



*Corso di Laurea Triennale in Informatica  
Università degli Studi della Basilicata*

# Reti di Calcolatori

Docente: Ugo Erra

*ugo.erra+reti@unibas.it*

16° Lezione – La sicurezza nelle reti

# Obiettivi



- Identificare le proprietà per una comunicazione sicura:
  - Tecniche crittografiche
  - Autenticazione
  - Integrità del messaggio
  - Distribuzione delle chiavi

# Sommario



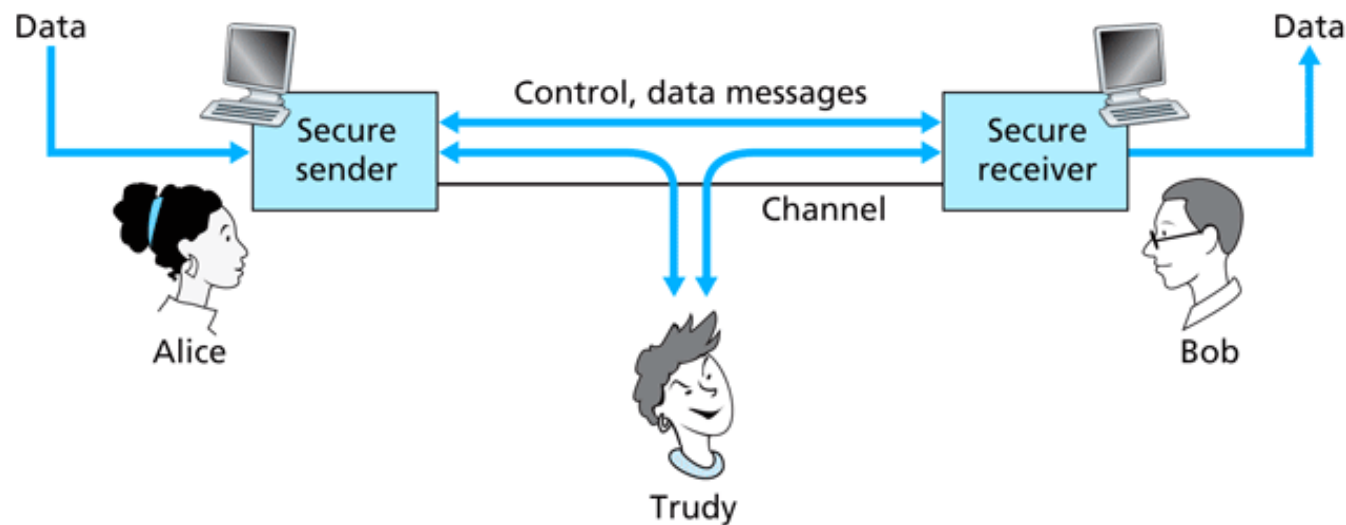
- **Sicurezza di rete**
- Principi di crittografia
- Integrità
- Distribuzione e certificazione delle chiavi

# Sicurezza nella comunicazione

- *Riservatezza*: 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
  - Client/Server di banche on-line
  - Server DNS
  - Sistemi che si scambiano tabelle d'instradamento altro

# Il mondo è un brutto posto



- Cosa può fare un nemico? Molto!
  - ▣ Spiare e 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!

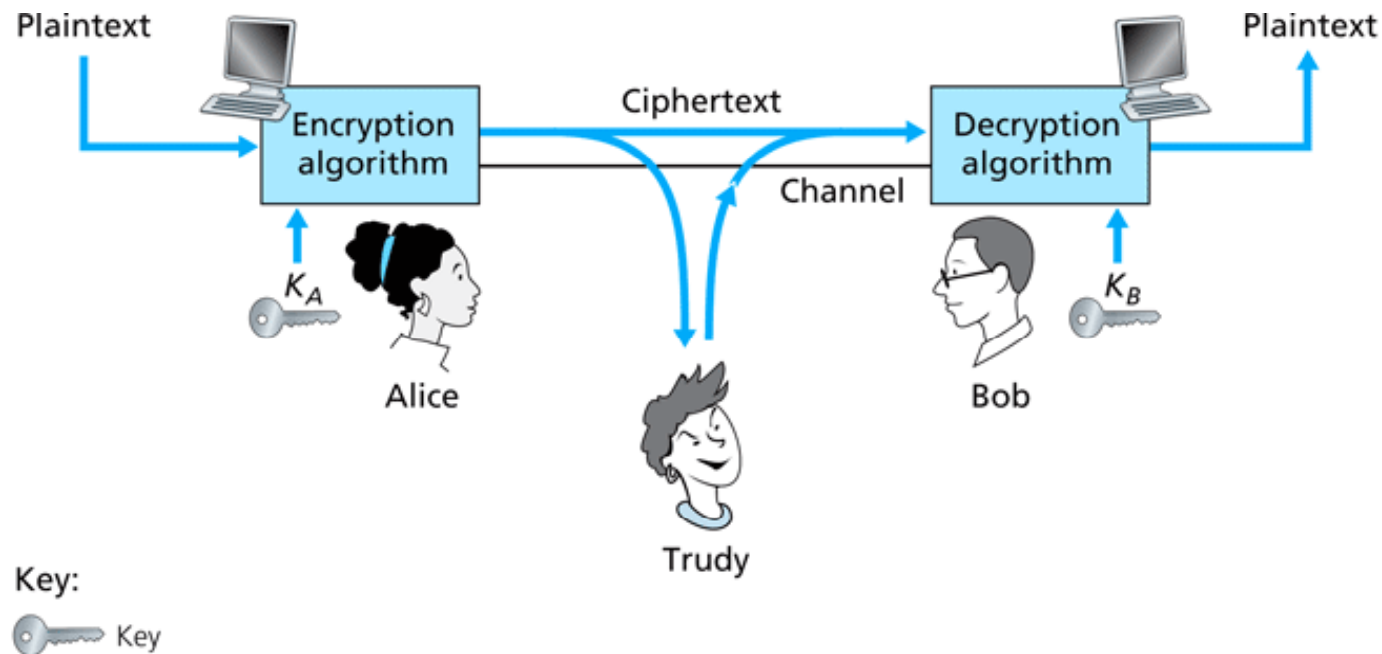
# Sommario



- Sicurezza di rete
- **Principi di crittografia**
- Integrità
- Distribuzione e certificazione delle chiavi

# Principi di crittografia

- **Sistemi a chiave simmetrica:** le chiavi del mittente e del destinatario sono identiche
- **Sistemi a chiave pubblica:** la chiave di cifratura è pubblica; la chiave di decifratura è privata



# Crittografia a chiave simmetrica

- L'algoritmo di cifratura consiste nella sostituzione di un messaggio in chiaro con uno codificato
- Cifrario monoalfabetico: sostituzione di una lettera con un'altra

Lettere in chiaro: abcdefghijklmnopqrstuvwxyz  
Lettere cifrate: mnbvcxzasdfghjklpoiuytrewq

- Esempio:

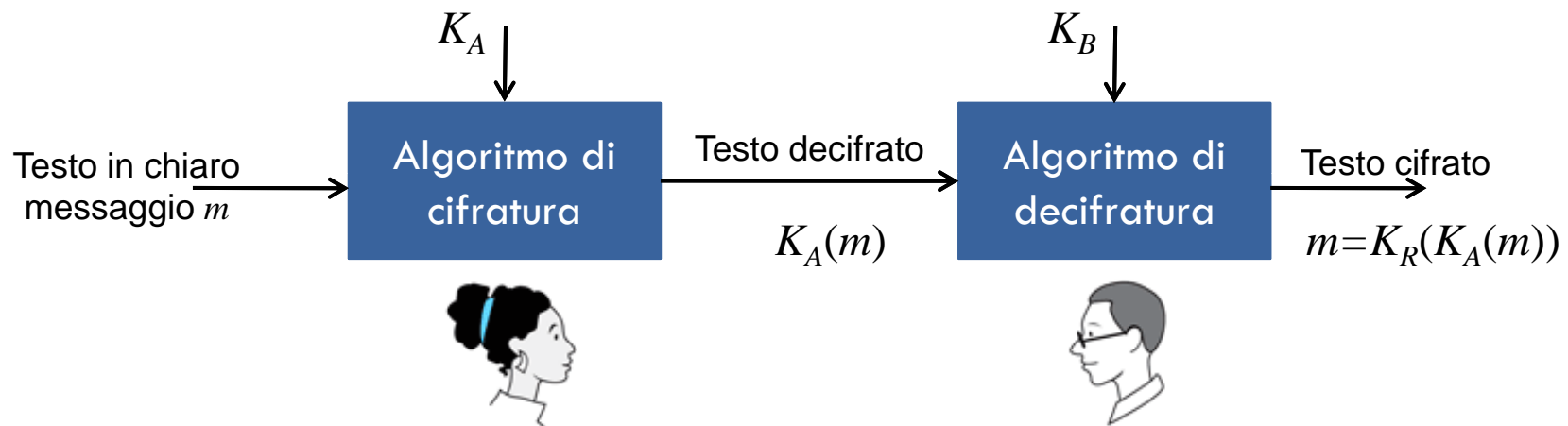
Testo in chiaro: bob. i love you. alice

Testo cifrato: nkn. s gktc wky. mgsbc

- Come violare questo sistema di cifratura?

# Crittografia a chiave simmetrica

- Nella **crittografia a chiave simmetrica** Alice e Roberto utilizzano la stessa chiave  $K_A = K_R$ 
  - ▣ Esempio: la chiave è un pattern di sostituzione monoalfabetico
- Come fanno Roberto e Alice a concordare la chiave?



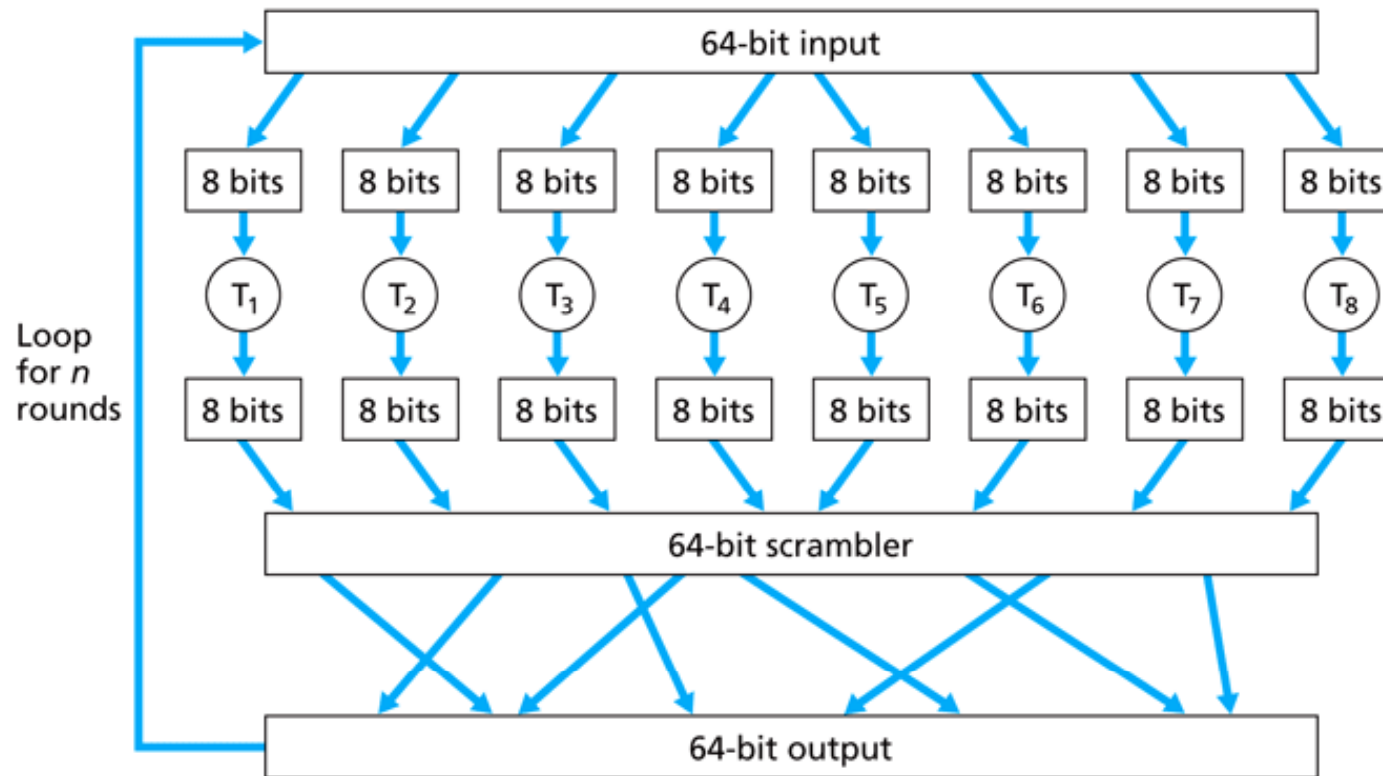
# Cifrari a blocchi

- In un cifrario a blocchi il messaggio è suddiviso in blocchi di  $k$  bit
  - ▣ Ad esempio se  $k=64$  il messaggio è diviso in blocchi da 64 bit
- Il cifrario usa una corrispondenza 1-a-1 per codificare un blocco di  $k$  bit
- Ad esempio supponiamo  $k=3$

Ingresso	Uscita	Ingresso	Uscita
000	110	100	011
001	111	101	010
010	101	110	000
011	100	111	001

- Quante corrispondenze sono possibili?
  - Numero degli ingressi  $2^3=8$
  - Numero di permutazioni  $8!=40320$

# Cifrario a blocchi



# Crittografia a chiave simmetrica DES

- DES (Data Encryption Standard) è lo standard codificato e aggiornato dall'U.S. National Bureau of Standards [NIST 1993]
- Codifica il testo in chiaro in blocchi di 64 bit; la lunghezza effettiva della chiave è di 56 bit
- Quanto è sicuro DES?
- DES Challenge: nel 1997, durante un concorso, la frase “Strong cryptography makes the world a safer place” fu individuata in meno di 4 mesi
- Come rendere DES più sicuro:
  - Usare sequenzialmente tre chiavi (3DES, triplo DES)
  - Utilizzare il concatenamento dei blocchi cifrati

# AES (Advanced Encryption Standard)



- Nel novembre 2001 NIST ha annunciato il sostituto di DES: AES
  - AES processa i blocchi a 128 bit
  - Opera con chiavi a 128, 192 e 256 bit
- Si stima che un calcolatore può individuare una chiave DES a 56 bit in 1 sec.; invece per violare una chiave AES a 128 bit ci impiegherebbe 149 miliardi di anni

# Problemi della crittografia simmetrica



- Nella crittografia a chiave simmetrica si richiede che mittente e destinatario condividano una chiave segreta
- Problema: come si concorda la chiave segreta (specialmente se i due interlocutori non si sono mai “incontrati”)?

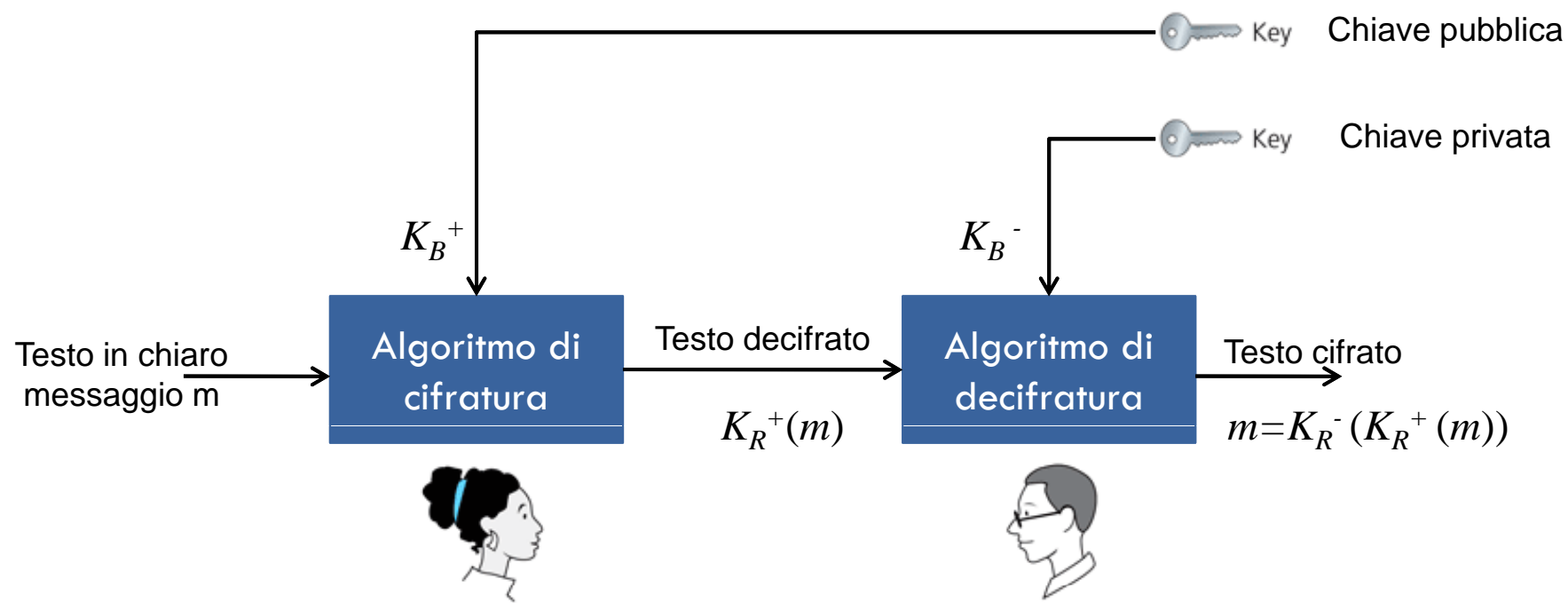
# Crittografia a chiave pubblica



- La **crittografia a chiave pubblica** è stata sviluppata da Diffie e Hellman nel 1976 utilizzando un approccio radicalmente diverso
- 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

- Roberto deve rendere pubblica la chiave  $K_R^+$  e conservare la chiave  $K_R^-$



# Algoritmi di cifratura a chiave pubblica

- Requisiti:
- $K_R^-(\cdot)$  e  $K_R^+(\cdot)$  devono essere scelti in modo tale che

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

- Data la chiave pubblica  $K_R^+$ , deve essere impossibile calcolare la chiave privata  $K_R^-$

# Algoritmo RSA

- L'**algoritmo RSA** è un acronimo derivato dal nome dei suoi autori Rivest, Shamir e Adelson
- Scelta delle chiavi:
  1. Scegliere due numeri primi di valore elevato:  $p, q$  ad esempio di 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 o coprimi)
  4. Scegliere  $d$  tale che  $ed-1$  sia esattamente divisibile per  $z$  (in altre parole:  $ed \bmod z = 1$ )
  5. La chiave pubblica è  $K_B^+ = (n, e)$
  6. La chiave privata è  $K_B^- = (n, d)$

# RSA: cifratura/decifratura

- Dati  $K_B^+ = (n, e)$  e  $K_B^- = (n, d)$  calcolati come abbiamo appena visto

- Per cifrare  $m$ , si calcola

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

- Per decifrare il messaggio ricevuto  $c$ , si calcola

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

- Ovvero:  $m = (m^e \bmod n)^d \bmod n$

# Un esempio di RSA

- Roberto sceglie  $p=5$  e  $q=7$ . Quindi  $n=5 \times 7=35$  e  $z=6 \times 4=24$
- Scegliamo  $e=5$  (così  $e, z$  sono relativamente primi)
- Scegliamo  $d=29$  (così  $ed-1$  è esattamente 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	4819685721067509150914118 25223071697	12	L

# Chiavi di sessione

- A causa dell'operazione di elevamento a potenza RSA richiede più tempo rispetto a DES
  - ▣ DES è 100 volte più veloce di RSA
- In pratica RSA è utilizzato congiuntamente a DES
  - ▣ Alice deve inviare dei dati a Roberto
  - ▣ Alice sceglie una **chiave di sessione**  $K_S$  da utilizzare durante la sessione DES
  - ▣ Alice invia  $K_S$  a Roberto mediante la chiave pubblica RSA di Roberto
  - ▣ Roberto riceve la chiave di sessione utilizzata da Alice

# Una importante proprietà di RSA

- Una proprietà particolarmente importante di RSA è la seguente:

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

↑  
Si usa prima la chiave pubblica, e poi quella privata

↑  
Si usa prima la chiave privata, e poi quella pubblica

- Il risultato non cambia!

# Efficacia di RSA



- L'efficacia di RSA si poggia sul fatto che non esistono algoritmi veloci per fattorizzare  $n$
- Conoscendo il valore  $n$  è computazionalmente proibitivo determinare i fattori primi  $p$  e  $q$ 
  - ▣ Dai quali si potrebbe calcolare la chiave privata segreta
- Inoltre non è noto se esista o meno un algoritmo veloce per la fattorizzazione di un numero

# Sulla fattorizzazione...

- La fattorizzazione in numeri primi consiste nel cercare un insieme di fattori del numero che siano tutti primi
  - ▣ La maggior parte dei numeri ha svariate fattorizzazioni possibili
  - ▣ Ogni numero naturale ha una ed una sola fattorizzazione in numeri primi
- Metodi di fattorizzazione:
  - ▣ Forza bruta
  - ▣ Metodo curve ellittiche
  - ▣ General Number Field Sieve
  - ▣ Metodo Peter Shor (veloce ma richiede un computer quantistico)

# Sommario



- Sicurezza di rete
- Principi di crittografia
- **Integrità**
- Distribuzione e certificazione delle chiavi

# Integrità dei messaggi



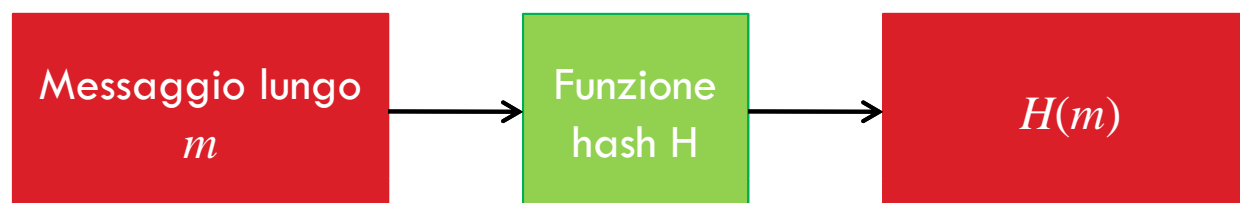
- La crittografia può essere utilizzata per garantire che un messaggio non è stato alterato lungo il suo tragitto
  - ▣ Ad esempio un router mentre scambia le proprie tabella con i vicini
- Supponiamo che Roberto riceva un messaggio inviato da Alice, il messaggio è autentico se:
  - ▣ Il messaggio è stato effettivamente originato da Alice
  - ▣ Non è stato alterato lungo il cammino verso Roberto

# Funzioni hash crittografiche

- Una **funzione hash**  $H$  prende in input un messaggio  $m$  e calcola una stringa di lunghezza fissa detta hash
- Una funzione hash crittografica deve soddisfare la seguente proprietà:

*E' impossibile trovare due messaggi  $x$  e  $y$  diversi tali che*  
$$H(x)=H(y)$$

- Ovvero un malintenzionato non può sostituire un messaggio con un altro messaggio che abbia la stessa funzione hash



# La checksum di Internet

- La checksum di Internet ha alcune delle proprietà di una funzione hash:
  - ▣ Crea sintesi di messaggi di lunghezza fissa
  - ▣ È multi-a-uno
- E' relativamente semplice trovare altri dati che utilizzano la stessa checksum del messaggio originale

Messaggio	Rappresentazione ASCII	Messaggio	Rappresentazione ASCII
I O U 1	49 4F 55 31	I O U <u>9</u>	49 4F 55 <u>39</u>
0 0 . 9	30 30 2E 39	0 0 . <u>1</u>	30 30 2E <u>31</u>
9 B O B	39 42 D2 42	9 B O B	39 42 D2 42
	<hr/>		<hr/>
	B2 C1 D2 AC		B2 C1 D2 AC

Messaggi diversi  
ma checksum identica!

# Algoritmi per le funzioni hash



- MD5 è un algoritmo molto utilizzato nella *sintesi* dei messaggi (RFC 1321)
  - ▣ Calcola una sintesi di 128 bit con un processo a quattro fasi.
  - ▣ Non è ancora stato verificato se MD5 soddisfa i requisiti di autenticazione
- SHA-1 è un altro importante algoritmo di sintesi
  - ▣ Standard federale statunitense [NIST, FIPS PUB 180-1]
  - ▣ Produce una sintesi del messaggio più lunga: 160 bit

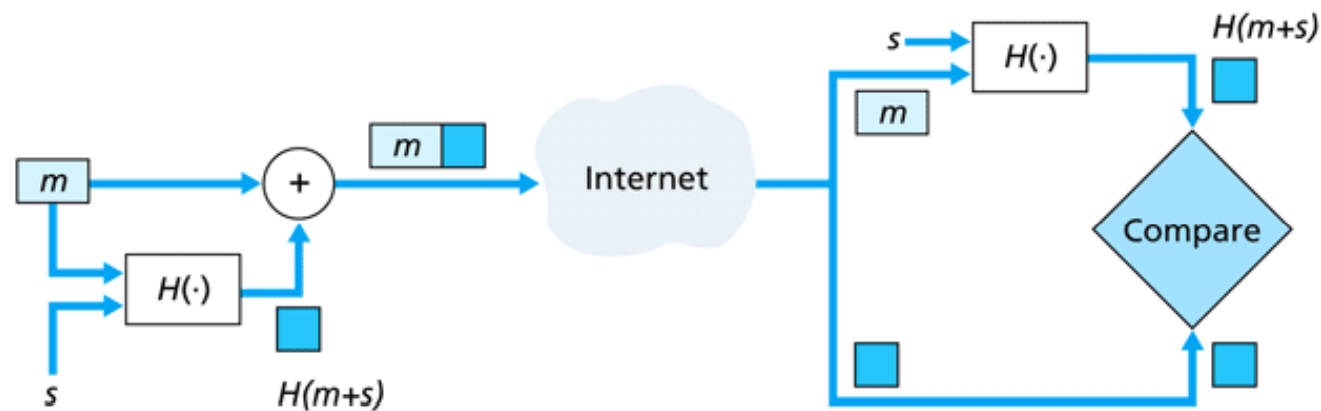
# Integrità di un messaggio - 1

1. Alice crea un messaggio  $m$  e calcola la stringa hash  $H(m)$  ad esempio con SHA-1
  2. Alice aggiunge  $H(m)$  al messaggio  $m$ , crea il messaggio esteso  $(m, H(m))$  e lo manda a Roberto
  3. Roberto riceve il messaggio  $(m, H(m))$  e calcola  $H(m)$ .  
Se  $H(m)=m$  Roberto conclude che va tutto bene
- Tommaso può creare un messaggio falso  $m'$ , calcola  $H(m')$  e manda a Roberto  $(m', H(m'))$  dicendo di essere Alice

# Integrità di un messaggio - 2

- Per realizzare l'integrità abbiamo bisogno di un segreto condiviso chiamato **chiave di autenticazione**  $s$
- Alice crea un messaggio  $m$  e concatena  $s$  con  $m$ .  
Calcola la stringa hash  $H(m+s)$  chiamato **codice di autenticazione messaggio** o MAC
- Alice aggiunge il MAC al messaggio  $m$ , crea il messaggio esteso  $(m, H(m+s))$  e lo manda a Roberto
- Roberto riceve il messaggio  $(m, H(m+s))$  e calcola  $H(m+s)$ . Se  $H(m+s)=m$  Roberto conclude che va tutto bene

# Codice di autenticazione del messaggio



Key:

$m$  = Message

$s$  = Shared secret

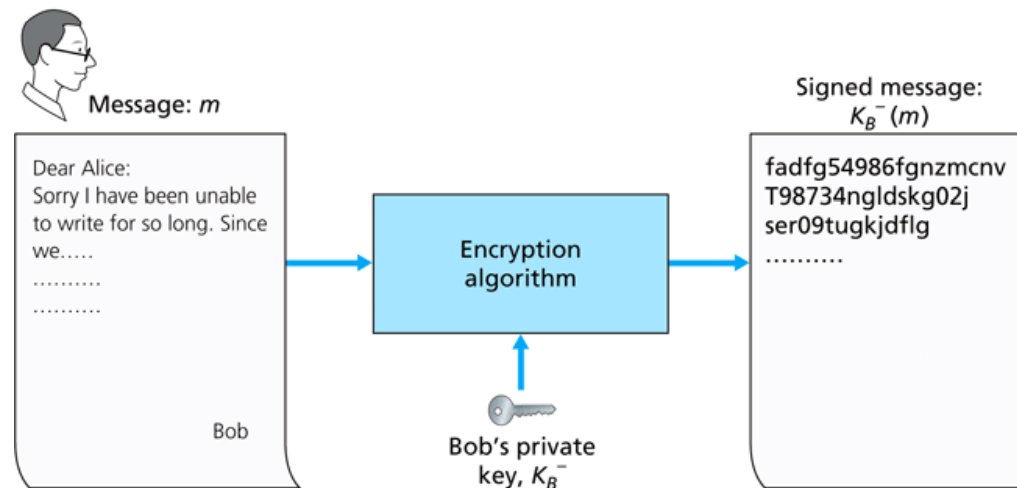
# Firme digitali



- La **firma digitale** permette di dimostrare che un certo documento:
  - ▣ E' stato inviato esattamente da un data persona (verifica)
  - ▣ Solo quella persona poteva realizzarlo (non falsificabile)
- E' sufficiente utilizzare il **MAC** per realizzare la firma digitale?

# Firma digitale con chiave privata

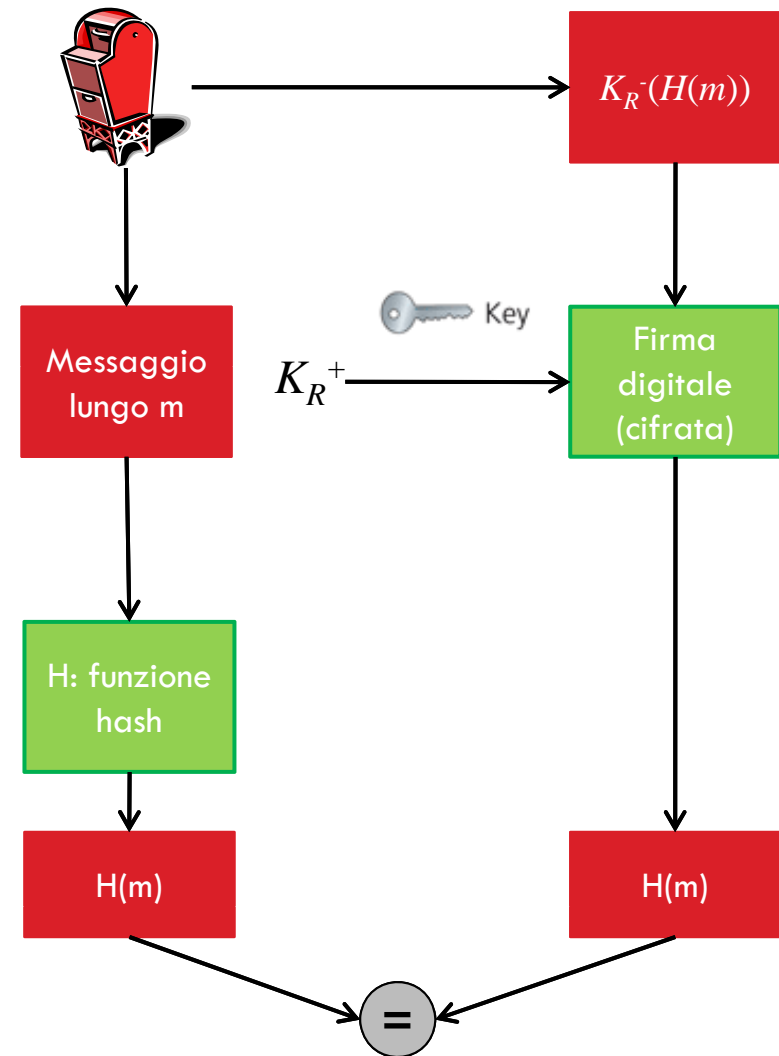
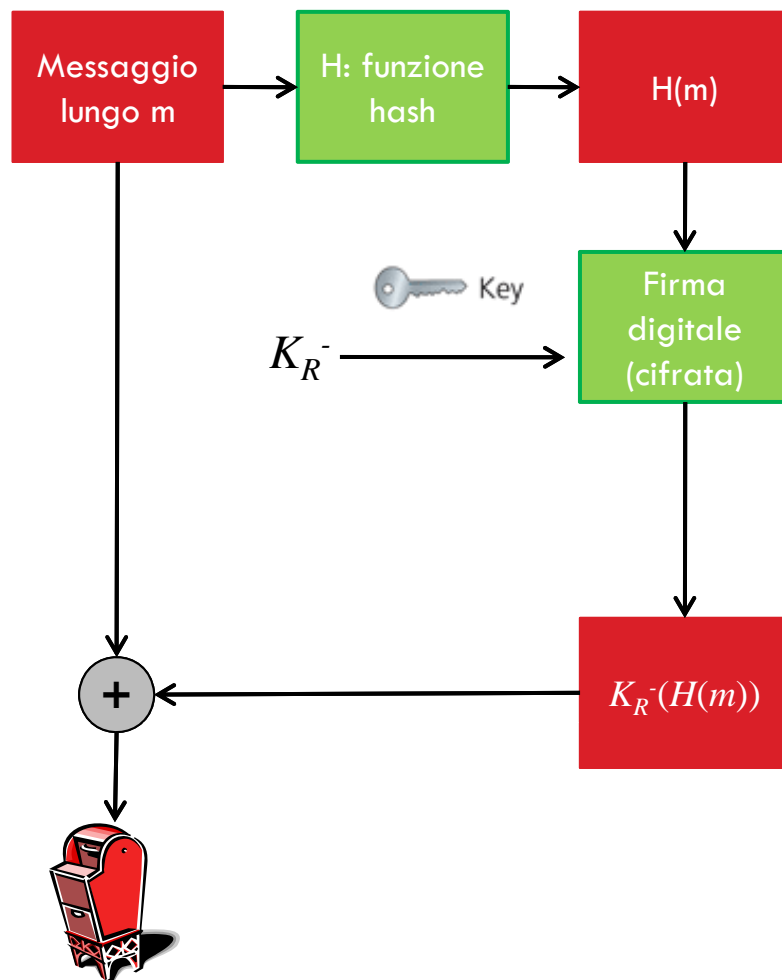
- Per firmare un documento Roberto può utilizzare la propria chiave privata  $K_R^-(m)$
- Roberto invia ad Alice  $(m, K_R^-(m))$
- Alice utilizza la chiave pubblica di Roberto per calcolare  $K_R^+(K_R^-(m)) = m$



# Perché Alice può dormire tranquilla?

- Alice può verificare che:
  - ▣ Roberto ha firmato  $m$
  - ▣ Nessun altro ha firmato  $m$
  - ▣ Roberto ha firmato  $m$  e non un eventuale  $m'$
- Non-ripudio:
  - ▣ Alice può prendere  $m$ , e la firma  $K_R^-$  per dimostrare che Roberto ha firmato  $m$
- Problema: la cifratura e la decifratura è onerosa dal punto di vista computazionale

# Firma digitale = messaggi digest firmati



# Sommario

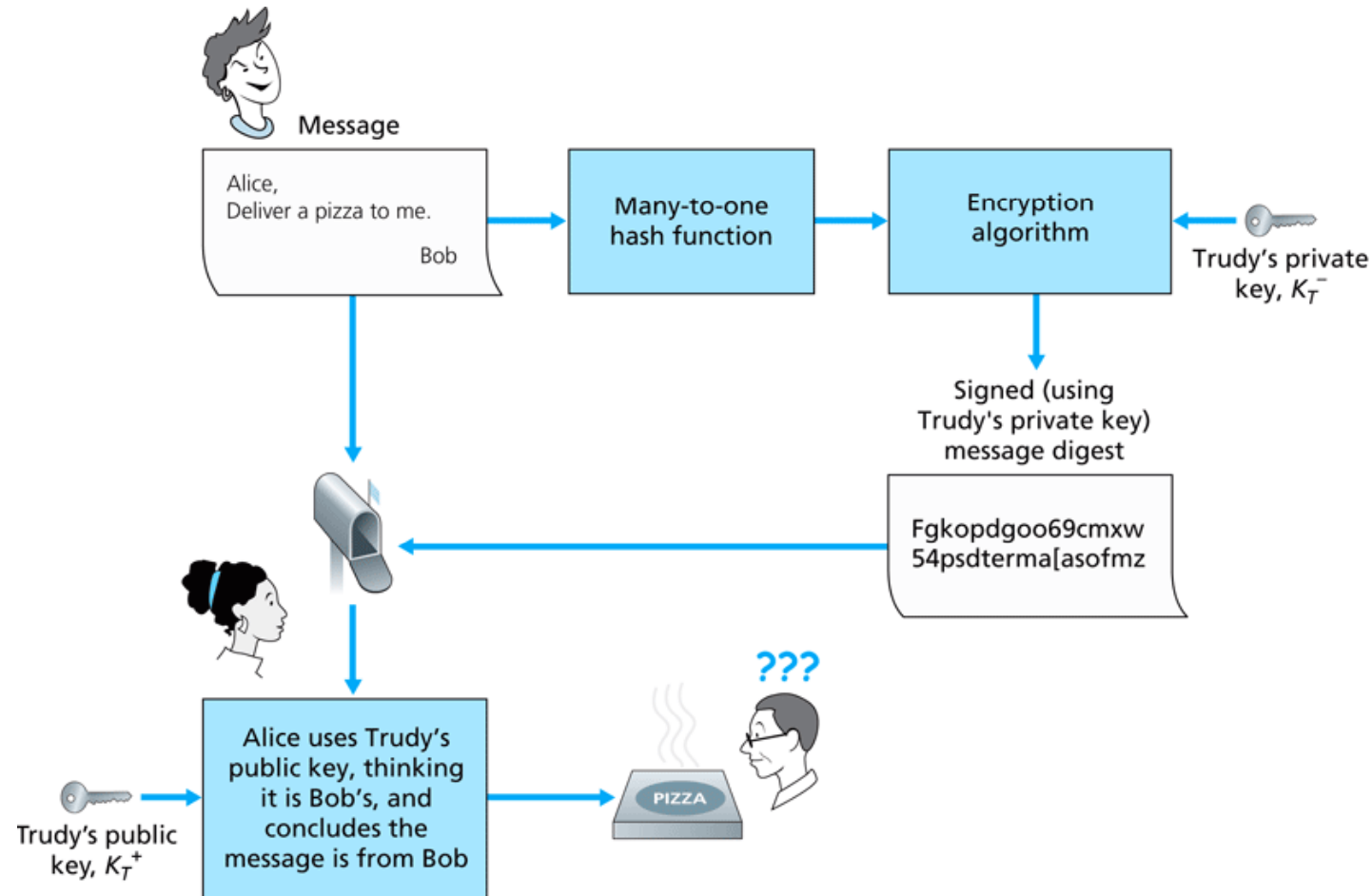


- Sicurezza di rete
- Principi di crittografia
- Integrità
- **Distribuzione e certificazione delle chiavi**

# Intermediario di fiducia

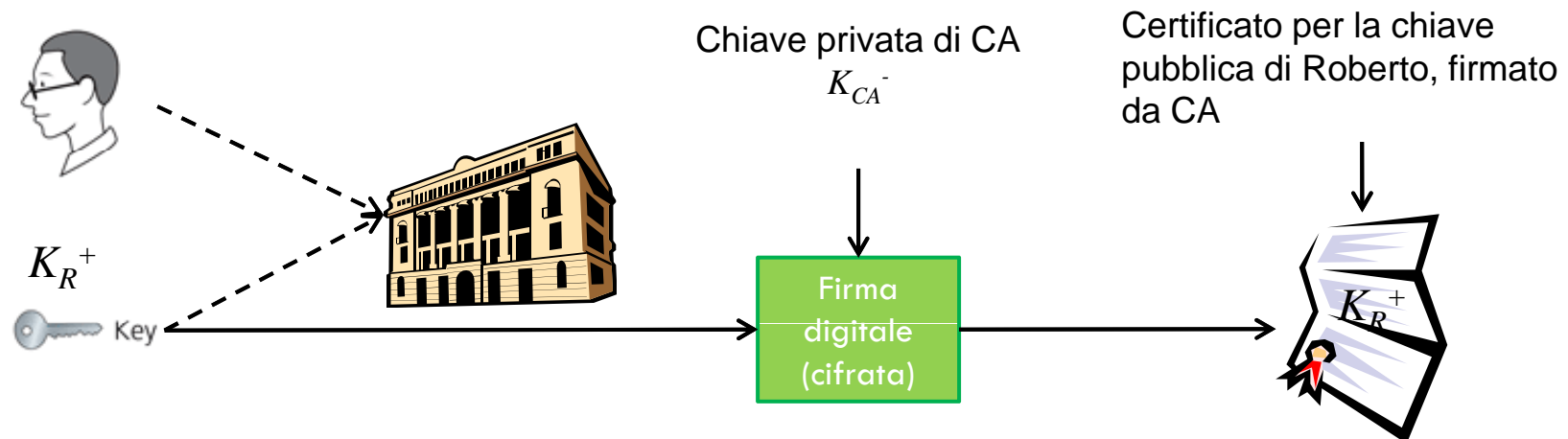
- Problema per la crittografia a chiave simmetrica:
  - ▣ Come possono le due parti concordare le chiavi prima di comunicare?
    - Un centro di distribuzione delle chiavi (KDC, key distribution center) di fiducia funge da intermediario tra le due entità
- Problema per la crittografia a chiave pubblica:
  - ▣ 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?
    - Autorità di certificazione (CA, certification authority)

# Tommaso finge di essere Roberto



# Autorità di certificazione

- L'**autorità di certificazione** (CA) collega una chiave pubblica a una particolare entità, E
- E può essere una persona fisica, router che 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”)



# Come fa Alice ad essere sicura di Roberto?

- 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

