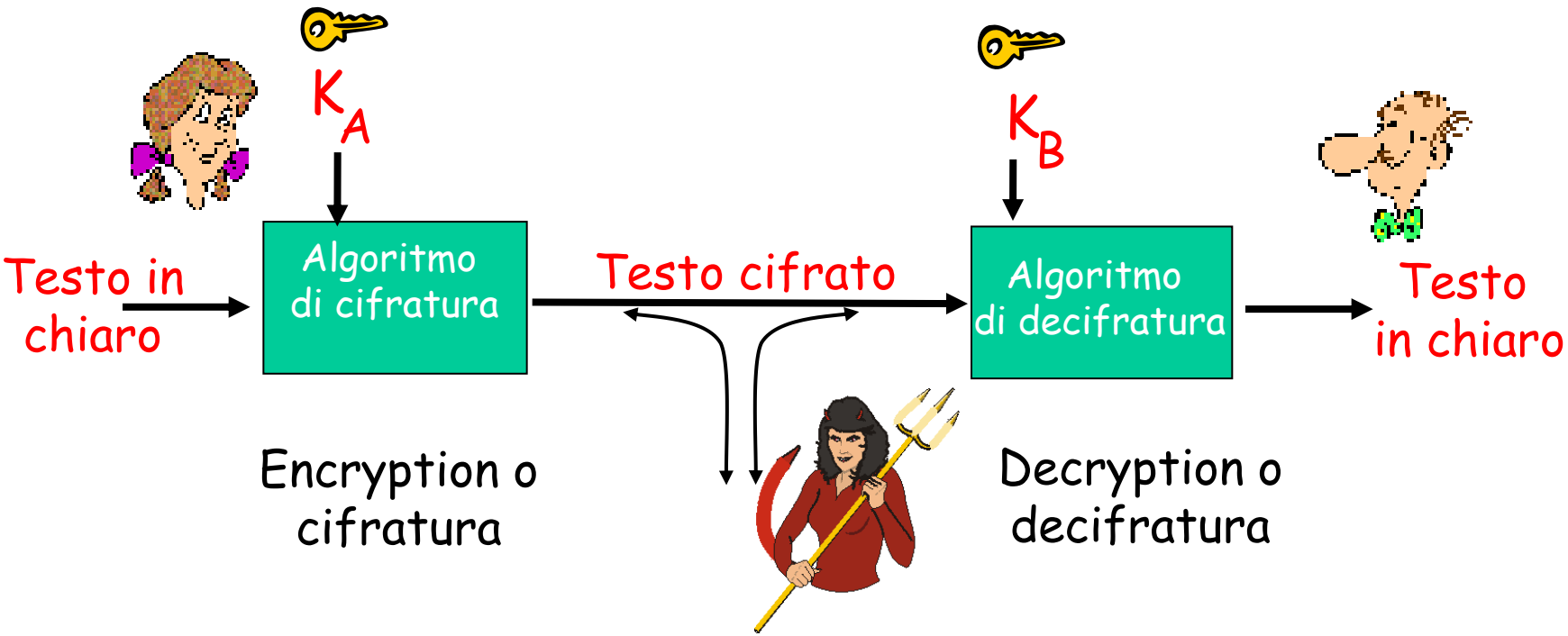
Principi di crittografia

Integrità dei messaggi

Autenticazione end-to-end

Rendere sicure le connessioni TCP: SSL

# Principi di crittografia

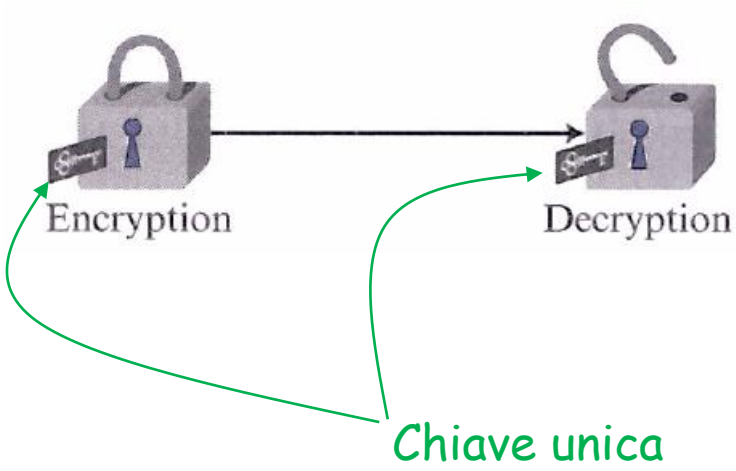



Crittografia: trasformazione dei messaggi per renderli sicuri e immuni agli attacchi

# Metodi

Sistemi a chiave simmetrica:

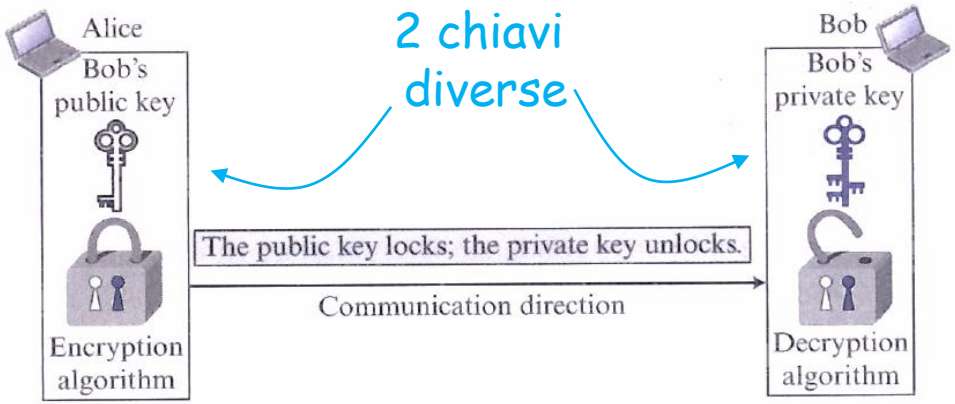
la chiave per la cifratura e decifratura è la stessa e può essere usata per comunicazione bidirezionale (da cui il termine simmetrica)



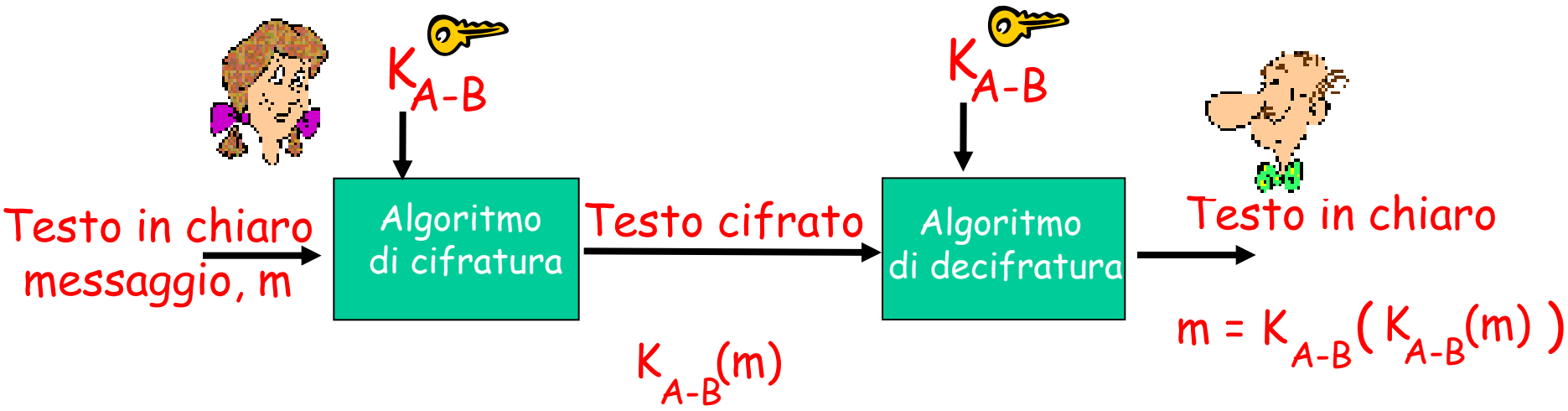

Sistemi a chiave asimmetrica:

si usa una coppia di chiavi associate al destinatario:

- la chiave di cifratura è **pubblica**
- la chiave di decifratura è **privata**
- Comunicazione unidirezionale con una coppia di chiavi



# Crittografia a chiave simmetrica




Per **cifrare** il messaggio è necessario:

- Algoritmo di cifratura
- Chiave simmetrica condivisa

Per **decifrare** il messaggio è necessario:

- Algoritmo di decifratura
- Chiave simmetrica condivisa

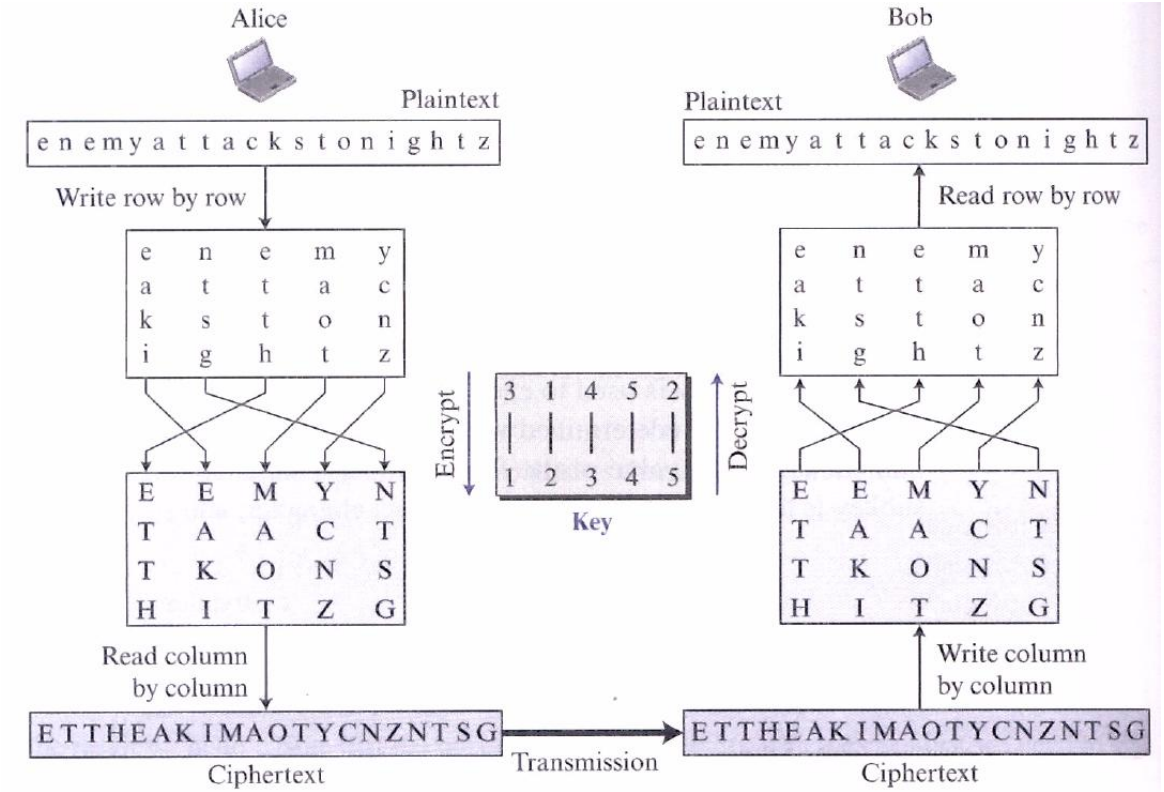
N.B. La chiave simmetrica **condivisa** è **segreta** e deve essere cambiata a priori (come viene stabilita???)

# Metodi di cifratura simmetrica

- Per sostituzione
  - Cifrario monoalfabetico: sostituzione di una lettera con un'altra

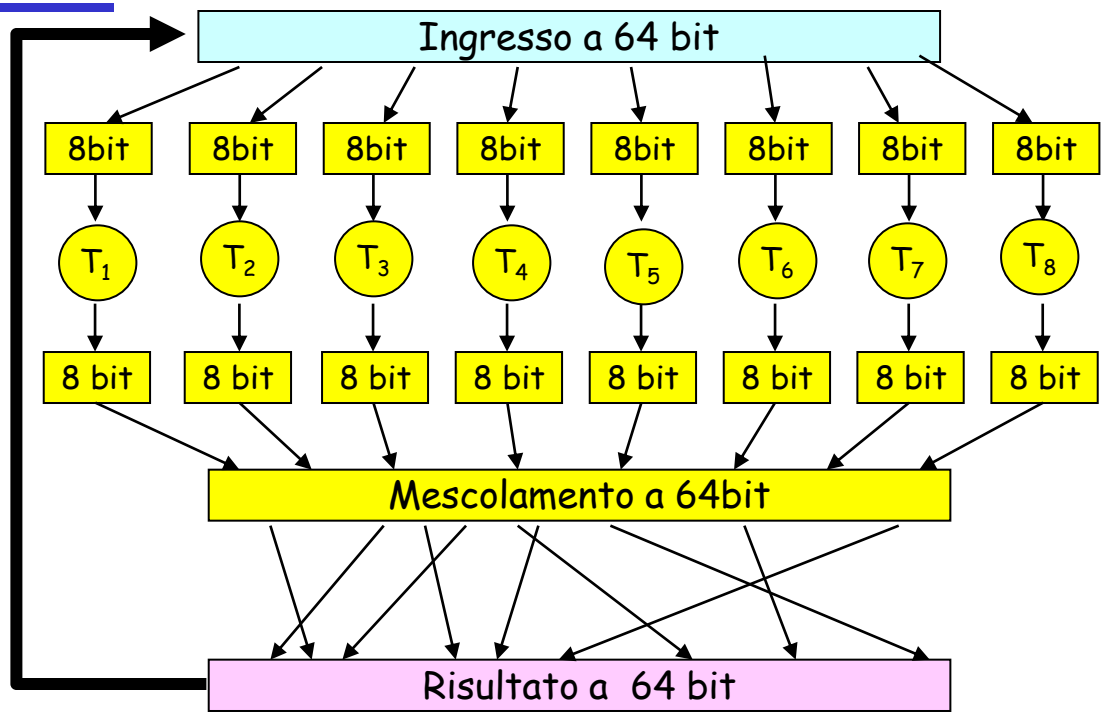
| Plaintext → | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Ciphertext → | N | O | A | T | R | B | E | C | F | U | X | D | Q | G | Y | L | K | H | V | I | J | M | P | Z | S | W |

- Per trasposizione



# Cifrario a blocchi

ciclo per
*n* iterazioni




- $T_i$: tabelle di corrispondenza tra blocco in chiaro e blocco cifrato

- Comuni cifrari a blocchi: DES, 3DES, AES

# Crittografia a chiave asimmetrica (o pubblica)

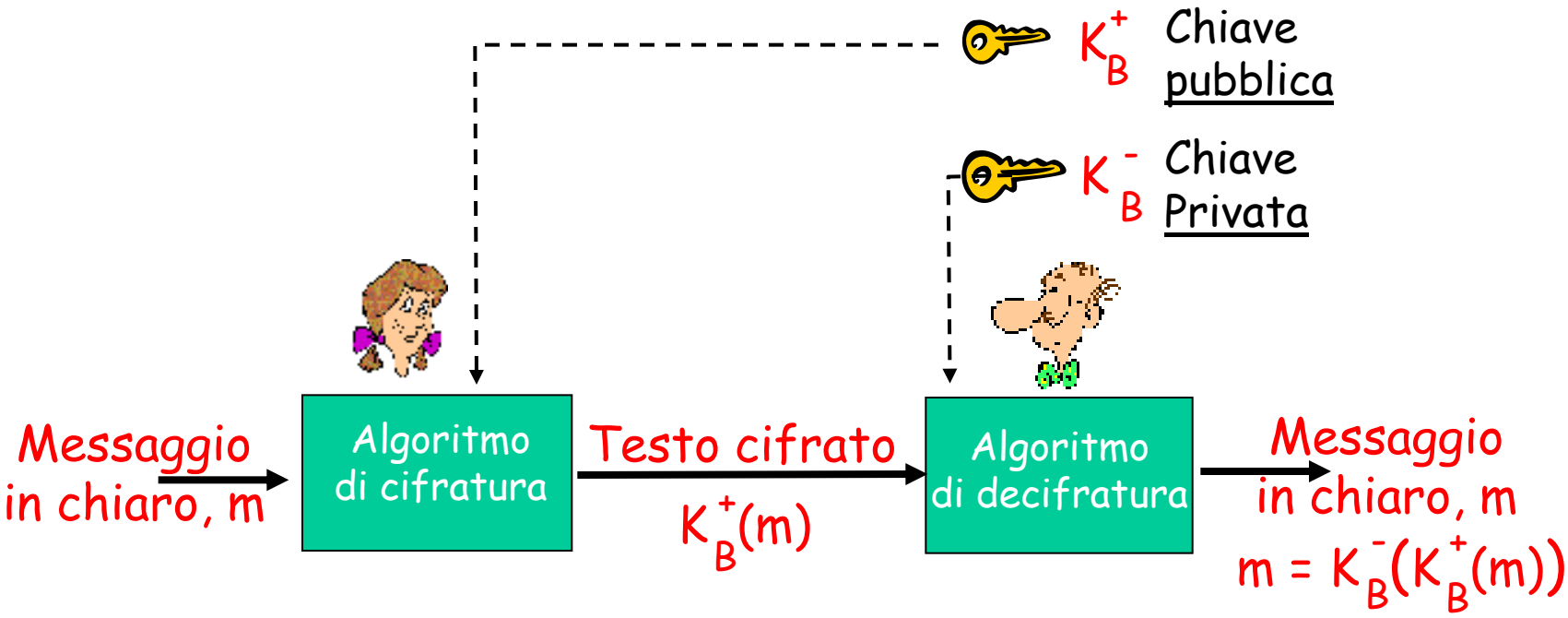
*Crittografia a chiave simmetrica*

- Richiede che mittente e destinatario condividano una chiave segreta
- D: come si concorda la chiave (specialmente se i due interlocutori non si sono mai "incontrati")?

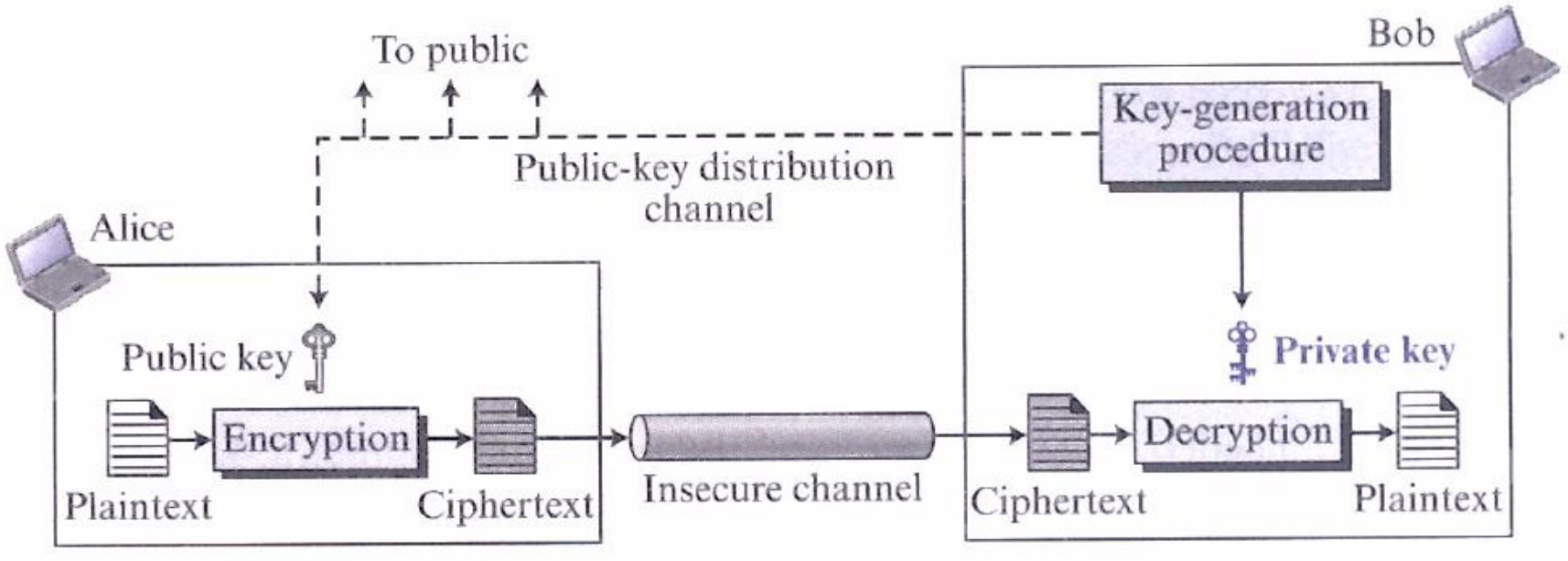
*Crittografia a chiave pubblica*

- approccio radicalmente diverso [Diffie-Hellman, RSA]
- mittente e destinatario *non* condividono una chiave segreta
- la chiave di cifratura *pubblica* è nota *a tutti*
- la chiave di cifratura *privata* è nota solo al destinatario

# Crittografia a chiave pubblica



# Crittografia a chiave asimmetrica




- Una coppia di chiavi può essere usata solo per comunicare in una direzione (il destinatario è il possessore delle chiavi)
- La coppia di chiavi è unica per tutti i mittenti

# Algoritmi di cifratura a chiave pubblica

Requisiti:

1. $K_B^+(\cdot)$ e $K_B^-(\cdot)$ tale che

$$K_B^-(K_B^+(m)) = m$$

2. data la chiave pubblica $K_B^+$, deve essere impossibile calcolare la chiave privata $K_B^-$

Algoritmo RSA: acronimo derivato dal nome dei suoi autori: Rivest, Shamir e Adelson

# RSA: scelta delle chiavi

1. Scegliere due numeri primi di valore elevato: *p, q.* (es.: 1024 bit ciascuno)

2. Calcolare *n = pq, z = (p-1)(q-1)*

3. Scegliere *e* (con *e < n*) tale che non abbia fattori in comune con z. (*e, z* sono "relativamente primi").

4. Scegliere *d* tale che *ed-1* sia esattamente divisibile per *z*. (in altre parole: *ed* mod *z = 1* ).

5. La chiave *pubblica* è $\underbrace{(n,e)}_{K_B^+}$, quella *privata* è $\underbrace{(n,d)}_{K_B^-}$.

# RSA: cifratura, decifratura

0. Dati $(n,e)$ e $(n,d)$ calcolati come abbiamo appena visto,

1. Per la codifica di $m$ si calcola

$$c = m^e \bmod n$$

2. Per decifrare il messaggio ricevuto, $c$, si calcola

$$m = c^d \bmod n$$

Incredibile! $m = (\underbrace{m^e \bmod n}_{c})^d \bmod n$

# Un esempio di RSA:

Roberto sceglie *p=5, q=7*. Poi *n=35, z=24*.

*e=5* (così *e, z* sono relativamente primi).

*d=29* (così *ed-1* è esattam. divisibile per z).

cifratura:

| lettera | m | $m^e$ | $c = m^e \bmod n$ |
|---|---|---|---|
| l | 12 | 1524832 | 17 |

decifratura:

| c | $c^d$ | $m = c^d \bmod n$ | lettera |
|---|---|---|---|
| 17 | 481968572106750915091411825223071697 | 12 | l |

# RSA: un'importante proprietà

La seguente proprietà sarà *molto* utile più avanti:

$$K_B^-(K_B^+(m)) = m = K_B^+(K_B^-(m))$$

Si usa prima la chiave pubblica, e poi quella privata

Si usa prima la chiave privata, e poi quella pubblica

*Il risultato non cambia!*

# La sicurezza nelle reti

- Sicurezza di rete
- Principi di crittografia → confidenzialità
- Integrità dei messaggi
- Autenticazione end-to-end
- HTTPS

# Integrità del messaggio

Roberto riceve un messaggio da Alice, e vuole essere sicuro che:

- il messaggio provenga effettivamente da Alice
- il messaggio non sia stato alterato lungo il cammino

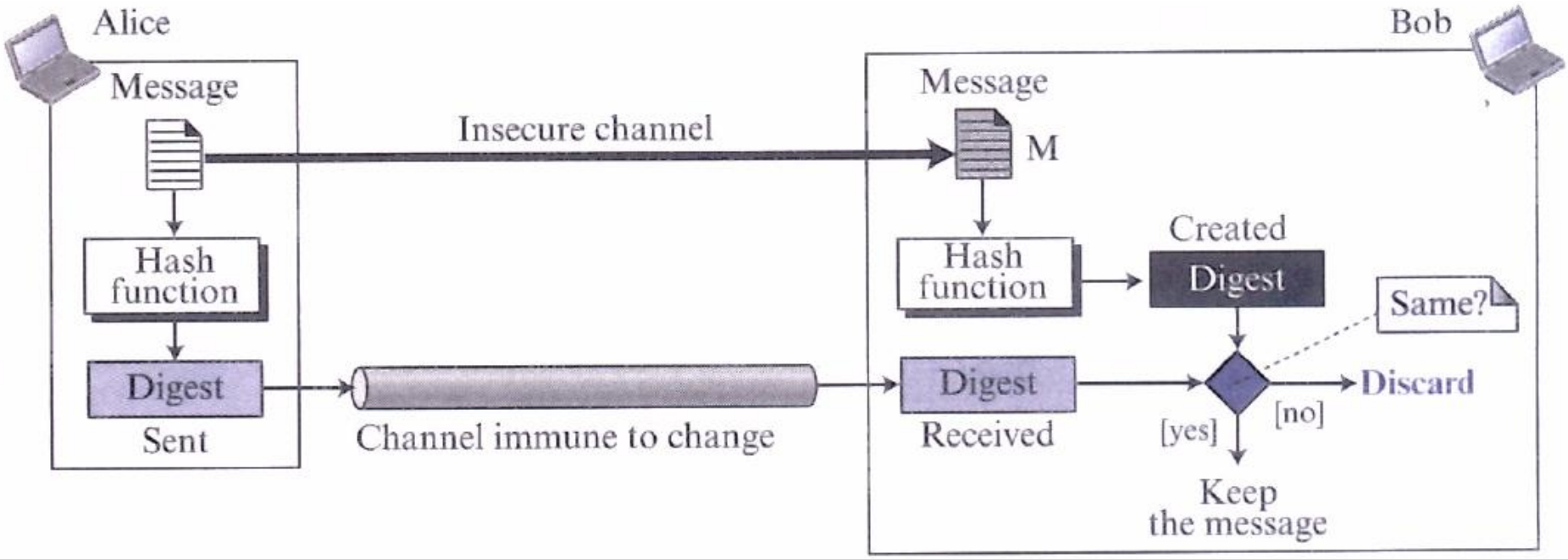
Funzioni hash crittografiche

- prende in input m, produce un valore a lunghezza fissa, H(m)
- Deve essere computazionalmente impossibile trovare due messaggi x e y tali che H(x) = H(y)
  - o anche: dato m = H(x), (con x sconosciuta), è impossibile determinare x.

# Funzione hash crittografica

- Garantisce integrità (il messaggio non è stato cambiato)
- Funzione hash che produce *digest* (immagine compressa del messaggio)
- Viene inviato sia il messaggio, sia il digest

N.B. il messaggio viaggia in chiaro!!!

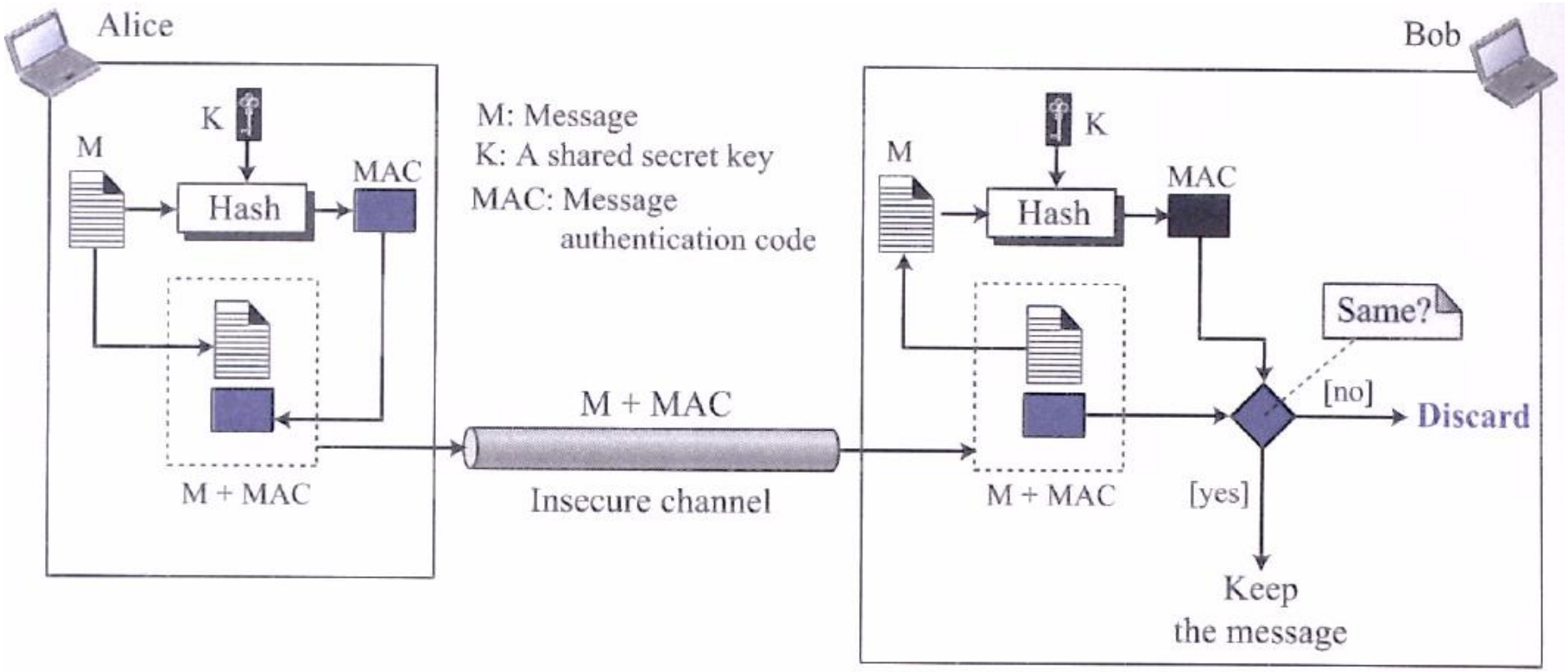


# Funzioni Hash

- MD5 è molto usato per l'hash dei messaggi (RFC 1321)
  - MD sta per Message Digest
  - Calcola una hash di 128 bit con un processo a 4 fasi
  - Con una stringa x di 128 bit arbitrari, appare difficile costruire un messaggio m il cui hash MD5 sia uguale a x
- È molto usato anche Secure Hash Algorithm (SHA)
  - Standard statunitense [NIST, FIPS PUB 180-1]
  - Produce una sintesi del messaggio di 160 bit

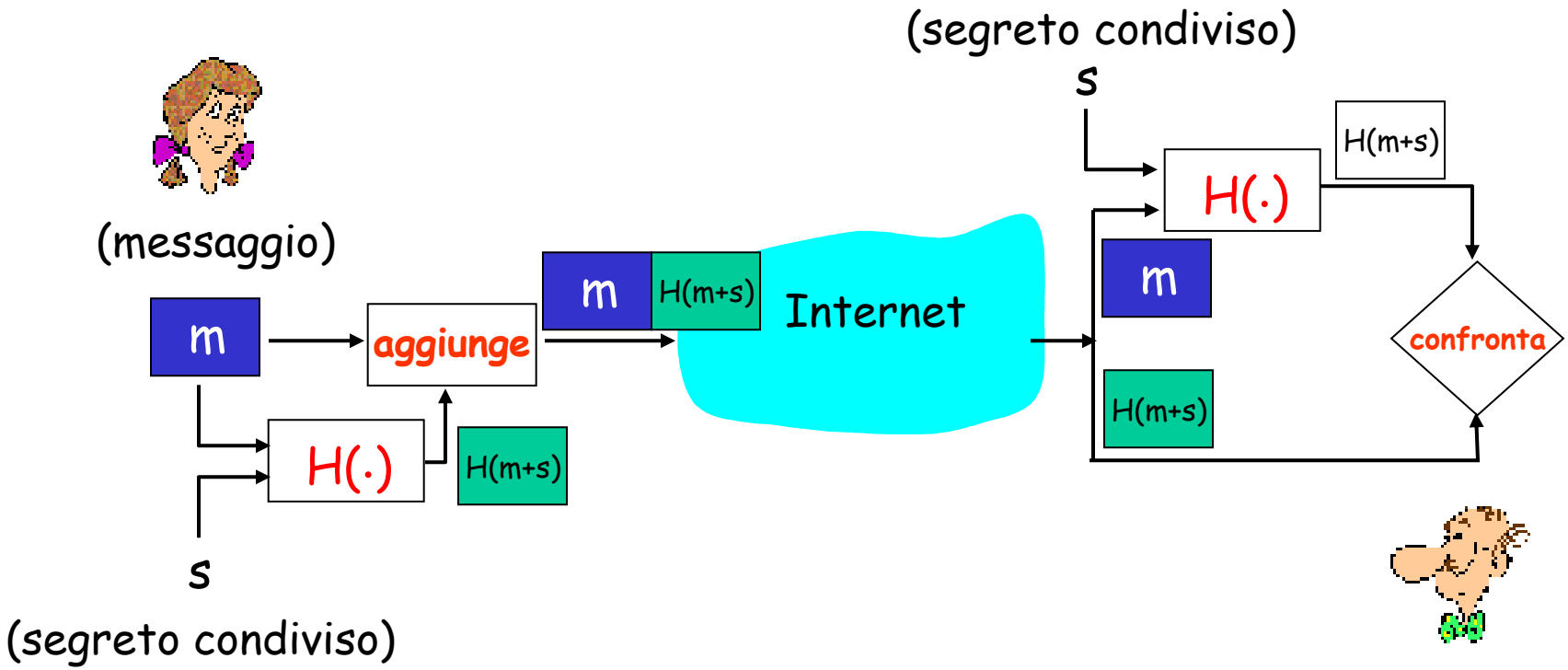
# Autenticazione dei messaggi

- Per autenticare l'origine del messaggio (Il messaggio è stato scritto da Alice) viene inserito un codice segreto condiviso tra Alice e Bob
- Si usa insieme alla hash crittografica per creare un Message Authentication code (MAC)



# MAC: riassunto

- Integrità + autenticazione
- Funzione hash + codice segreto condiviso



# Firma digitale

Tecnica crittografica analoga all'invio di una tradizionale "firma scritta"

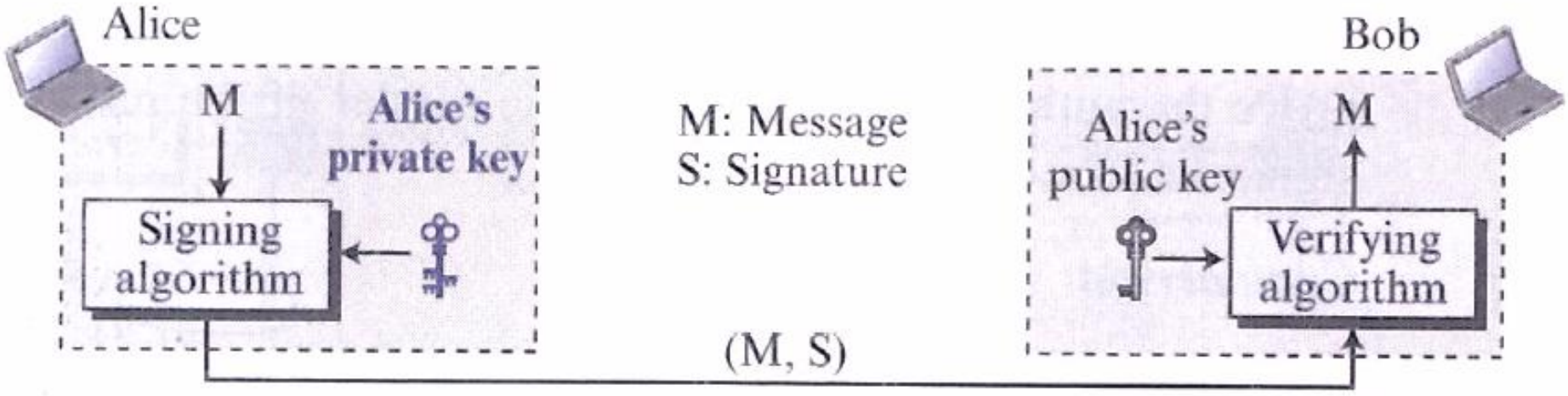
- Il mittente (Alice) firma digitalmente un documento, stabilendo che lui è l'unico proprietario/creatore del messaggio.
- Verificabile e non falsificabile: il destinatario (Roberto) può dimostrare che Alice e nessun altro (Roberto incluso) può aver firmato il documento.
- Avviene mediante l'uso di una coppia di chiave pubblica e privata di chi firma

# Firma digitale

Creazione della firma digitale di un messaggio M:

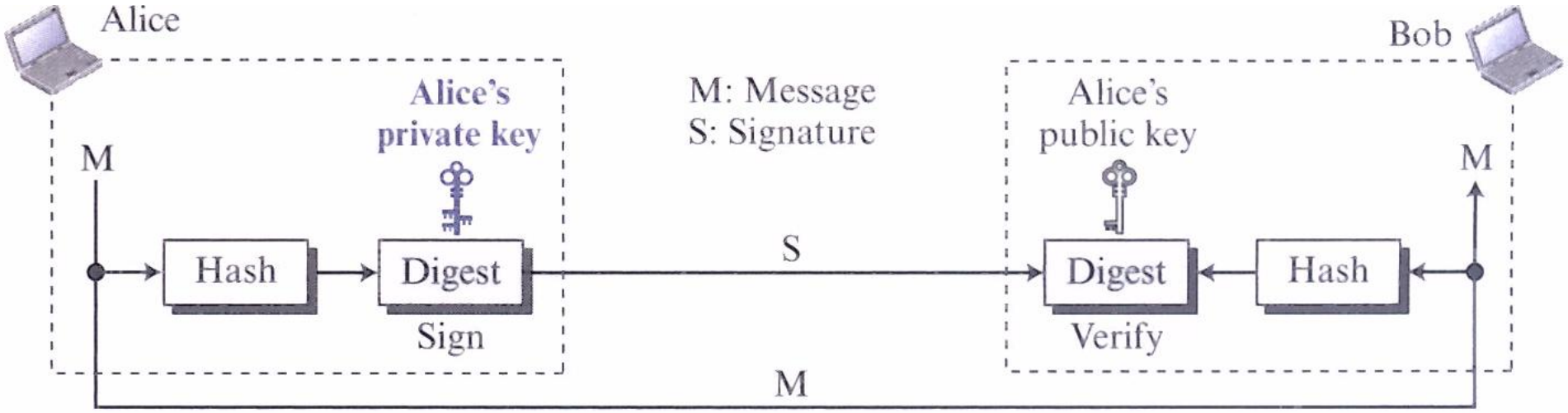
- Alice firma un messaggio M, cifrandolo con la sua chiave privata, creando così un messaggio "firmato"
- Bob decifra il messaggio applicando la chiave pubblica di Alice

N.B. è il meccanismo di crittografia asimmetrica applicata in modo inverso



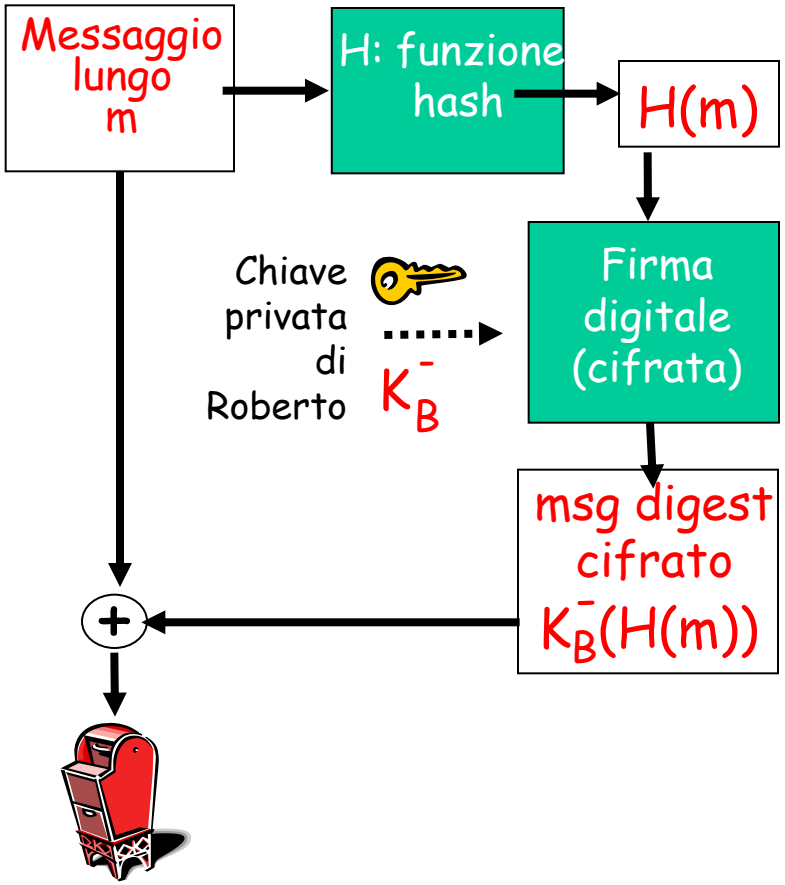
# Firma digitale (messaggi digest firmati)

- Impossibile firmare un messaggio di un'altra persona a meno che non si conosca la sua chiave segreta
- La firma digitale garantisce la proprietà di non ripudio: Alice non può negare di aver firmato il documento e Roberto può verificare che
  - ✓ Alice ha firmato M
  - ✓ Nessun altro ha firmato M
  - ✓ Alice ha firmato M e non M'.
- Poiché la crittografia asimmetrica è inefficiente su messaggi lunghi, in genere si applica la firma digitale a un MESSAGE DIGEST

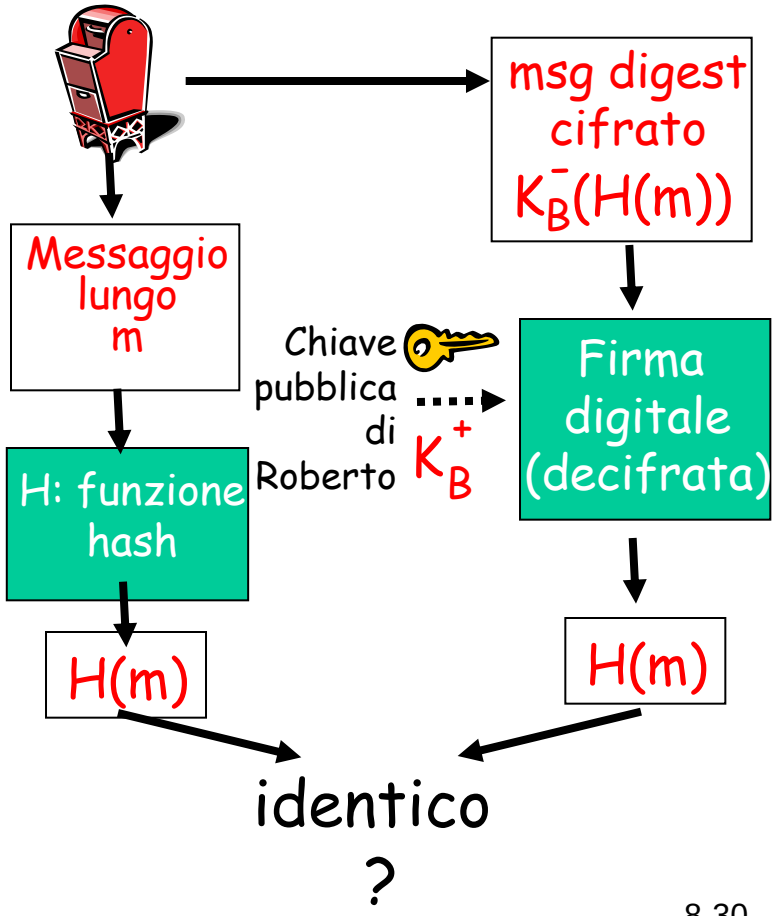


# Firma digitale = messaggi digest firmati

Roberto invia un messaggio con la firma digitale:




Alice verifica la firma e l'integrità del messaggio con la firma digitale:



# Certificazione della chiave pubblica

Problema per la crittografia a chiave pubblica:

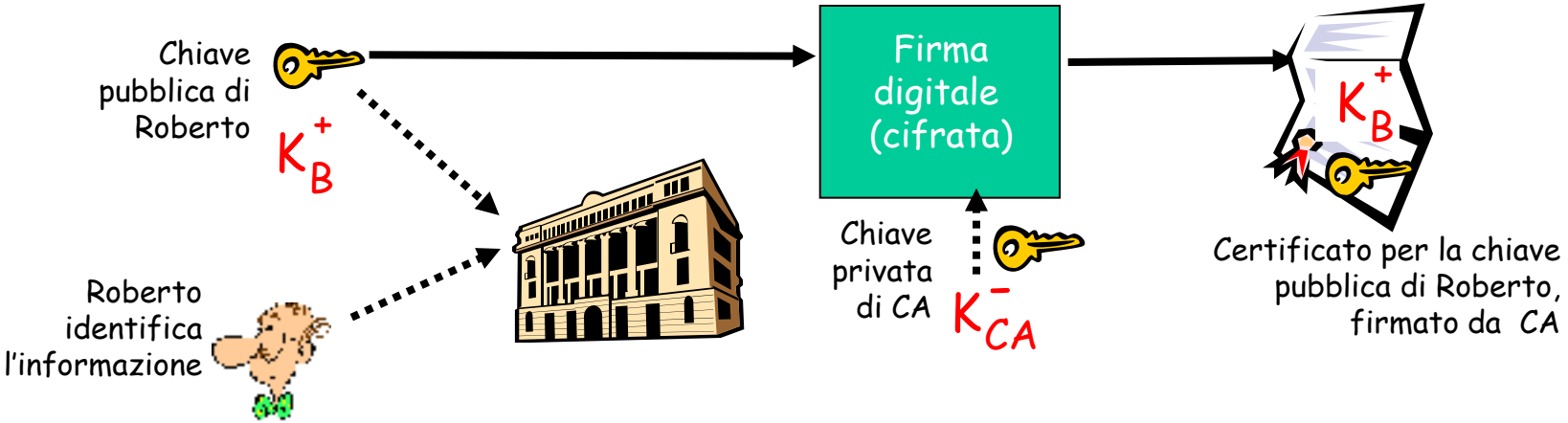
- Quando Alice riceve la chiave pubblica di Roberto (attraverso un dischetto, il sito web o via e-mail), come fa a sapere che è veramente la chiave pubblica di Roberto e non, magari, quella di Tommaso?

Soluzione:

- Autorità di certificazione (CA, *certification authority*)

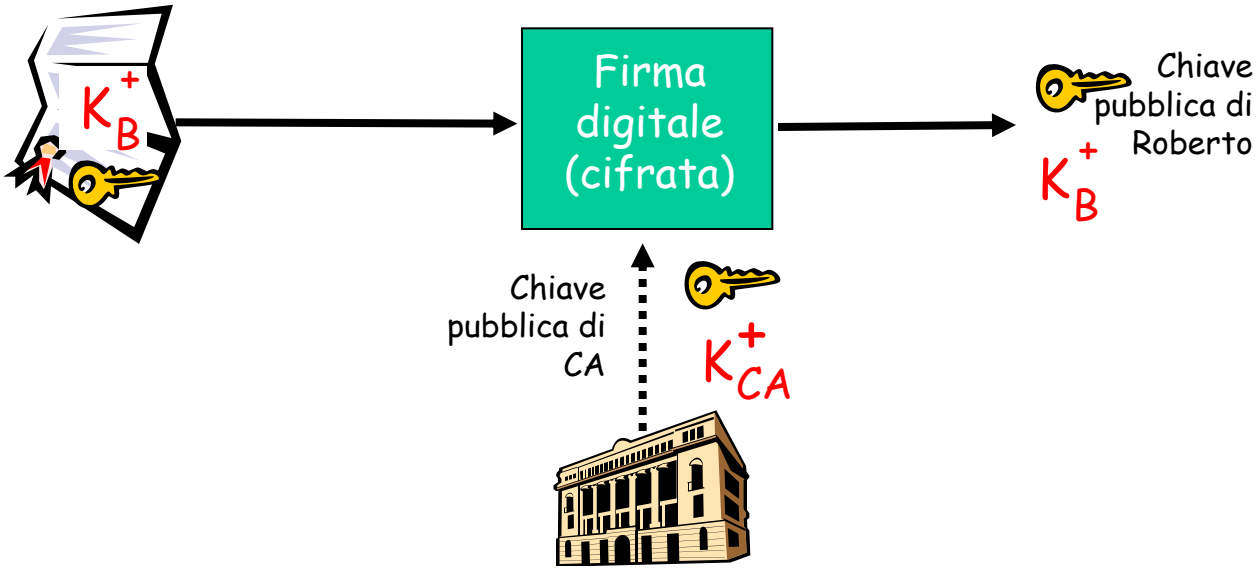
# Autorità di certificazione

- Autorità di certificazione (CA): collega una chiave pubblica a una particolare entità, E.
- E (persona fisica, router) registra la sua chiave pubblica con CA.
  - E fornisce una "prova d'identità" a CA.
  - CA crea un certificato che collega E alla sua chiave pubblica.
  - Il certificato contiene la chiave pubblica di E con firma digitale di CA (CA dice "questa è la chiave pubblica di E")



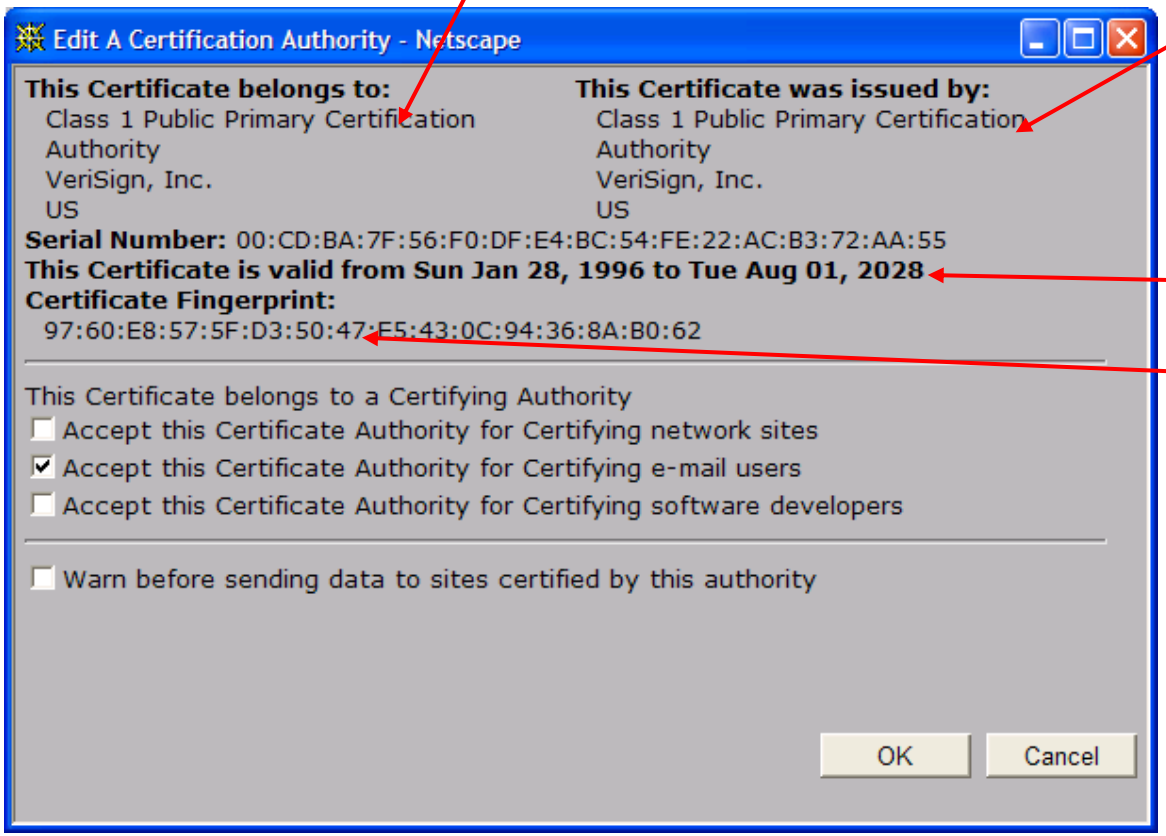
# Autorità di certificazione

- Quando Alice vuole la chiave pubblica di Roberto:
  - prende il certificato di Roberto
  - applica la chiave pubblica di *CA* al certificato pubblico di Roberto e ottiene la chiave pubblica di Roberto



# Un certificato contiene:

- Numero di serie
- Informazioni sul titolare, compreso l'algoritmo e il valore della chiave (non illustrato)




- Informazioni su chi ha emesso il certificato
- Date valide
- Firma digitale di chi lo ha emesso

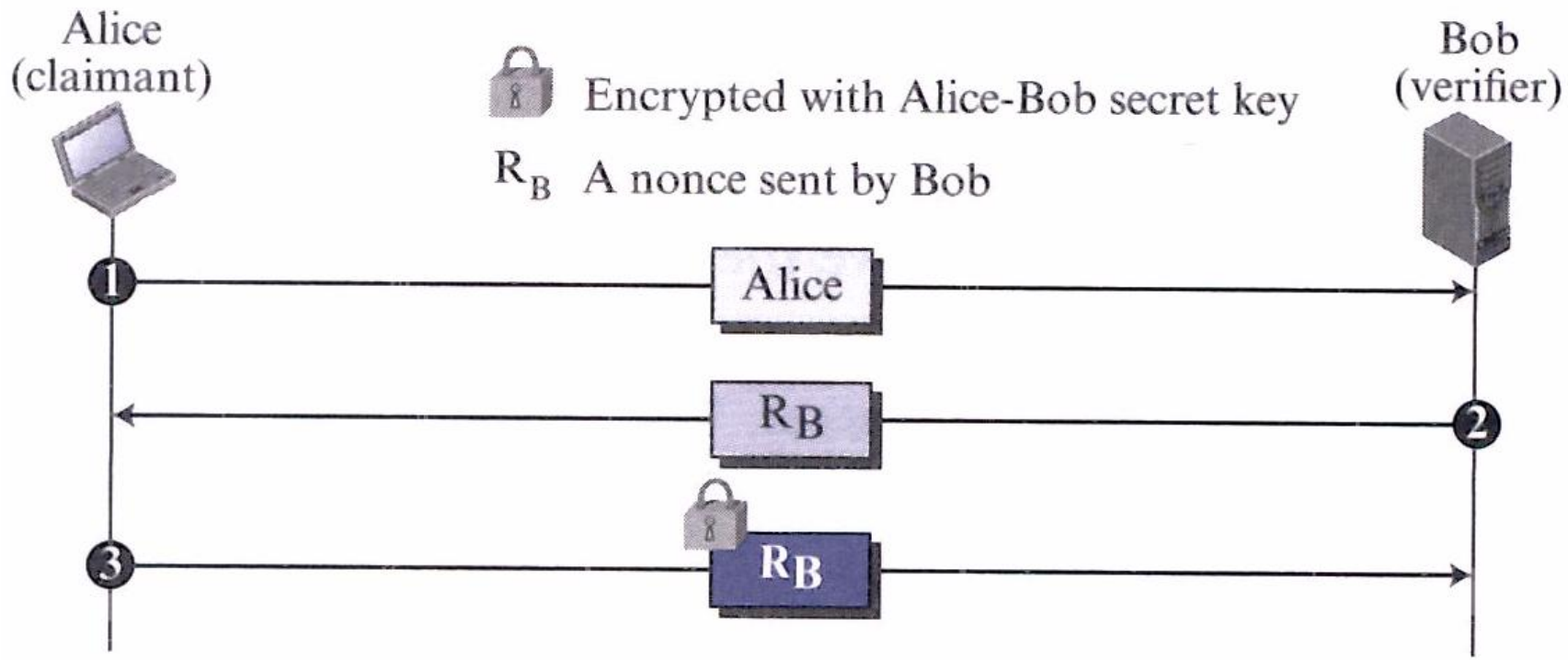
# La sicurezza nelle reti

Sicurezza di rete
Principi di crittografia
Integrità dei messaggi
Autenticazione end-to-end
HTTPS

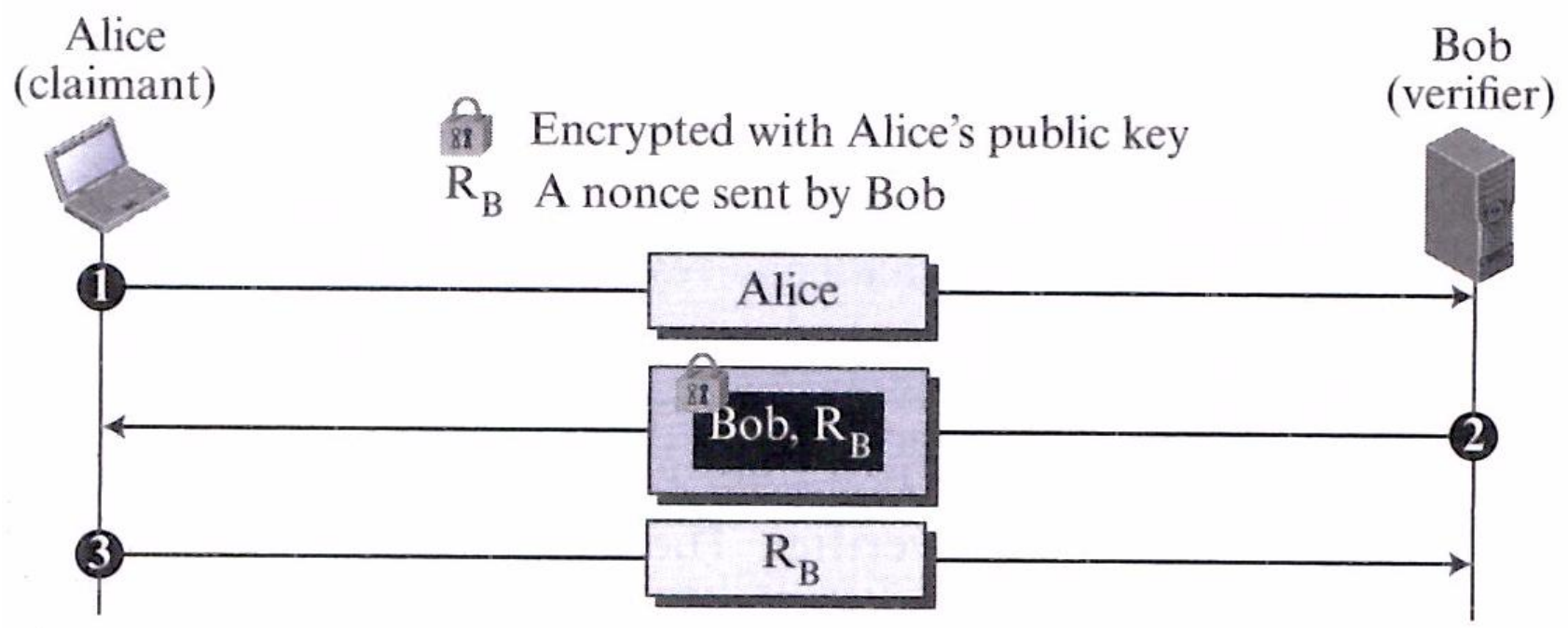
# Autenticazione end-to-end

- Tecnica che permette a un'entità (verifier) di autenticare l'entità (claimant) con cui si vuole interagire
- L'entità può essere una persona, un processo, un client o un server
- In genere avviene mediante un meccanismo che permette di verificare che si conosce un segreto senza scambiare (trasmettere) il segreto

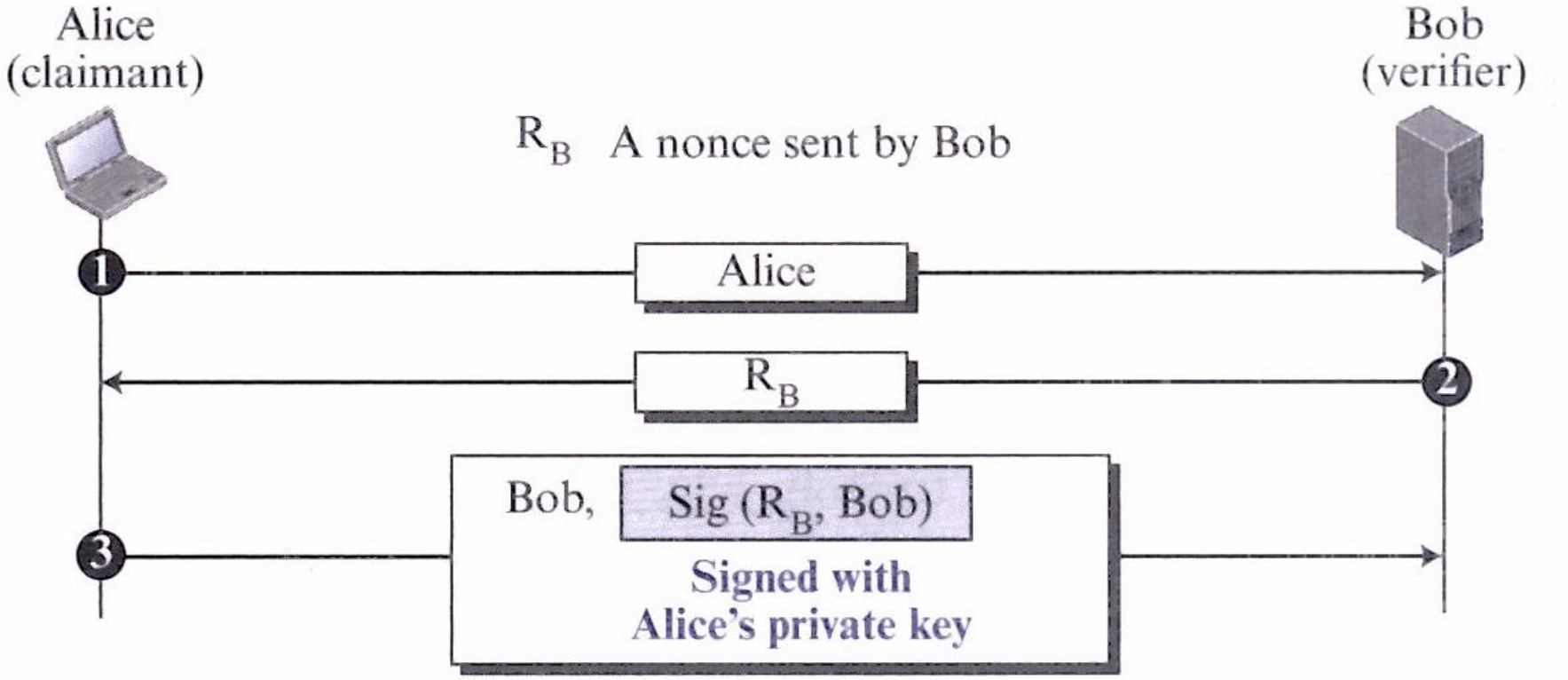
# Autenticazione mediante chiave simmetrica



# Autenticazione mediante chiave asimmetrica



# Autenticazione mediante firma digitale



# La sicurezza nelle reti

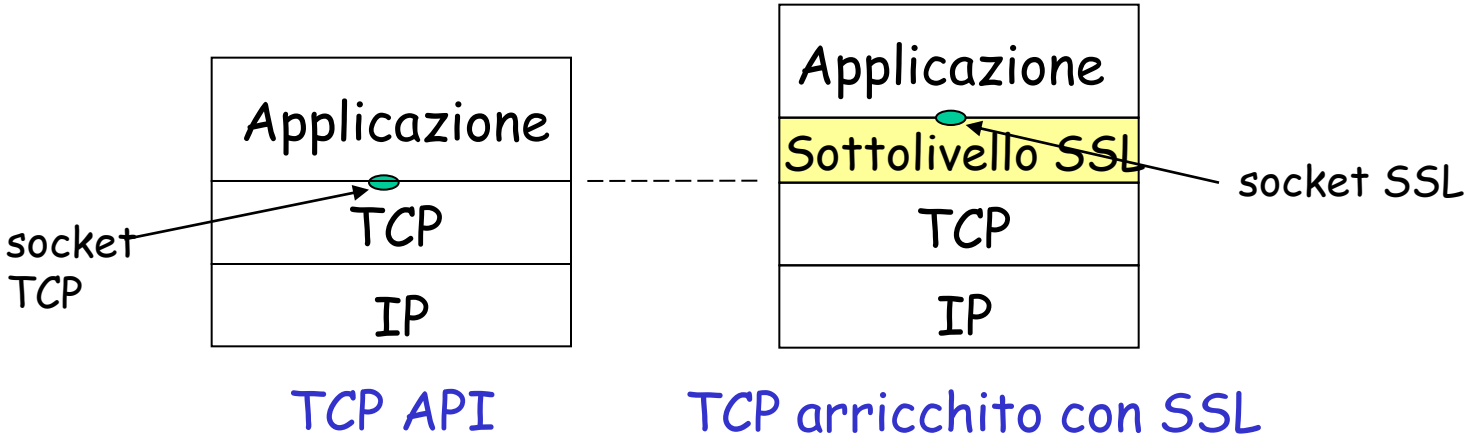
Sicurezza di rete
Principi di crittografia
Integrità dei messaggi
Autenticazione end-to-end
HTTPS

# HTTPS

- Garantisce **riservatezza**, **integrità** e **autenticazione** per le sessioni HTTP
- HTTPS usa il Secure Socket Layer (SSL) o Trasport Layer Security (TLS) per realizzare comunicazioni sicure
- Usa la porta 443
- RFC 2246 (prima versione)

# Livello di socket sicura (SSL)

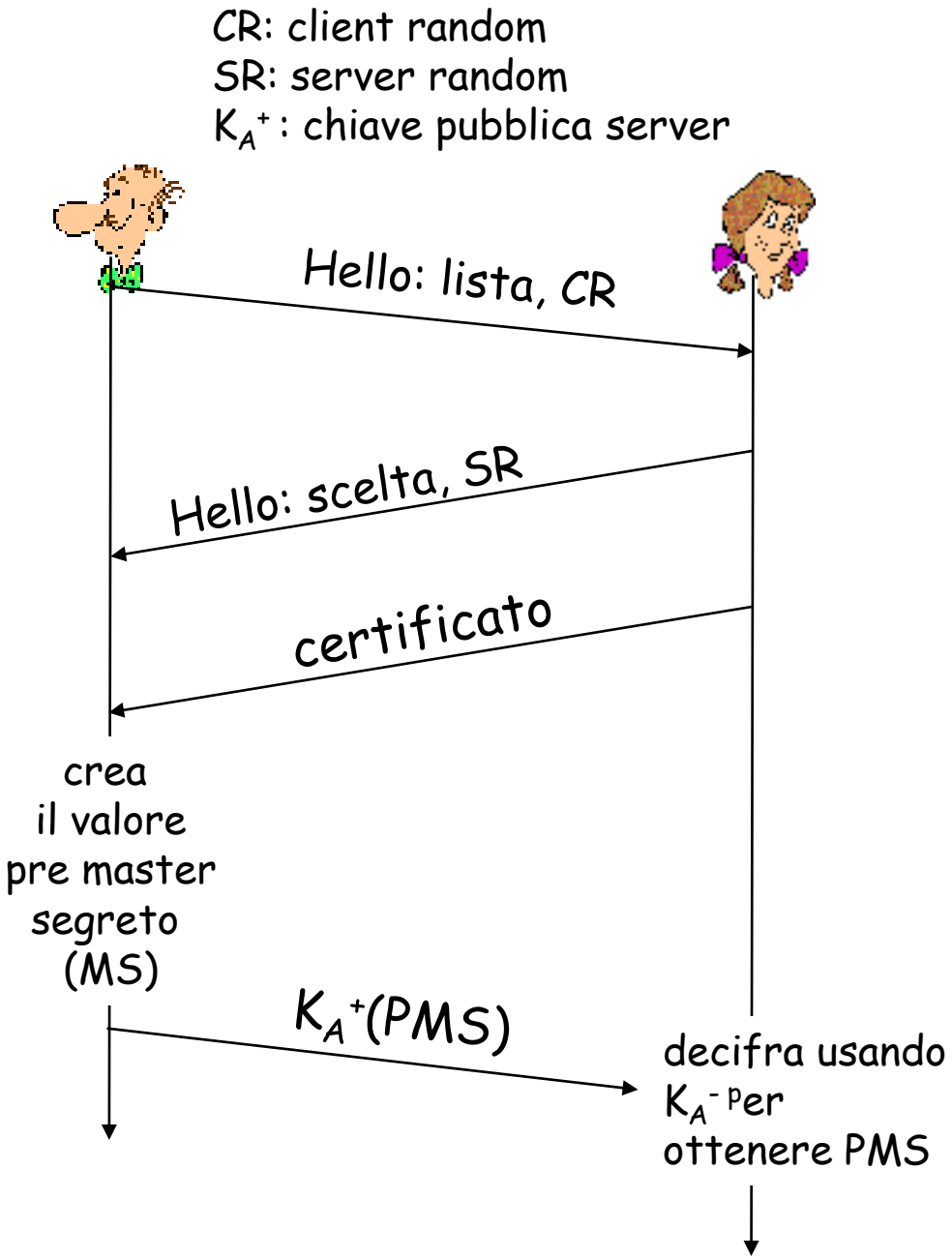
- Costituisce la base del protocollo di sicurezza a livello di trasporto.
  - Ampiamente utilizzato nelle transazioni commerciali e finanziarie su Internet (http)
- Servizi di sicurezza:
  - Autenticazione del server, cifratura dei dati, autenticazione del client (opzionale)



# SSL: tre fasi

*1. Handshake:*

- Scopo: far accordare client e server sui meccanismi di sicurezza da usare e su un valore segreto PMS (pre master secret) che servirà per la generazione delle chiavi
- Client invia Hello con lista di meccanismi di sicurezza e numero random (CR)
- Server risponde Hello contenente meccanismi di sicurezza scelti e numero random CR
- Server invia proprio certificato firmato dalla CA contente chiave pubblica
- Client genera PMS e lo invia cifrato mediante crittografia asimmetrica al server



# SSL: tre fasi

## *2. Derivazione delle chavi:*

- Il PMS è noto sia al client (che lo ha generato) che al server (che lo ha ricevuto cifrato dal client)
- Il PMS viene usato (da server e client) insieme ai due numeri random CR e SR per generare il master secret (MS), mediante una funzioe pseudorandom
- Il segreto condiviso MS viene usato (da server e client) per generare 4 chiavi
  - $E_C$: chiave di cifratura di sessione per i dati inviati da client a server
  - $E_S$: chiave di cifratura di sessione per i dati inviati da server a client
  - $M_S$: chiave MAC di sessione per i dati inviati da server a client
  - $M_C$: chiave MAC di sessione per i dati inviati da client a server

# SSL: tre fasi

## 3. Trasferimento dati