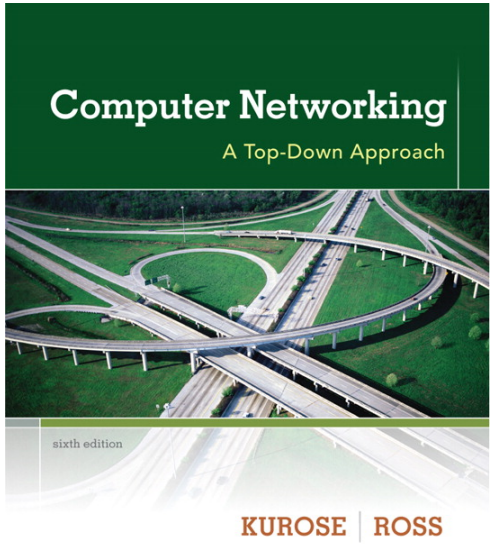
# Chapter 8
# Security




Roberto Petroccia
petroccia@di.uniroma1.it

Computer Networking: A Top Down Approach
6th edition
Jim Kurose, Keith Ross
Addison-Wesley
March 2012

- Queste slide sono un adattamento delle slide fornite dal libro di testo e pertanto protette da copyright.
All material copyright 1996-2012 J.F Kurose and K.W. Ross, All Rights Reserved
- Materiale aggiuntivo fornito dalla Prof.ssa Gaia Maselli e preso in rete da:
  - Sikurezza.org - Enrico Zimuel

# La sicurezza nelle reti

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

# Introduzione

- La protezione delle informazioni ha subito negli ultimi decenni un radicale cambiamento
- Classicamente, tale protezione era garantita mediante strumenti fisici (serrature, casseforti, etc.)
- L'avvento dei computer ha richiesto l'introduzione di strategie alternative per la protezione delle informazioni.
- In più, l'uso delle reti richiede strumenti che permettano di proteggere le informazioni durante la loro trasmissione

# Definizioni

*Computer Security - Nome generico che indica l'insieme dei tool utilizzati per proteggere i dati immagazzinati nei computer*

*Network Security - Insieme di misure utilizzate per proteggere i dati durante la loro trasmissione, ad esempio su una LAN o su una rete telefonica*

*Internet Security - Insieme di misure utilizzate per proteggere i dati durante la loro trasmissione sulla rete Internet*

# Cosa si intende per network security?

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

- Mittente cifra il messaggio
- Ricevente decifra il messaggio

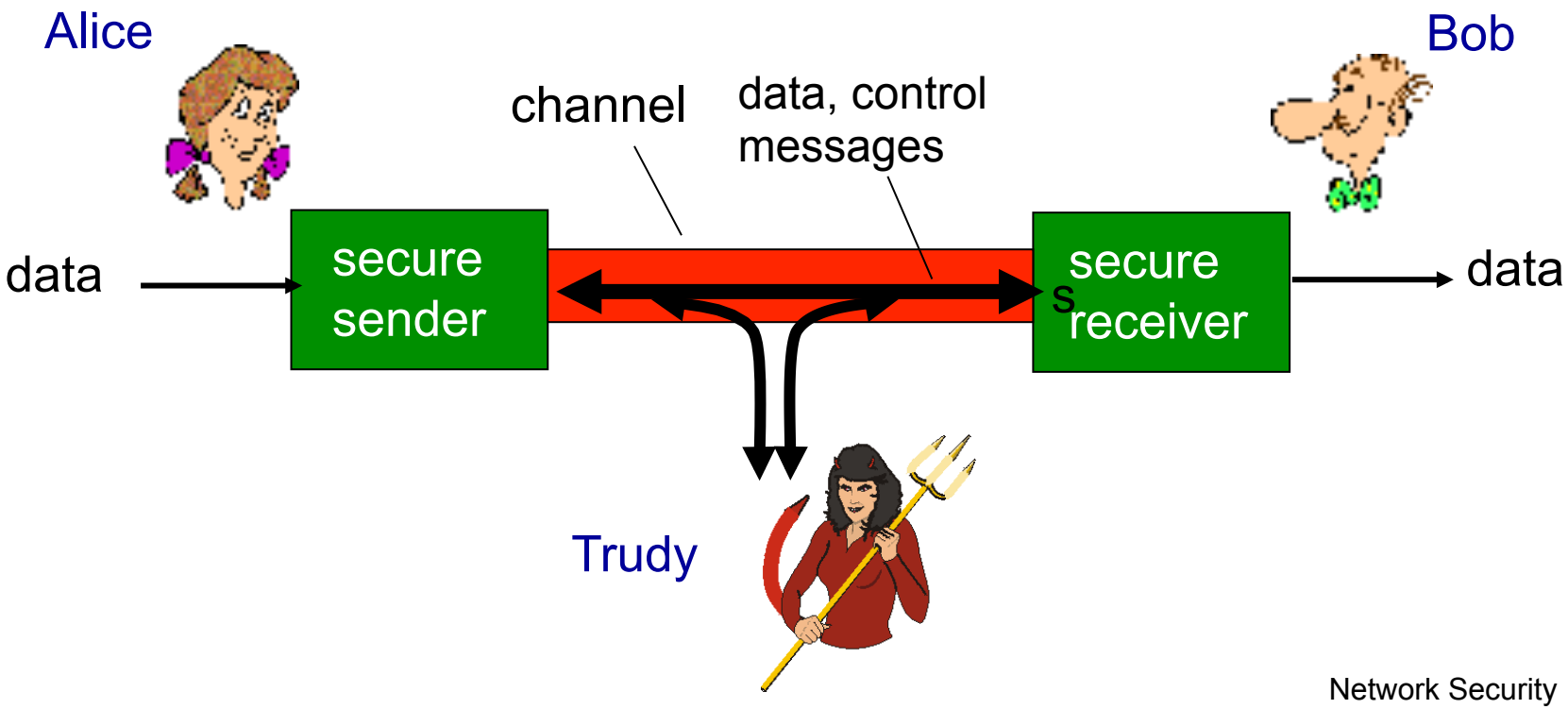
Autenticazione: mittente e destinatario devono essere sicuri della loro identità

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, Bob, Trudy

- Scenario ben noto nel mondo della sicurezza di rete
- Bob, Alice vogliono ocmunicare in maniera sicura
- Trudy (intruso) può intercettare, rimuovere, aggiungere messaggi o modificare il loro contenuto



# Chi possono essere Bob ed Alice?

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
- *Causare denial of service - causando disservizi (tramite un sovraccarico della rete e delle risorse)*

*E molto altro ancora! .....*

# Chapter 8 roadmap

8.1 What is network security?

*8.2 Principles of cryptography*

8.3 Message integrity, authentication

8.4 Securing e-mail

8.5 Securing TCP connections: SSL

8.6 Network layer security: IPsec

8.7 Securing wireless LANs

8.8 Operational security: firewalls and IDS

# Che cos'è la crittografia?

- La crittografia (dal greco kryptos, nascosto, e graphein, scrivere). È la scienza che si occupa dello studio delle scritture "segrete".
- È nata come branca della matematica e dell'informatica grazie all'utilizzo di tecniche di teoria dei numeri e di teoria dell'informazione.
- La crittografia è una scienza in continua espansione, a causa delle veloci e repentine evoluzioni tecnologiche

# Origini storiche

- La crittografia è una scienza antichissima utilizzata nell'antichità per nascondere messaggi tra regnanti, imperatori, nobili.
- La **scitala lacedemonica** è un antico esempio di un sistema per cifrare messaggi tramite l'utilizzo di un bastone cilindrico, cifrario a trasposizione (secondo gli scritti di Plutarco, in uso dai tempi di Licurgo, IX sec a.C.).
- Il periodo díoro della crittologia è relativo alla seconda guerra mondiale quando **Alan Turing**, il padre dell'informatica teorica, insieme al gruppo di ricerca del Bletchley Park formalizzò la matematica necessaria per uno studio sistematico dei cifrari.

# Origini storiche

- **Enigma** è una delle macchine cifranti più famose della seconda guerra mondiale.
- **Claude Shannon**, l'ideatore della moderna teoria dell'informazione, che nel 1949 pubblicò un articolo rimasto nella storia "Communication theory of secrecy systems".
- Nasce nel 1943 in Inghilterra il primo elaboratore elettronico il **Colossus** utilizzato per decifrare le comunicazioni "segrete" dei nemici.

# La moderna crittografia

- Le basi teoriche della moderna crittografia, quella attualmente utilizzata, sono ancora più giovani e risalgono a circa 30 anni fa a partire dal 1969 con le prime ricerche di **James Ellis** del quartier generale governativo delle comunicazioni britanniche (GCHQ).
- Sviluppata ed affinata nel 1976 in America grazie al contributo di **Whitfield Diffie** e **Martin Hellman** con la nascita del termine crittografia a chiave pubblica.
- Nasce nel 1977 il cifrario a chiave pubblica **RSA** da tre ricercatori del MIT (Massachusetts Institute of Technology), **Ronald Rivest, Adi Shamir e Leonard Adleman**.
- Con il cifrario RSA si inizia a parlare di strong-encryption, crittografia forte.

# Le operazioni di cifratura e decifrazione

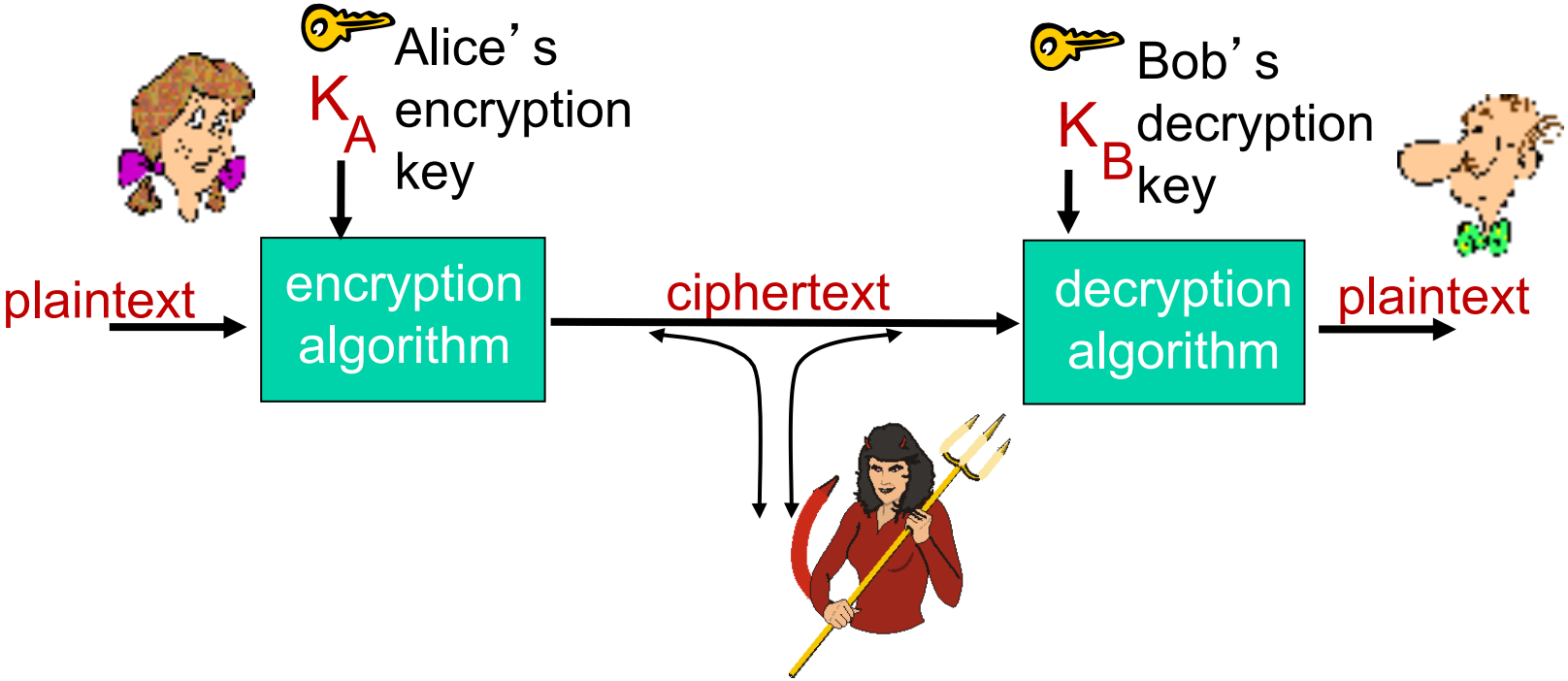
- Definiamo con **Msg** "l'insieme di tutti i messaggi" e con **Critto** "l'insieme di tutti i crittogrammi".
- **Cifratura**: operazione con cui si trasforma un generico messaggio in chiaro **m** in un crittogramma **c** applicando una funzione **C: Msg->Critto**.
- **Decifrazione**: operazione che permette di ricavare il messaggio in chiaro **m** a partire dal crittogramma **c** applicando una funzione **D: Critto -> Msg**.
- Matematicamente D(C(m))=m le funzioni C e D sono una inversa dell'altra e la funzione C deve essere iniettiva, ossia a messaggi diversi devono corrispondere crittogrammi diversi.

# Che cos'è un cifrario?

- Un cifrario è un sistema, di qualsiasi tipo, in grado di trasformare un testo in chiaro (messaggio) in un testo inintellegibile (testo cifrato o crittogramma).

- Per poter utilizzare un cifrario è necessario definire due operazioni: la cifratura del messaggio e la decifrazione del crittogramma.

# The language of cryptography




m plaintext message

$K_A(m)$ ciphertext, encrypted with key $K_A$

$m = K_B(K_A(m))$

# Breaking an encryption scheme

- cipher-text only attack: Trudy has ciphertext she can analyze
- two approaches:
  - brute force: search through all keys
  - statistical analysis
- known-plaintext attack: Trudy has plaintext corresponding to ciphertext
  - e.g., in monoalphabetic cipher, Trudy determines pairings for a,l,i,c,e,b,o,
- chosen-plaintext attack: Trudy can get ciphertext for chosen plaintext

# Breaking an encryption scheme

- cipher-text only attack: Trudy has ciphertext she can analyze
- two approaches:
  - brute force: search through all keys
  - statistical analysis

Possibili combinazioni per gli attacchi di forza bruta

| Lunghezza chiave (in bit) | Numero combinazioni possibili |
|---|---|
| 40 | $2^{40}$ = 1.099.511.627.776 |
| 56 | $2^{56}$ = 7,205759403793 x $10^{16}$ |
| 64 | $2^{64}$ = 1,844674407371 x $10^{19}$ |
| 112 | $2^{112}$ = 5,192296858535 x $10^{33}$ |
| 128 | $2^{128}$ = 3,402823669209 x $10^{38}$ |

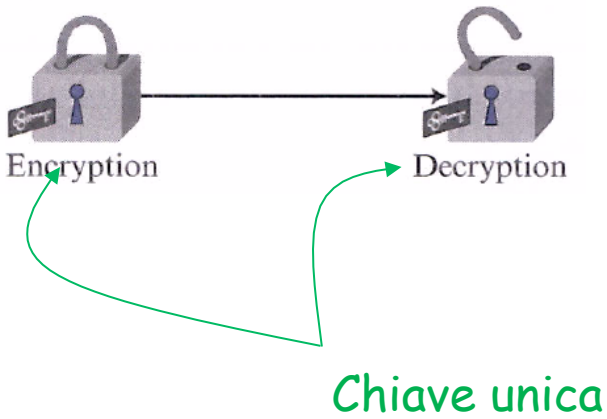
- known-plaintext attack: Trudy has plaintext corresponding to ciphertext
  - e.g., in monoalphabetic cipher, Trudy determines pairings for a,l,i,c,e,b,o,
- chosen-plaintext attack: Trudy can get ciphertext for chosen plaintext

# Diverse tecniche di crittografia

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)

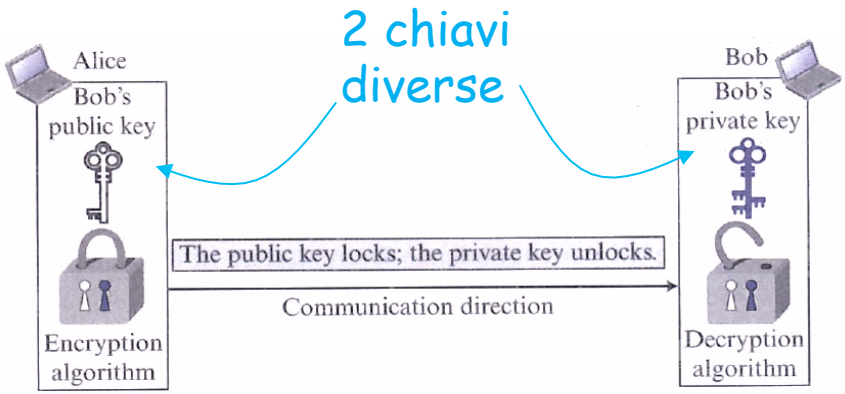
- Crittografia a chiave monoalfabetica
- Crittografia a chiave polialfabetica



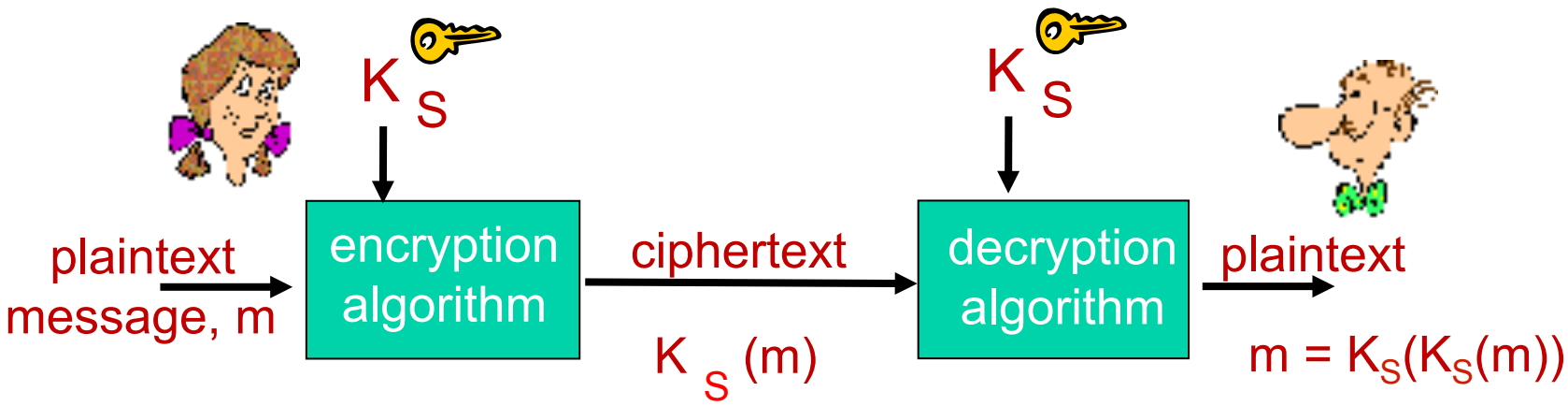

Sistemi a chiave asimmetrica:

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



# Symmetric key cryptography




Per **cifrare** il messaggio è necessario:

- Algoritmo di cifratura
- Chiave simmetrica condivisa

Per **decifrare** il messaggio è necessario:

- Algoritmo di decifratura
- Chiave simmetrica condivisa

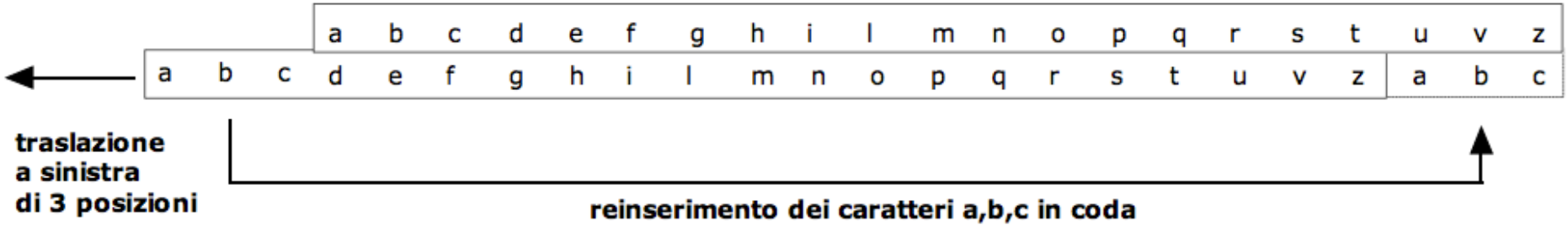
N.B. La chiave simmetrica **condivisa** è **segreta** e deve essere concordata priori (come viene stabilita???)

# Simple encryption scheme

## *Cifrario di Cesare*

Consideriamo l'alfabeto italiano, costruiamo un cifrario che sostituisce ad ogni lettera di questo alfabeto la lettera che si trova 3 posizioni in avanti.

**Cifrario di Cesare**




Ad esempio il testo in chiaro "prova di trasmissione" viene cifrato nel crittogramma "surbd gn zudvpnvvnrqh".

*Encryption key:* mapping from set of 26 letters to set of 26 letters

# Simple encryption scheme

*Cifrario di Cesare*

- Cifrario molto semplice ma all'epoca garantiva una buon grado di sicurezza visto che le persone in grado di leggere erano pochissime.

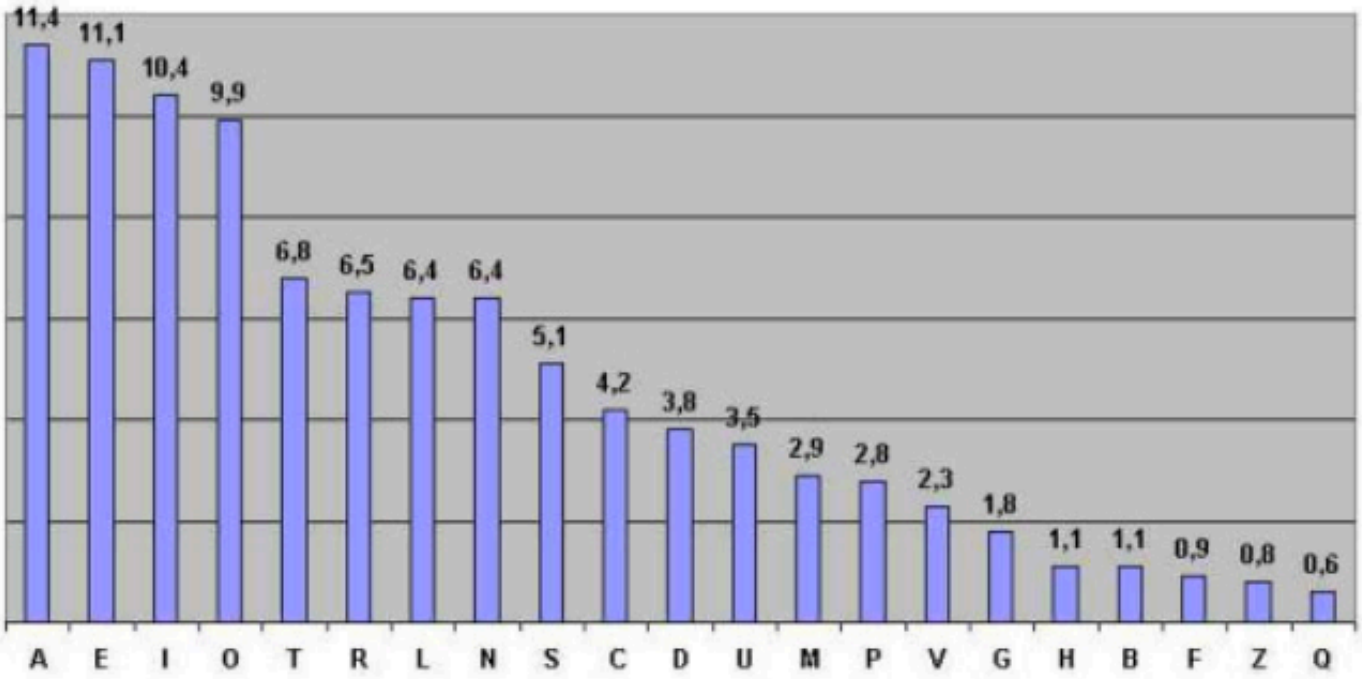
- Si è scoperto che il boss mafioso Provenzano utilizzava questo stesso sistema per cifrare i suoi "pizzini"

# Simple encryption scheme

## La crittoanalisi statistica

- Tramite l'utilizzo di tecniche statistiche sulla frequenze dei caratteri o sottostringhe del testo cifrato si ottengono informazioni utili sul testo in chiaro.

**Distribuzione in % delle lettere in un testo italiano**



# Sistema a chiave polialfabetica

- L'unica differenza rispetto alla chiave monoalfabetica è che il numero degli alfabeti utilizzati è sempre maggiore di uno. I metodi di modifica degli alfabeti rimangono gli stessi.

- Esempio più famoso è il Cifrario di Vigenere, la cui evoluzione è stata usata anche durante la Seconda Guerra Mondiale.

- Si basa sulla segretezza di due elementi: Il grado di spostamento (come per il cifrario di Cesare) ed il cosiddetto "versetto di decodifica" (o "verme")

# Sistema a chiave polialfabetica

- Metodo di codifica semplice: Si somma valore della lettera del messaggio in chiaro alla corrispondente lettere del versetto per determinare la codifica. Se si eccede la massima dimenzione dell'alfabeto si cicla (ripartendo dalla prima lettera).

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| C | I | V | E | D | I | A | M | O | S | T | A | S | E | R | A |
| V | E | R | S | E | T | T | O | V | E | R | S | E | T | T | O |

Risultato

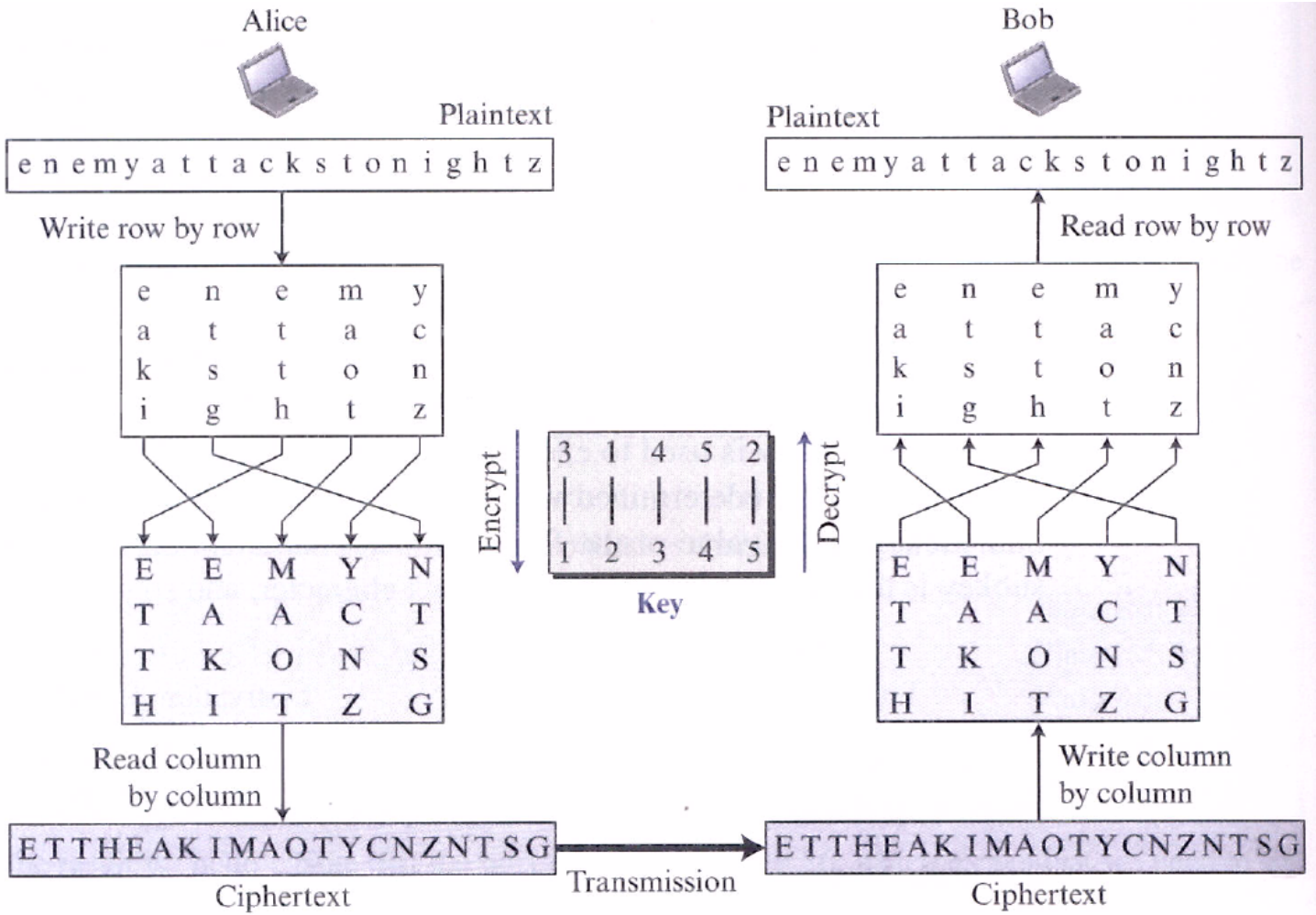
| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| X | M | M | W | H | B | T | A | J | W | K | S | W | X | K | O |

*Encryption key:* n substitution ciphers, and cyclic pattern

- key need not be just n-bit pattern

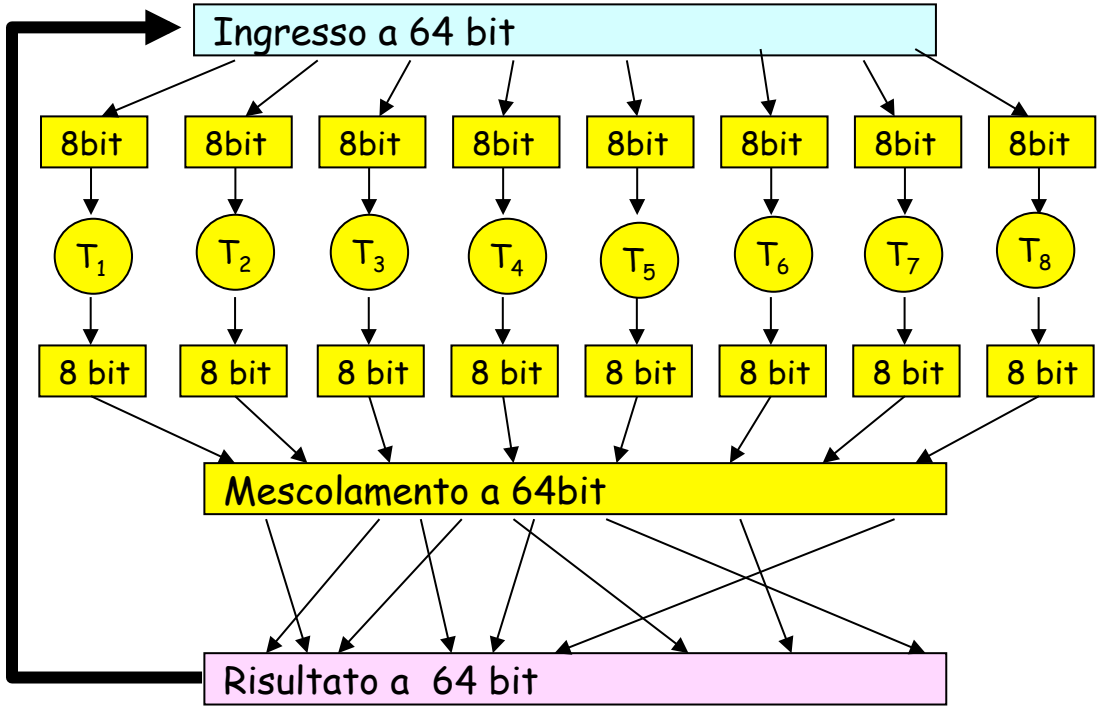
# Sistema per trasposizione

- Per trasposizione



# Cifrari 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

# Symmetric key crypto: DES

## DES: Data Encryption Standard

- US encryption standard [NIST 1993]
- 56-bit symmetric key, 64-bit plaintext input
- block cipher with cipher block chaining
- how secure is DES?
  - DES Challenge: 56-bit-key-encrypted phrase decrypted (brute force) in less than a day
  - no known good analytic attack
- making DES more secure:
  - 3DES: encrypt 3 times with 3 different keys

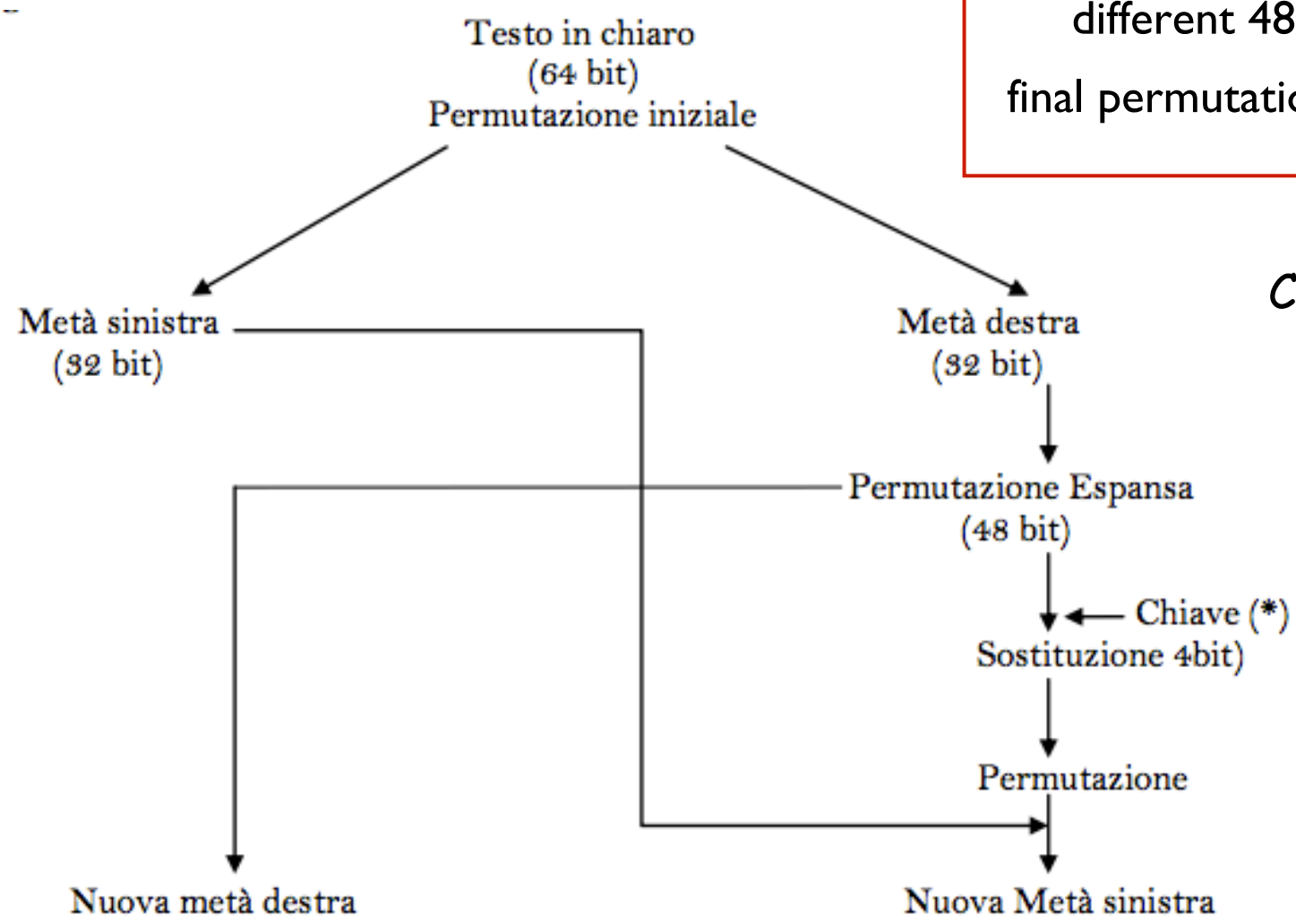
# Symmetric key crypto: DES

*DES operation*

initial permutation

16 identical "rounds" of function application, each using different 48 bits of key

final permutation



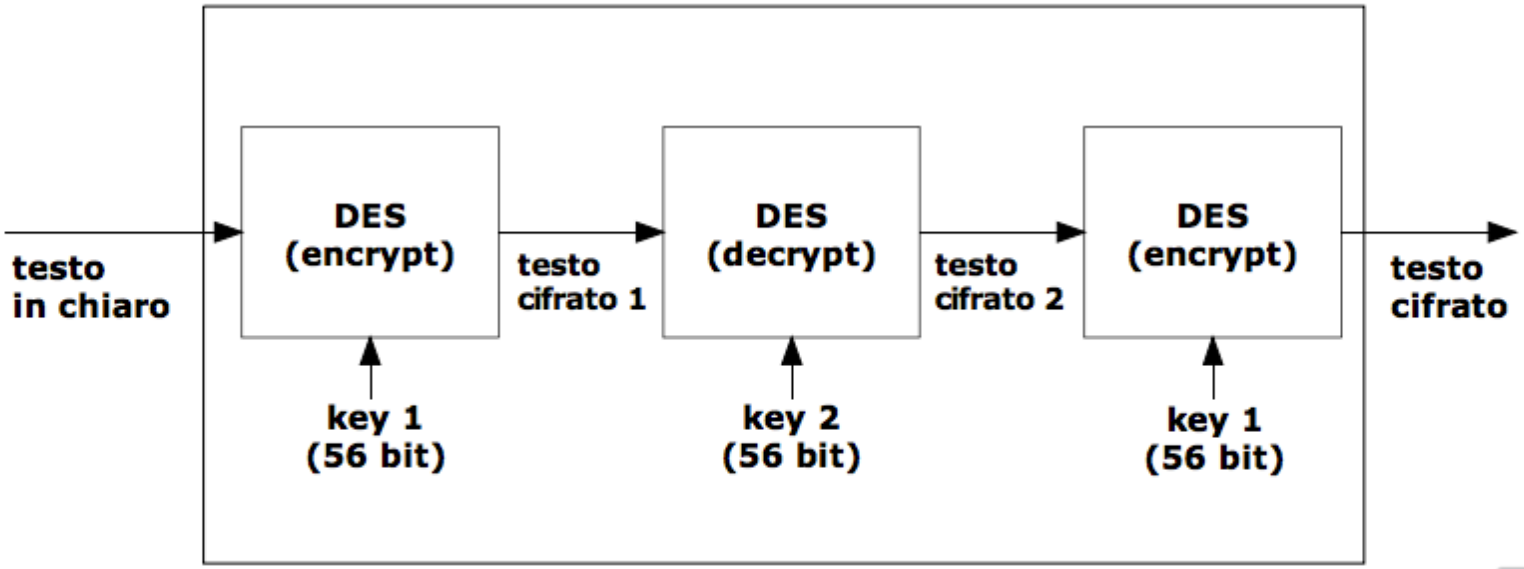

Chiave a 56 bit

Chiave(*) e' una chiave di 48 bit ottenuta dalla chiave originale con operazioni di shifting e permutazionia 56 bit

# Symmetric key crypto: 3DES

## 3DES

- Evoluzione del DES, è basato su un utilizzo del cifrario DES ripetuto, chiavi di 112 bit.
- Si utilizza la tecnica della codifica-decodifica-codifica (EDE, Encrypt-Decrypt-Encrypt) utilizzando il cifrario DES.




3DES (key = key1+key2, 112 bit)

# AES: Advanced Encryption Standard

- symmetric-key NIST standard, replacied DES (Nov 2001)
- processes data in 128 bit blocks
- 128, 192, or 256 bit keys
- brute force decryption (try each key) taking 1 sec on DES, takes 149 trillion years for AES

# Scambio delle chiavi

- Esiste davvero un modo sicuro per scambiare le chiavi tra mittente e destinatario?
- Per una comunicazione sicura tra n utenti si dovranno scambiare in tutto (n-1)*n/2 chiavi, ad esempio con 100 utenti occorreranno 4950

# Crittografia a chiave asimmetrica (o pubblica)

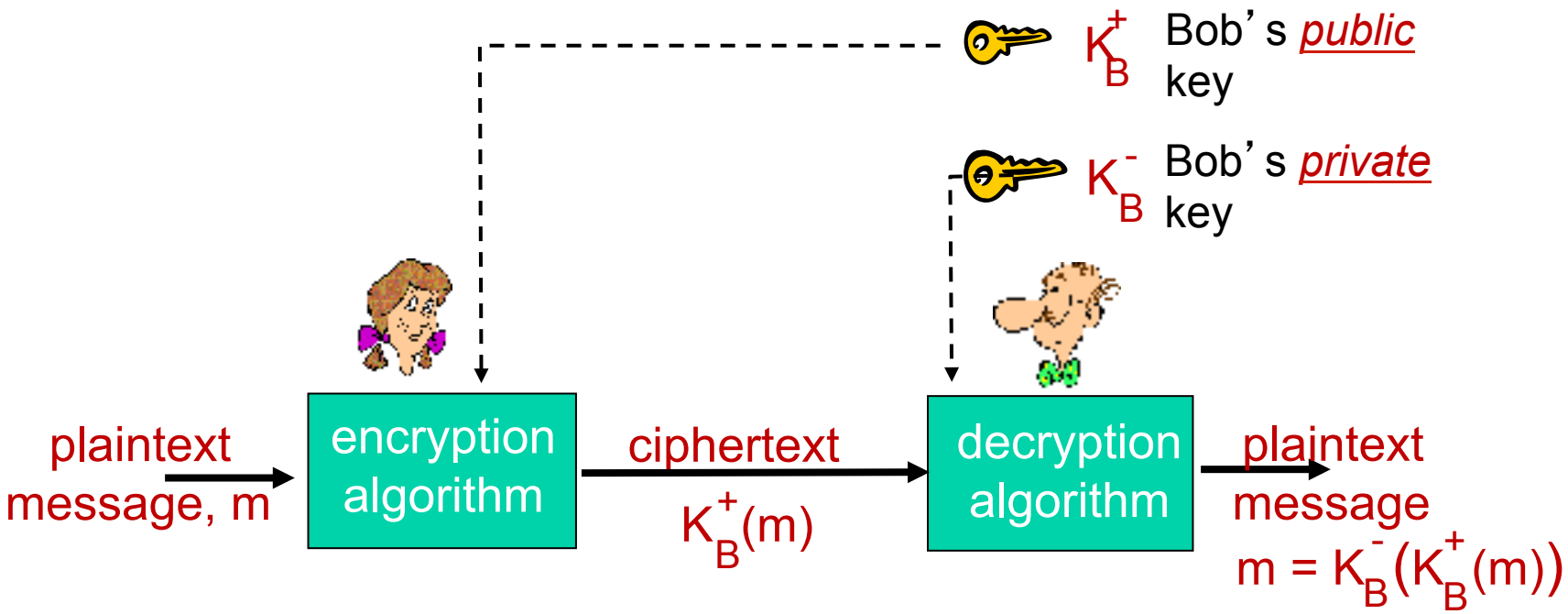
*Crittografia a chiave simmetrica*

- 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*

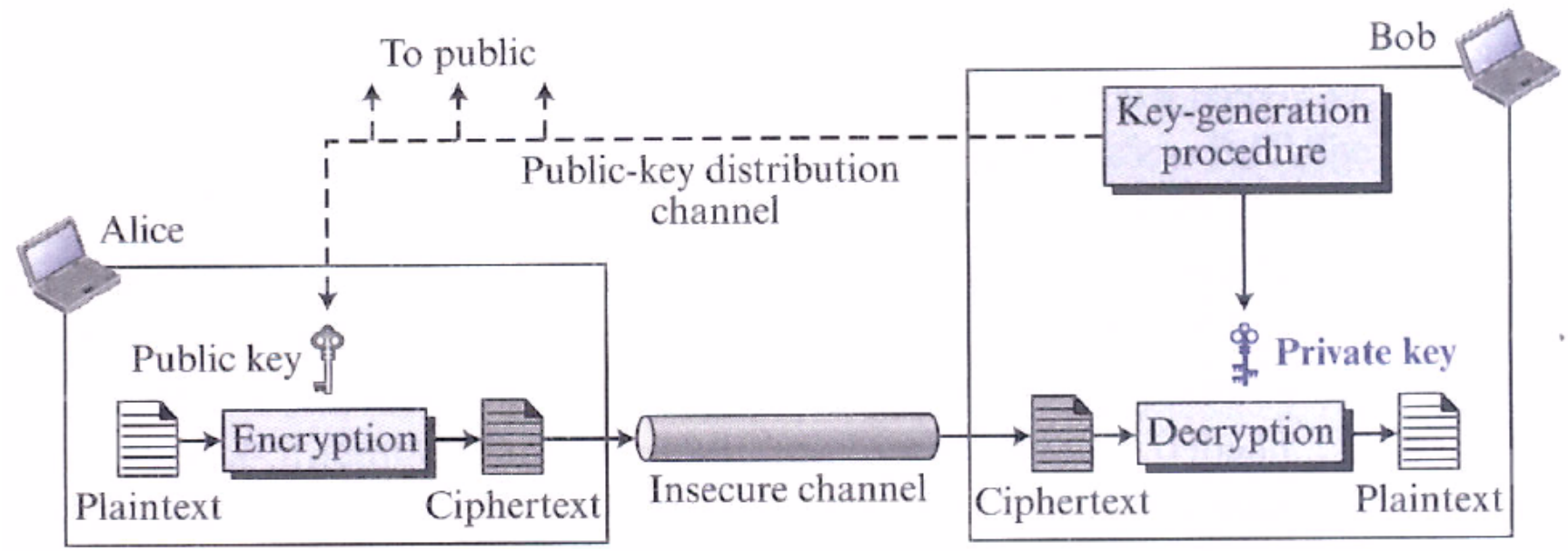
- approccio radicalmente diverso [Diffie-Hellman76, RSA78]
- mittente e destinatario *non* condividono una chiave segreta
- la chiave di cifratura *pubblica* è nota *a tutti*
- la chiave di cifratura *privata* è nota solo al destinatario

# Crittografia a chiave pubblica




Sono intrinsecamente sicuri poiché utilizzano tecniche di tipo matematico basate sulla teoria dei numeri, sulla teoria delle curve ellittiche, etc.

# Crittografia a chiave pubblica




- 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

# Prerequisite: modular arithmetic

- x mod n = remainder of x when divide by n
- facts:

  $[(a \bmod n) + (b \bmod n)] \bmod n = (a+b) \bmod n$

  $[(a \bmod n) - (b \bmod n)] \bmod n = (a-b) \bmod n$

  $[(a \bmod n) * (b \bmod n)] \bmod n = (a*b) \bmod n$
- thus

  $(a \bmod n)^d \bmod n = a^d \bmod n$
- example: x=14, n=10, d=2:

  $(x \bmod n)^d \bmod n = 14^2 \bmod 10 = 6$

  $x^d = 14^2 = 196 \quad x^d \bmod 10 = 6$

# RSA: getting ready

- message: just a bit pattern
- bit pattern can be uniquely represented by an integer number
- thus, encrypting a message is equivalent to encrypting a number.

*example:*

- m= 10010001 . This message is uniquely represented by the decimal number 145.
- to encrypt m, we encrypt the corresponding number, which gives a new number (the ciphertext).

# 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: $ed \bmod z = 1$ ).

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

# RSA: encryption, decryption

0. given $(n,e)$ and $(n,d)$ as computed above

1. to encrypt message $m$ $(<n)$, compute

   $c = m^e \bmod n$

2. to decrypt received bit pattern, $c$, compute

   $m = c^d \bmod n$

*magic happens!*

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

# RSA example:

Bob chooses $p=5$, $q=7$. Then $n=35$, $z=24$.

$e=5$ (so $e$, $z$ relatively prime).

$d=29$ (so $ed-1$ exactly divisible by z).

encrypting 8-bit messages.

| | bit pattern | $m$ | $m^e$ | $c = m^e \bmod n$ |
|---|---|---|---|---|
| encrypt: | 0000l000 | 12 | 24832 | 17 |

| | $c$ | $c^d$ | $m = c^d \bmod n$ |
|---|---|---|---|
| decrypt: | 17 | 48196857210675091509141182522307169 7 | 12 |

# Why does RSA work?

- must show that $c^d \bmod n = m$
  where $c = m^e \bmod n$
- fact: for any x and y: $x^y \bmod n = x^{(y \bmod z)} \bmod n$
  - where $n = pq$ and $z = (p-1)(q-1)$
- thus,

$$
\begin{aligned}
c^d \bmod n &= (m^e \bmod n)^d \bmod n \\
&= m^{ed} \bmod n \\
&= m^{(ed \bmod z)} \bmod n \\
&= m^1 \bmod n \\
&= m
\end{aligned}
$$

# RSA: un'importante proprietà

La seguente proprietà sarà *molto* utile più avanti:

$$\underbrace{K_B^-(K_B^+(m))}_{} = m = \underbrace{K_B^+(K_B^-(m))}_{}$$

Si usa prima la chiave pubblica, e poi quella privata

Si usa prima la chiave privata, e poi quella pubblica

*Il risultato non cambia!*

# Why $K_B^-(K_B^+(m)) = m = K_B^+(K_B^-(m))$?

follows directly from modular arithmetic:

$$
\begin{aligned}
(m^e \bmod n)^d \bmod n &= m^{ed} \bmod n \\
&= m^{de} \bmod n \\
&= (m^d \bmod n)^e \bmod n
\end{aligned}
$$

# Why is RSA secure?

- suppose you know Bob's public key (n,e). How hard is it to determine d?
- essentially need to find factors of n without knowing the two factors p and q
  - fact: factoring a big number is hard

# RSA in practice: session keys

- exponentiation in RSA is computationally intensive
- DES is at least 100 times faster than RSA
- use public key crypto to establish secure connection, then establish second key – symmetric session key – for encrypting data

*session key, $K_S$*

- Bob and Alice use RSA to exchange a symmetric key $K_S$
- once both have $K_S$, they use symmetric key cryptography

Il 6 Settembre 2000 l'algoritmo RSA è diventato di dominio pubblico, prima era di proprietà dell'omonima azienda RSA Security Inc.

# Tecniche di crittoanalisi

- **Brute-force**, ossia tramite il calcolo di tutte le possibili combinazioni di chiavi del cifrario. Con l'aumento della potenza di calcolo degli elaboratori questa tecnica banale stà diventando sempre più efficace. Basti pensare al Cracking del DES a 56 bit con un computer multiprocessore costruito dall'EFF in grado di violare l'algoritmo in meno di 3 giorni di calcolo.
- **Crittoanalisi differenziale**, tramite l'analisi delle "distanze" numeriche dei caratteri presenti nel testo cifrato e l'ausilio di sofisticate tecniche matematiche unite ad algoritmi sempre più veloci.
- **Man-in-the-middle**, sfruttando il sistema delle infrastrutture PKI un eventuale intruso può posizionarsi tra un mittente ed un destinatario e scambiare le loro chiavi pubbliche e private con altre opportunamente modificate.

# Chapter 8 roadmap

8.1 What is network security?

8.2 Principles of cryptography

*8.3 Message integrity,* authentication

8.4 Securing e-mail

8.5 Securing TCP connections: SSL

8.6 Network layer security: IPsec

8.7 Securing wireless LANs

8.8 Operational security: firewalls and IDS

# Integrità del messaggio

Bob 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

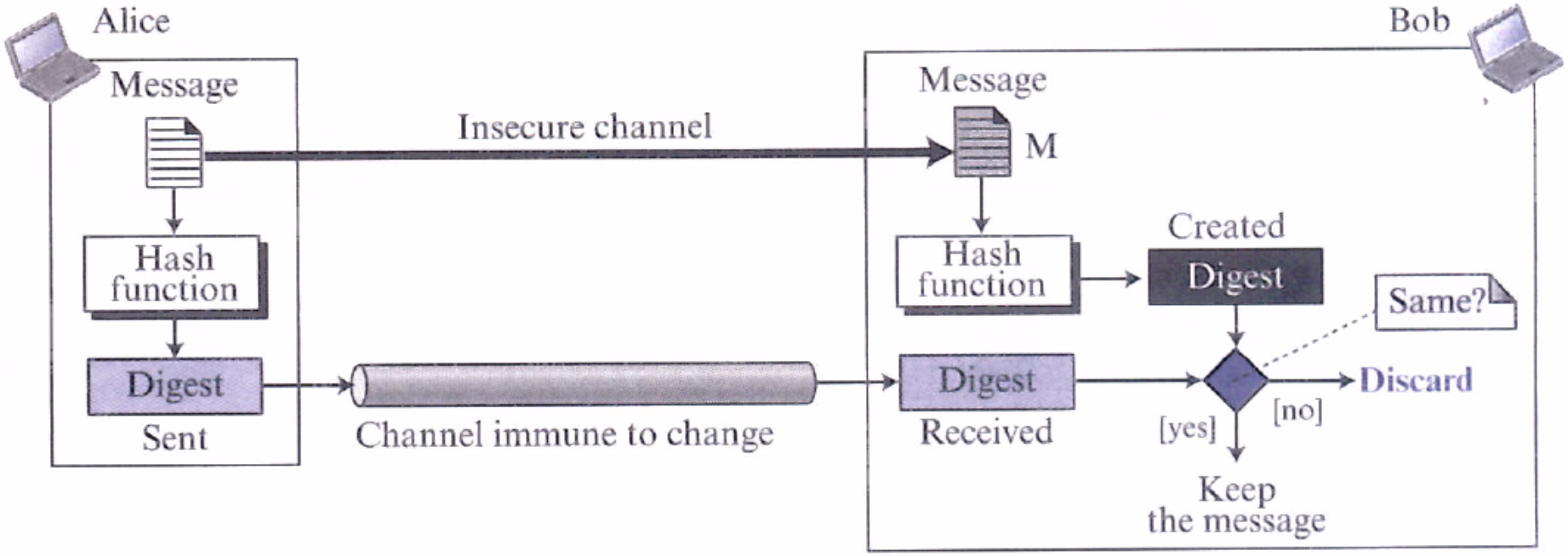
- prende in input m, produce un valore a lunghezza fissa, H(m)
- Deve essere computazionalmente impossibile trovare due messaggi x e y tali che H(x) = H(y)
  - o anche: dato m = H(x), (con x sconosciuta), è impossibile determinare x.


8-49

# 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!!!



# Esempio di funzione Hash

- Tutte le funzioni hash operano sulla base del seguente principio: i dati in ingresso sono considerati come una sequenza di blocchi di n bit, essi vengono elaborati un blocco alla volta iterativamente per produrre una funzione hash di n bit.
- Una delle più semplici funzioni hash è quella che esegue lo XOR (+) bit a bit di ciascun blocco, ossia:

$$C_i = b_{i1} + b_{i2} + \ldots + b_{im}$$

- Dove $C_i$ rappresenta l'i-esimo bit del codice hash, m il numero di blocchi di n bit, $b_{ij}$ l'i-esimo bit all'interno del j-esimo blocco e l'operatore + l'operazione di XOR.
- La probabilità che un errore nei dati produca lo stesso valore hash è $2^{-n}$, con n=128 bit $2^{-128} \approx 2{,}9387*10^{-39}$.

# Funzione Hash

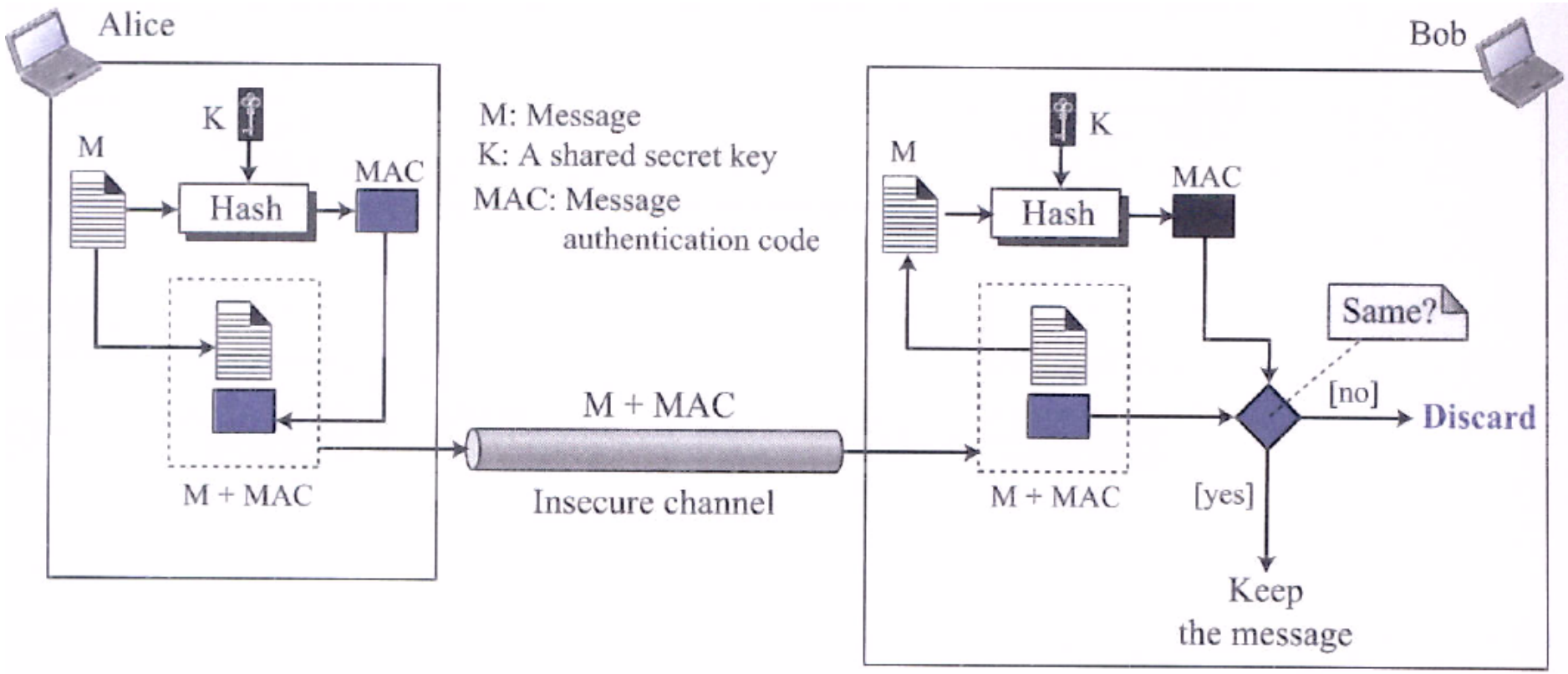
- MD5 è molto usato per l'hash dei messaggi (RFC 1321)
  - MD sta per Message Digest
  - Calcola una hash di 128 bit con un processo a 4 fasi
  - Con una stringa x di 128 bit arbitrari, appare difficile costruire un messaggio m il cui hash MD5 sia uguale a x

- È molto usato anche Secure Hash Algorithm (SHA-1)
  - Standard statunitense [NIST, FIPS PUB 180-1]
  - Produce una sintesi del messaggio di 160 bit


8-52

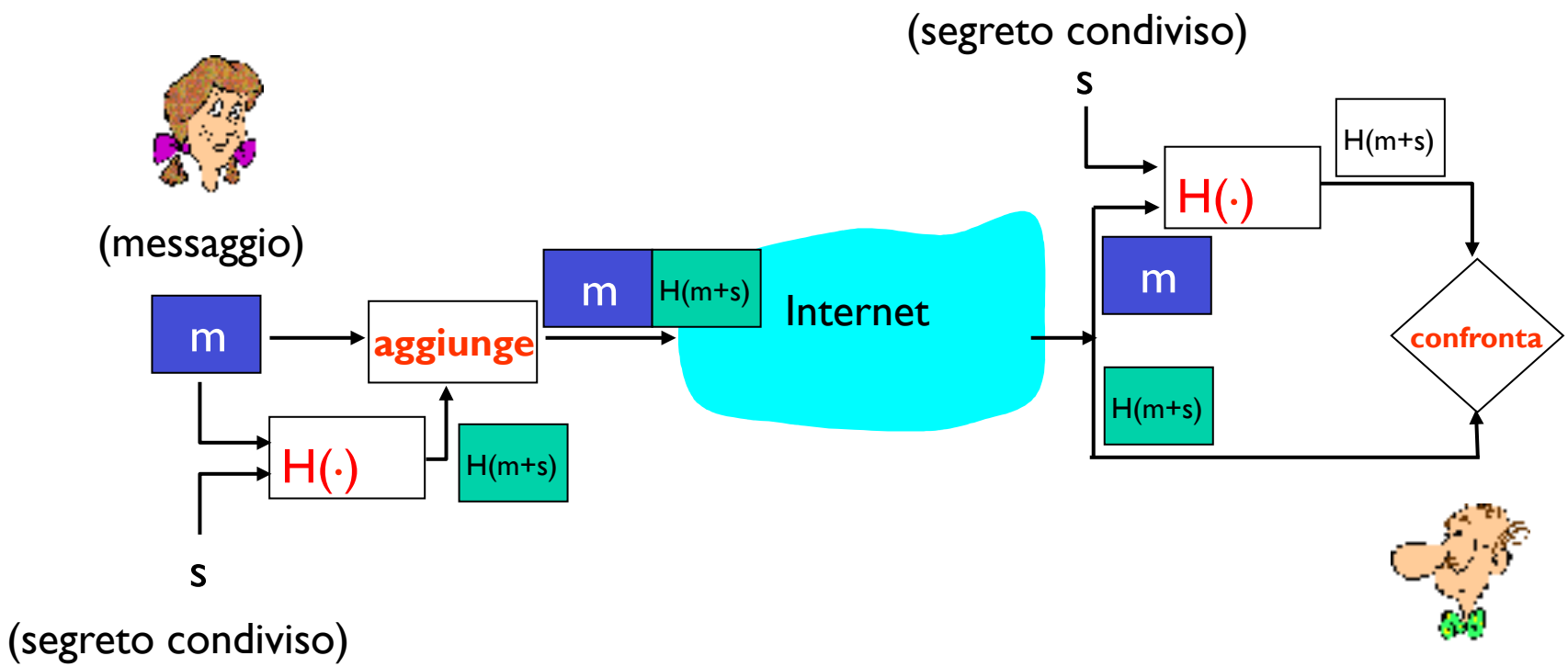
# Autenticazione dei messaggi

- 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





8-54

# 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 (Bob) può dimostrare che Alice e nessun altro (Bob incluso) può aver firmato il documento.
- Avviene mediante l'uso di una coppia di chiave pubblica e privata di chi firma

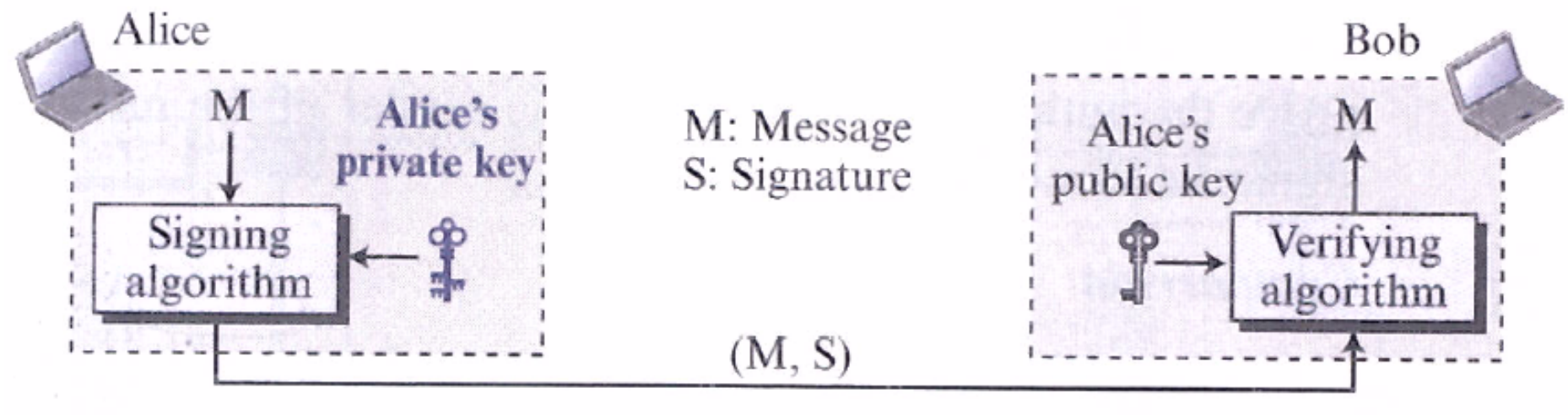

8-55

# Firma digitale

Creazione della firma digitale di un messaggio M:

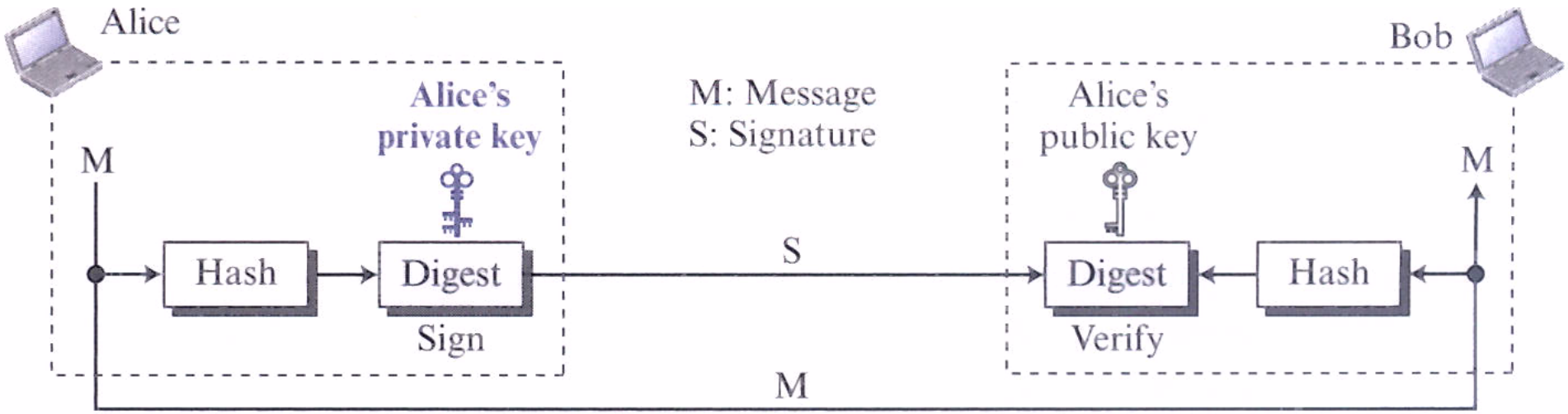
- Alice firma un messaggio M, cifrandolo con la sua chiave privata, creando così un messaggio "firmato" -
- Bob decifra il messaggio applicando la chiave pubblica di Alice

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



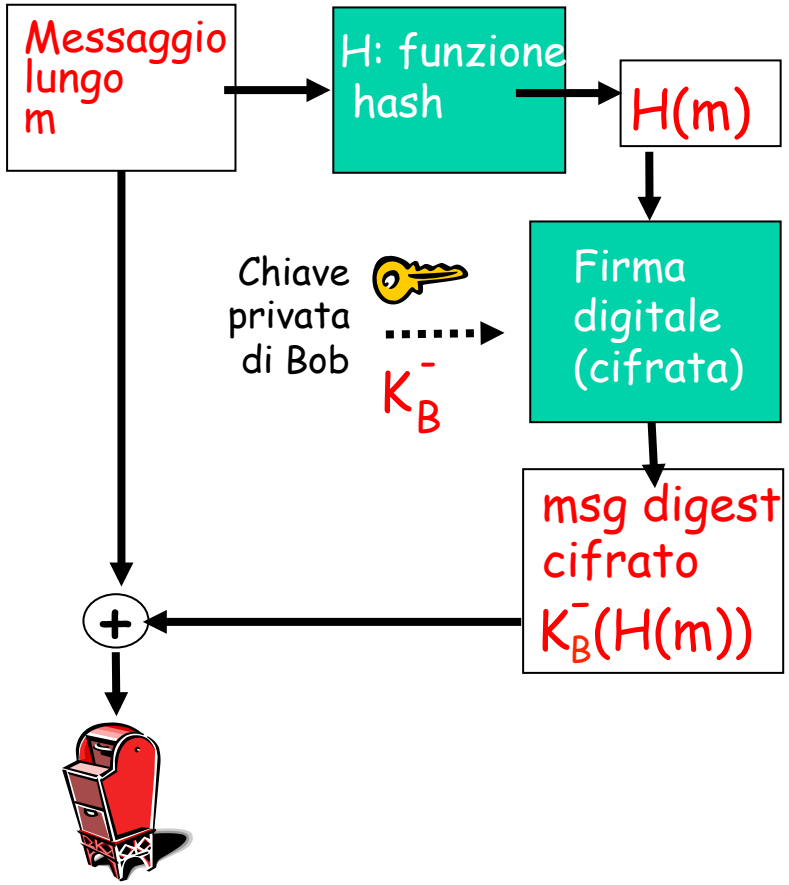
# Firma digitale (messaggi digest firmati)

- 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 Bob 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

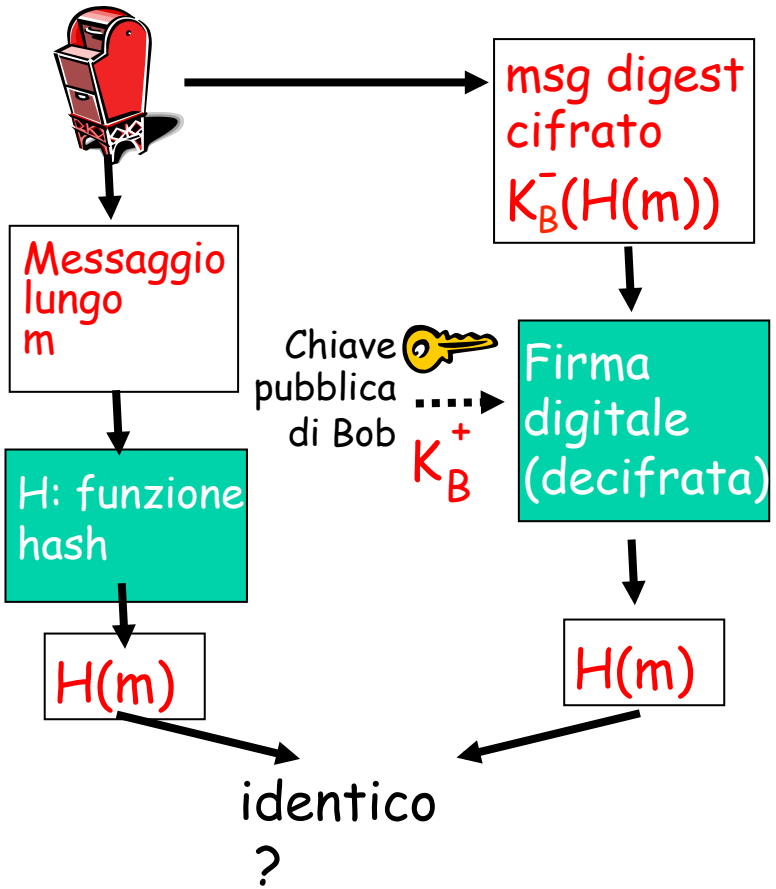


# Firma digitale = messaggi digest firmati

Bobinvia un messaggio con la firma digitale:




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



# Chapter 8 roadmap

8.1 What is network security?

8.2 Principles of cryptography

*8.3* Message integrity*, authentication*

8.4 Securing e-mail

8.5 Securing TCP connections: SSL

8.6 Network layer security: IPsec

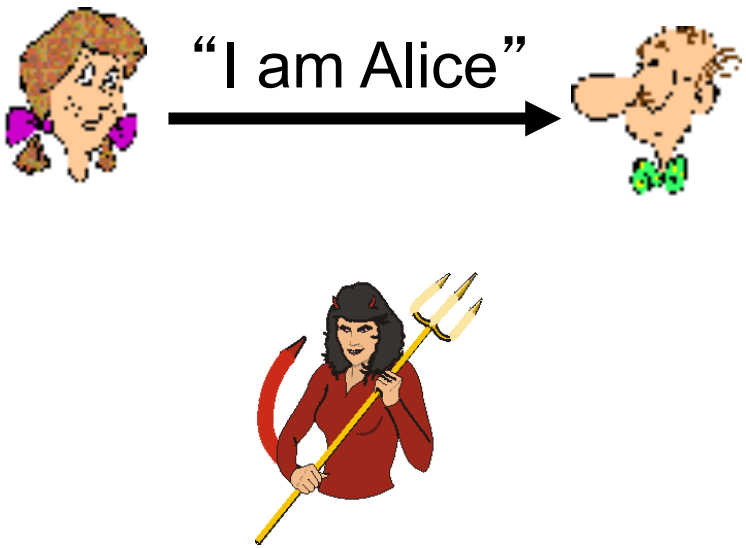
8.7 Securing wireless LANs

8.8 Operational security: firewalls and IDS

# Authentication

*Goal:* Bob wants Alice to "prove" her identity to him

*Protocol ap1.0:* Alice says "I am Alice"

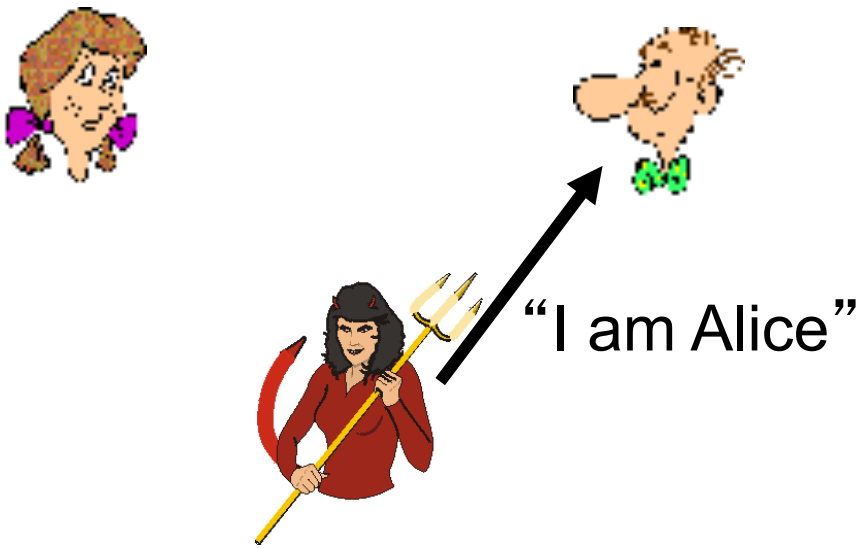



Failure scenario??

# Authentication

*Goal:* Bob wants Alice to "prove" her identity to him

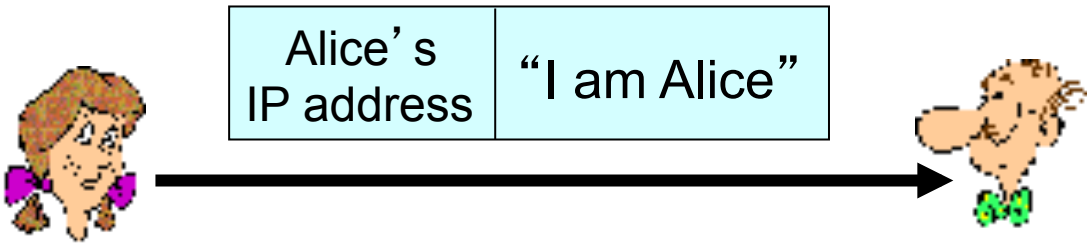
*Protocol ap1.0:* Alice says "I am Alice"




in a network,
Bob can not "see" Alice,
so Trudy simply declares
herself to be Alice

# Authentication: another try

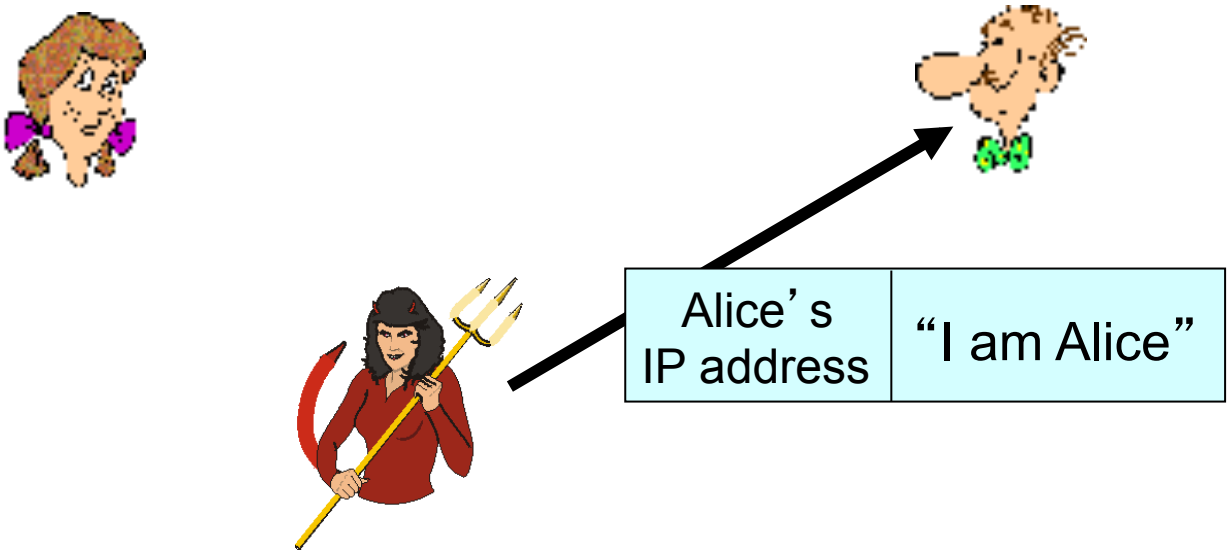
*Protocol ap2.0:* Alice says "I am Alice" in an IP packet containing her source IP address




Failure scenario??

# Authentication: another try

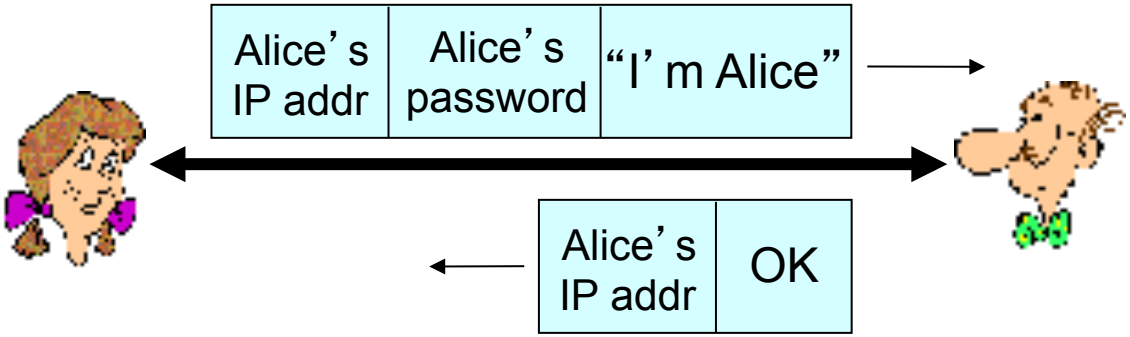
*Protocol ap2.0:* Alice says "I am Alice" in an IP packet containing her source IP address




Trudy can create a packet "spoofing" Alice's address

# Authentication: another try

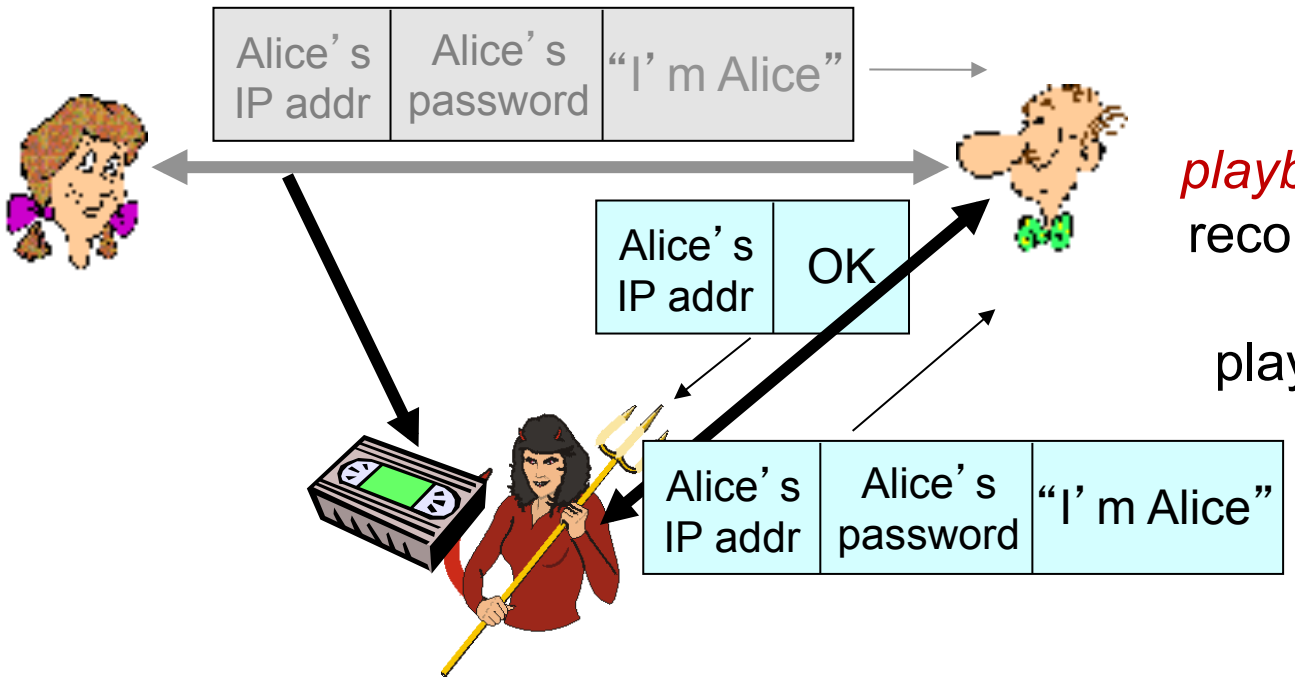
*Protocol ap3.0:* Alice says "I am Alice" and sends her secret password to "prove" it.



Failure scenario??

# Authentication: another try

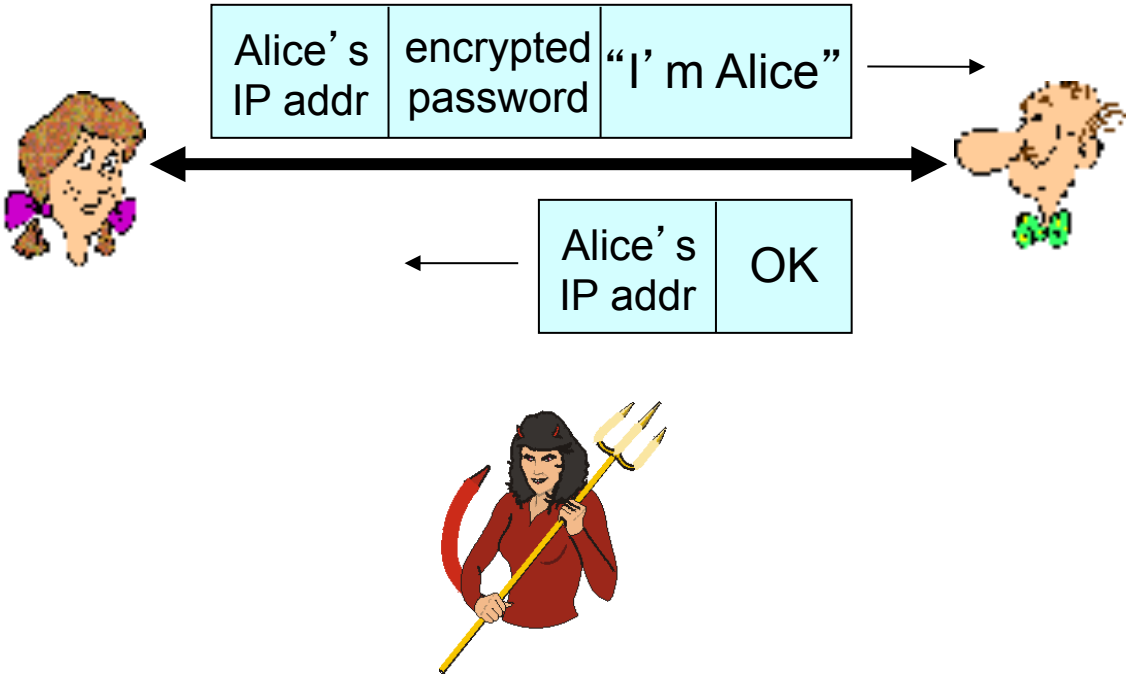
*Protocol ap3.0:* Alice says "I am Alice" and sends her secret password to "prove" it.




*playback attack:* Trudy records Alice's packet and later plays it back to Bob

# Authentication: yet another try

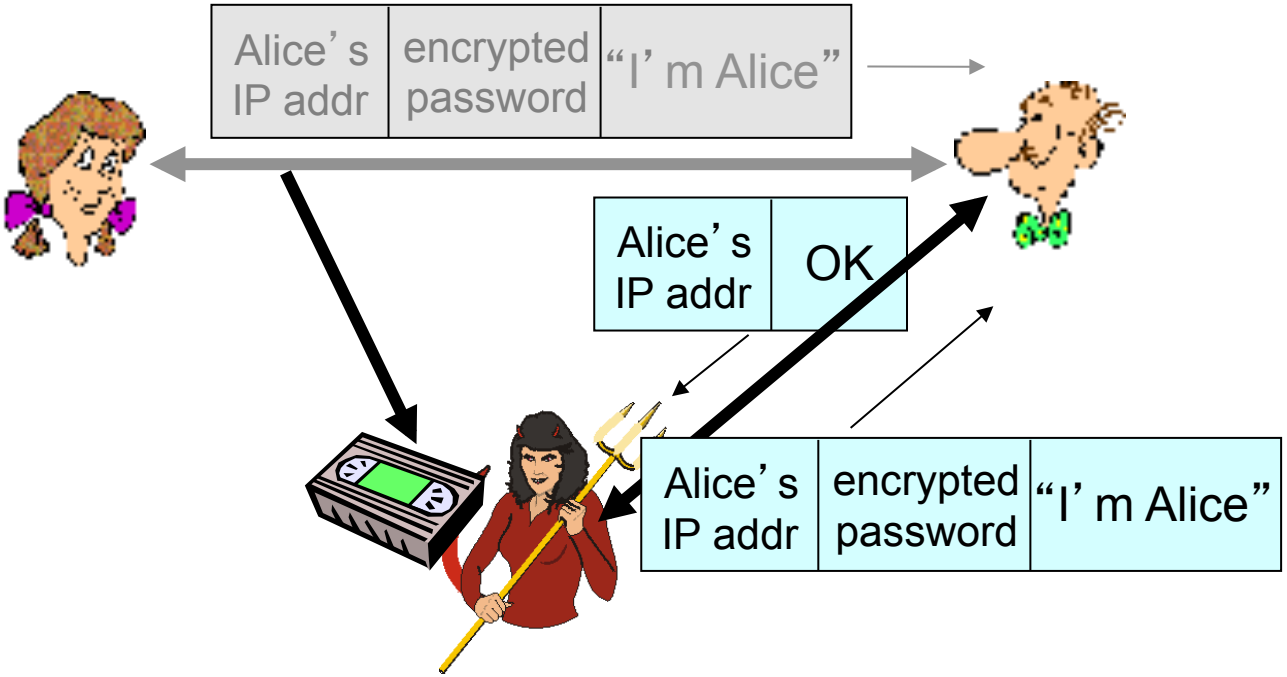
*Protocol ap3.1:* Alice says "I am Alice" and sends her *encrypted* secret password to "prove" it.




Failure scenario??

# Authentication: yet another try

*Protocol ap3.1:* Alice says "I am Alice" and sends her *encrypted* secret password to "prove" it.



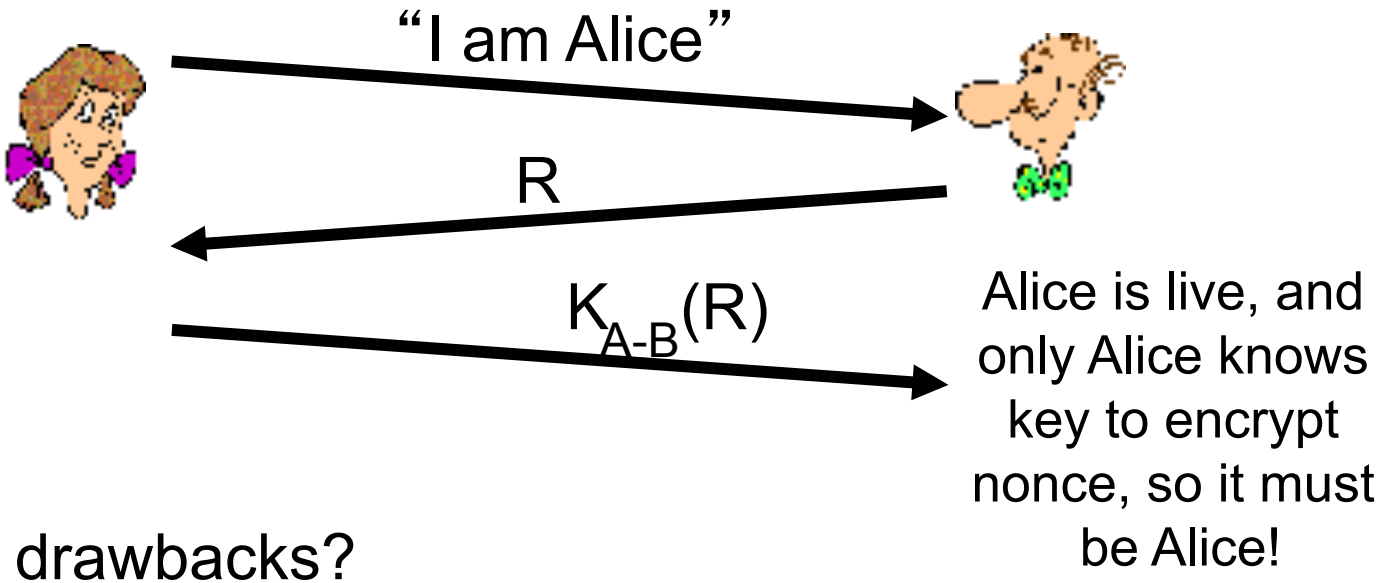

record and playback *still* works!

# Authentication: yet another try

*Goal:* avoid playback attack

*nonce:* number (R) used only *once-in-a-lifetime*

*ap4.0:* to prove Alice "live", Bob sends Alice *nonce*, R. Alice must return R, encrypted with shared secret key



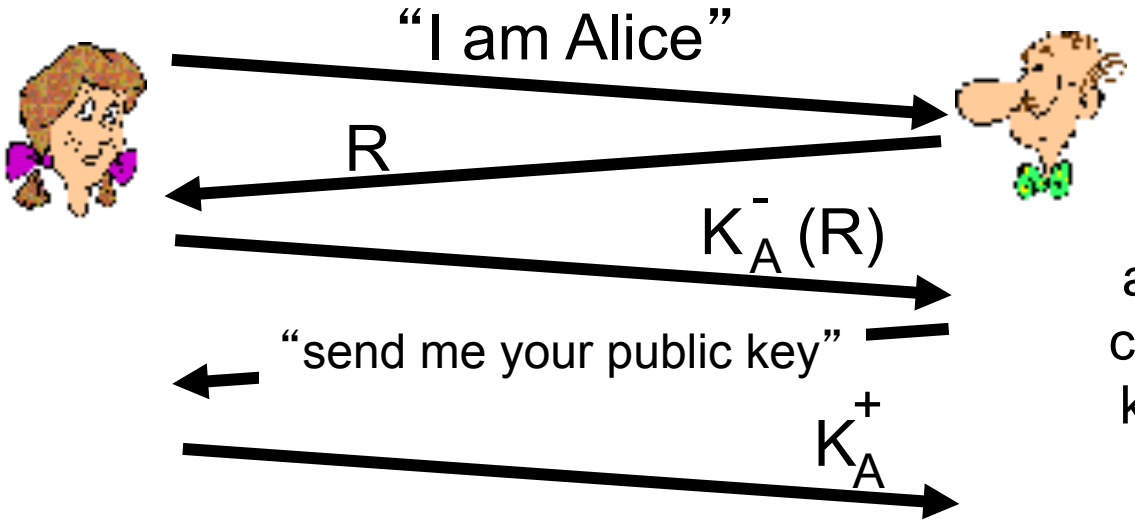

Failures, drawbacks?

# Authentication: ap5.0

ap4.0 requires shared symmetric key

- can we authenticate using public key techniques?

*ap5.0:* use nonce, public key cryptography




Bob computes

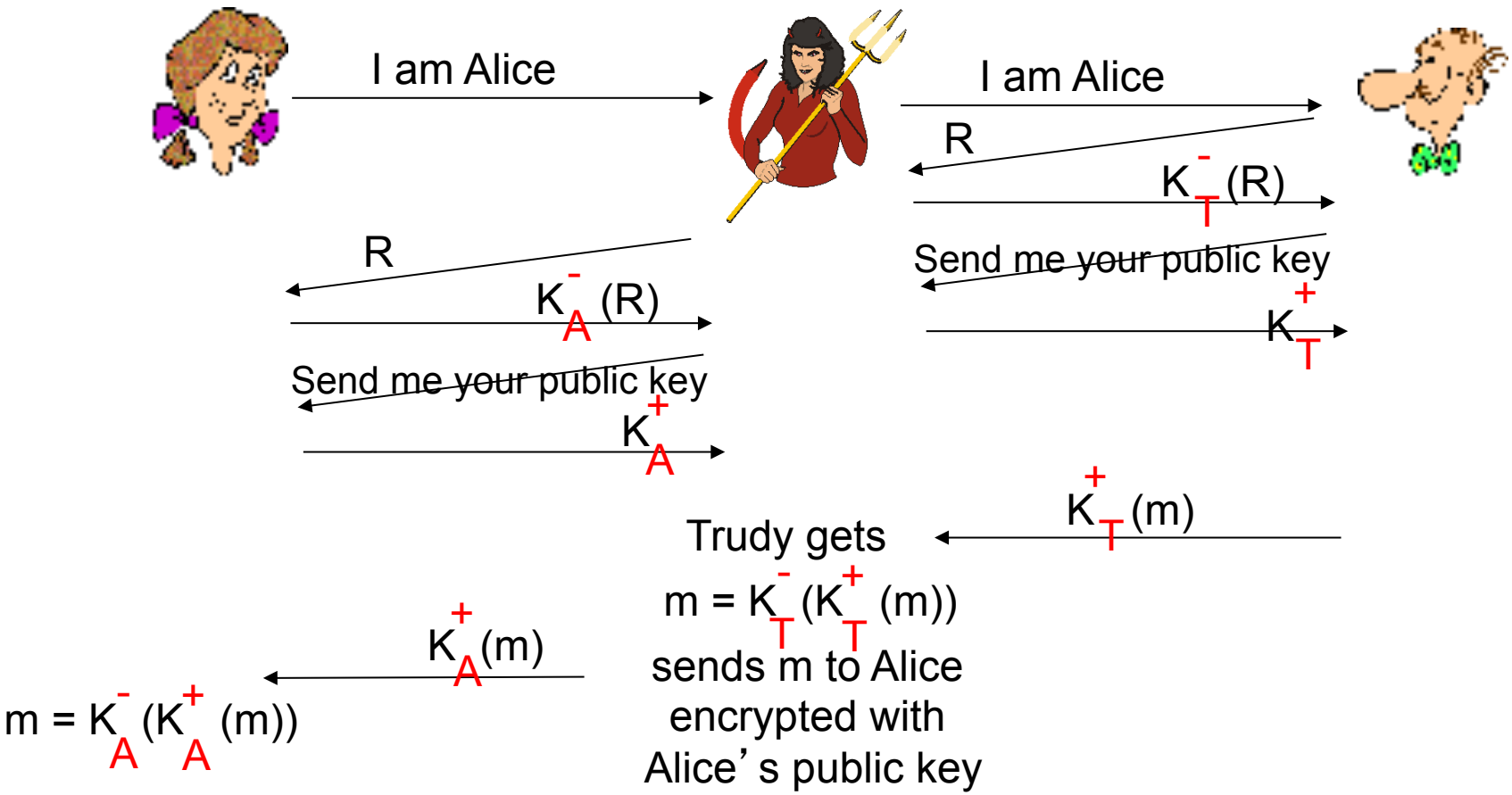
$K_A^+(K_A^-(R)) = R$

and knows only Alice could have the private key, that encrypted R such that

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

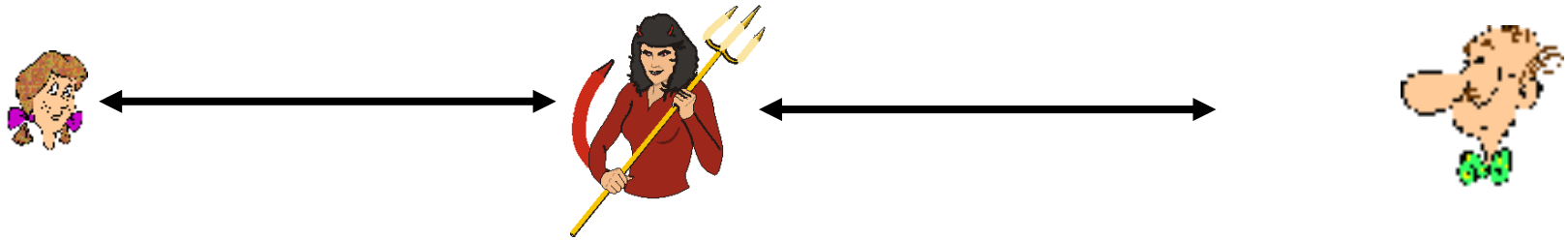
# ap5.0: security hole

*man (or woman) in the middle attack:* Trudy poses as Alice (to Bob) and as Bob (to Alice)



# ap5.0: security hole

*man (or woman) in the middle attack:* Trudy poses as Alice (to Bob) and as Bob (to Alice)



difficult to detect:

- Bob receives everything that Alice sends, and vice versa. (e.g., so Bob, Alice can meet one week later and recall conversation!)
- problem is that Trudy receives all messages as well!

# Certificazione della chiave pubblica

Problema per la crittografia a chiave pubblica:

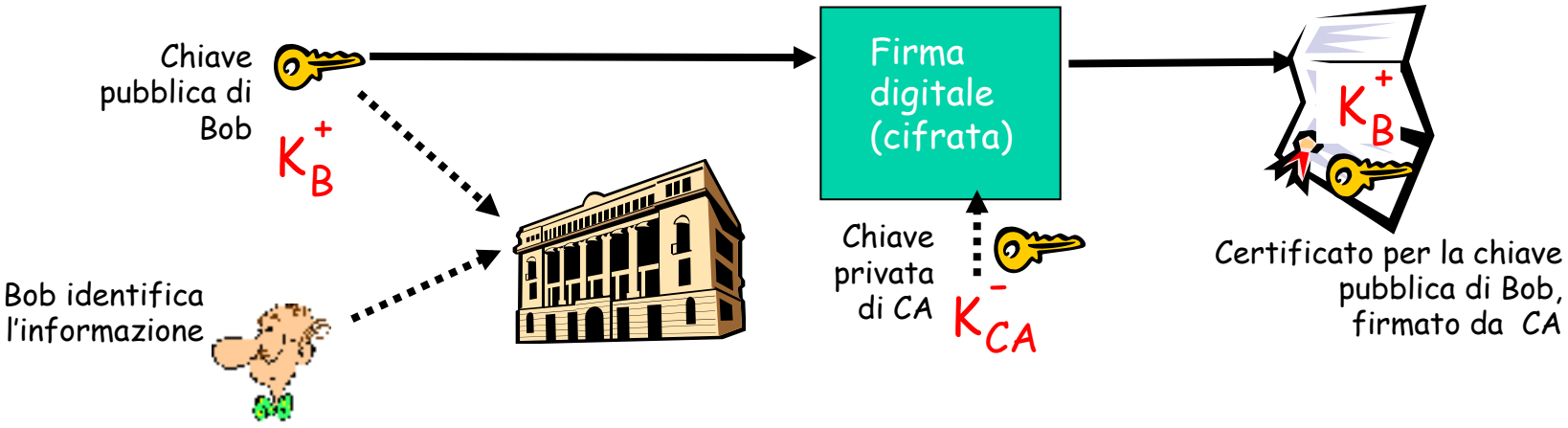
- Quando Alice riceve la chiave pubblica di Bob(attraverso un dischetto, il sito web o via e-mail), come fa a sapere che è veramente la chiave pubblica di Bobe non, magari, quella di Trudy?

Soluzione:

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

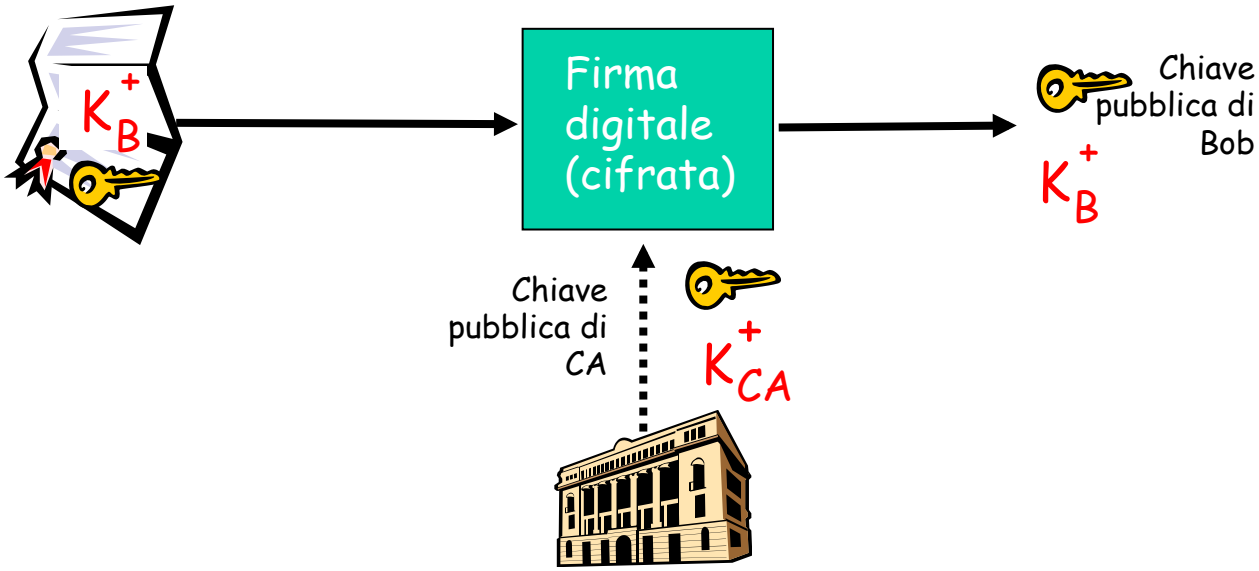
# Autorità di certificazione

- 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")



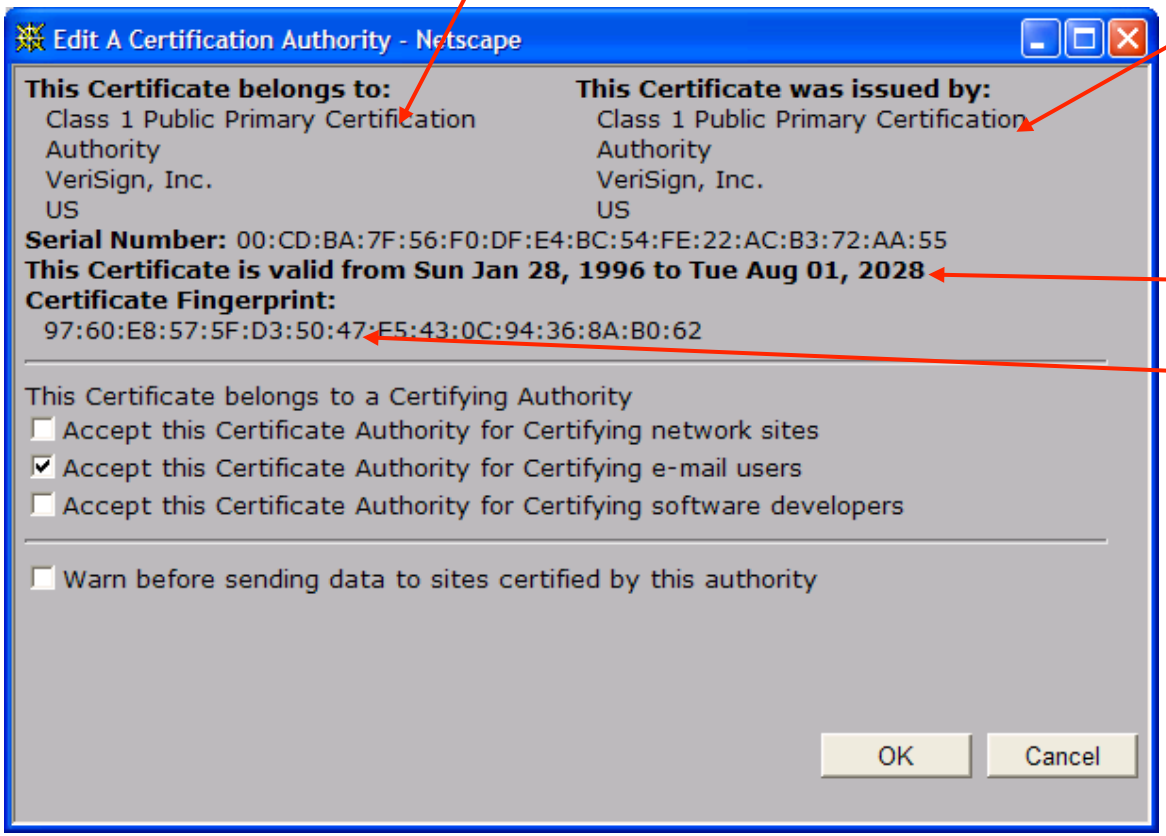
# Autorità di certificazione

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



# 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

# DOMANDE?