

# Sicurezza in Internet

---

## Crittografia

# Sommario

---

- Introduzione
- Cifratura a Chiave Simmetrica
- Cifratura a Chiave Pubblica
- Autenticazione
- Firma Elettronica
- Distribuzione delle Chiavi
- Protocolli

# Sicurezza nelle reti

---

- Ci sono tre aree in cui bisogna intervenire per rendere una rete sicura
    - **Riservatezza (Confidentiality)**: il messaggio deve essere accessibile (visualizzabile o rilevabile la sua presenza) solo ad entità autorizzate.
    - **Autenticazione**: L'identità delle entità coinvolte nella comunicazioni deve poter essere verificata.
    - **Integrità (Integrity)** (ed eventuale “firma”): impedire che i dati possano essere modificati se non da autorità autorizzate (con firma: anche legate all'autore, che non ne possa disconoscere la paternità).
-

# Sicurezza nelle reti: Attacchi

---

## Passivi

- **Accesso al contenuto**: venire a conoscenza di informazioni riservate.  
Ad esempio lo *Sniff* (il fiutare) di pacchetti su LAN a mezzo condiviso.
- **Analisi del traffico**: senza vedere i contenuti specifici, riconoscere l'entità dei comunicanti e tipo e frequenza dei messaggi.
- Sono difficili da rilevare, quindi si devono prevenire.

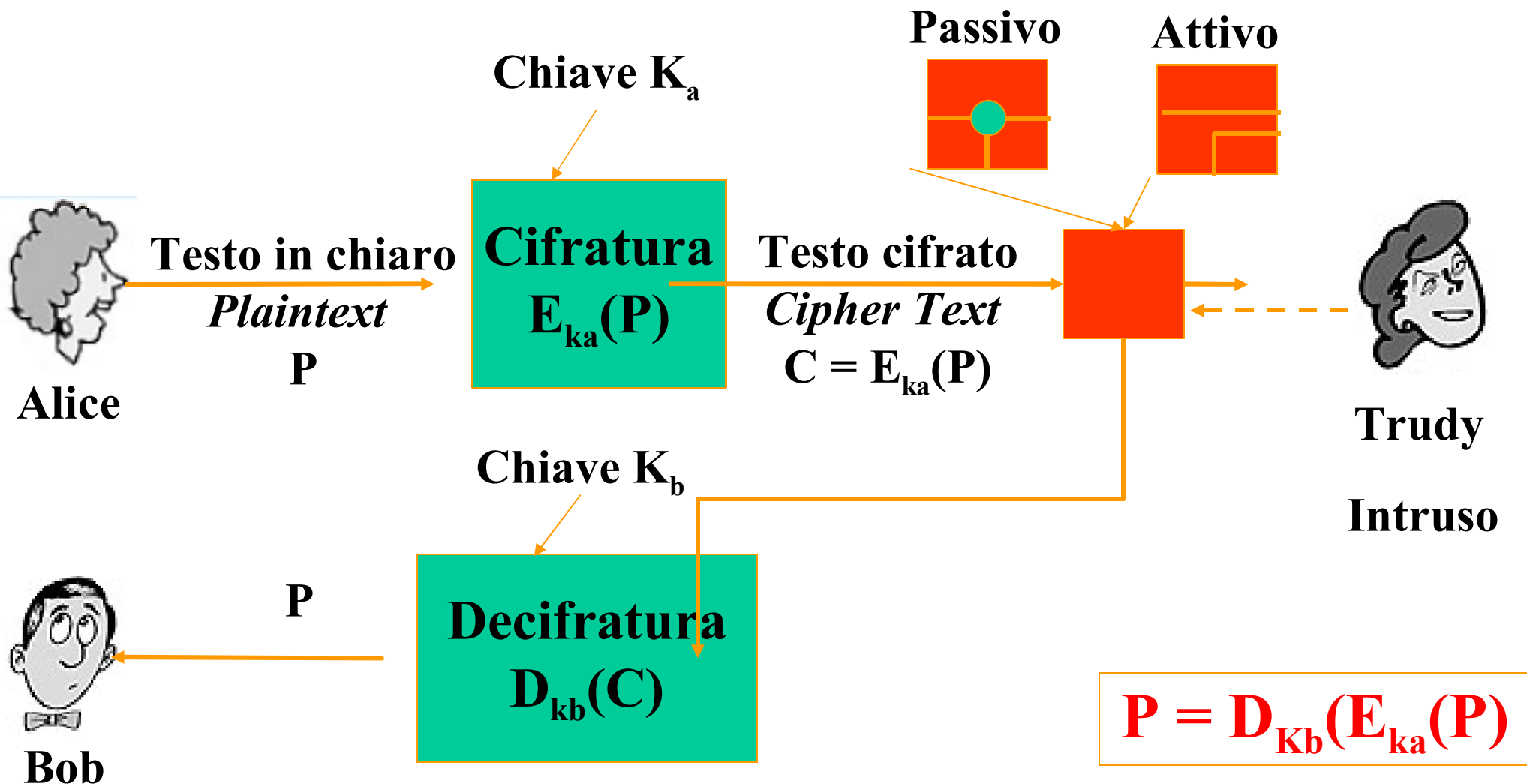
# Sicurezza nelle reti: Attacchi

---

## Attivi

- **Sostituzione**: Farsi passare per un altro  
Ad esempio lo *Spoofing* (imbroglio) IP.
- **Replica**: copiare e riproporre un messaggio per ottenere effetti non autorizzati (ad esempio, un doppio versamento).
- **Alterazione**: modifica anche solo dell'ordine
- **Negazione del servizio**: inibire l'uso o la gestione di un sistema (anche dell'intera rete), ad esempio per impedire la generazione o arrivo di messaggi di allarme (*SYN Attack*).
- Possono sia essere rilevati e quindi fermati che prevenuti

# Riservatezza: Cifratura



# Cifratura a chiave Simmetrica

---

- E' una tecnica antica (Giulio Cesare)
- $K_A = K_B = K$ : una sola chiave
- Deve rispettare due requisiti per essere sicura:
  - Robustezza dell'algoritmo: anche conoscendo l'algoritmo ed avendo campioni di testo in chiaro e cifrato, l'intruso non deve essere in grado di decifrare il testo e scoprire la chiave
  - Mittente e destinatario devono poter ottenere in modo sicuro la chiave e custodirla efficacemente.

# Attacchi al testo cifrato

---

- **Attacco al testo cifrato (cipertext only)**: chi attacca ha a disposizione solo la conoscenza di una certa quantità di testo cifrato.
- **Attacco al testo in chiaro conosciuto (known plaintext)**: chi attacca conosce alcuni campioni di testo in chiaro e i corrispondenti messaggi cifrati.
- **Attacco al testo in chiaro scelto (chosen plaintext)**: chi attacca ha la possibilità di criptare il testo in chiaro desiderato.



# Cifratura a chiave simmetrica

---

- Per scardinare un algoritmo di cifratura esistono due tecniche:
  - **Criptoanalisi**: che si basa sulla natura degli algoritmi, su campioni, su caratteristiche statistiche di P.
  - **Forza bruta**.

<b>Dim. chiave</b>	<b># di chiavi possibili</b>	<b>Tempo (1 cifr./<math>\mu</math>s)</b>	<b>Tempo (<math>10^6</math> cifr./<math>\mu</math>s)</b>
32	$2^{32} = 4,3 \times 10^9$	231 s = 35,8 min.	2,15 ms
56	$2^{56} = 7,2 \times 10^{16}$	255 s = 1142 anni	10,01 ore
128	$2^{128} = 3,4 \times 10^{38}$	2127 s = $5,4 \cdot 10^{24}$ anni	$5,4 \cdot 10^{18}$ anni
168	$2^{168} = 3,7 \times 10^{50}$	2167 s = $5,9 \cdot 10^{36}$ anni	$5,9 \cdot 10^{30}$ anni

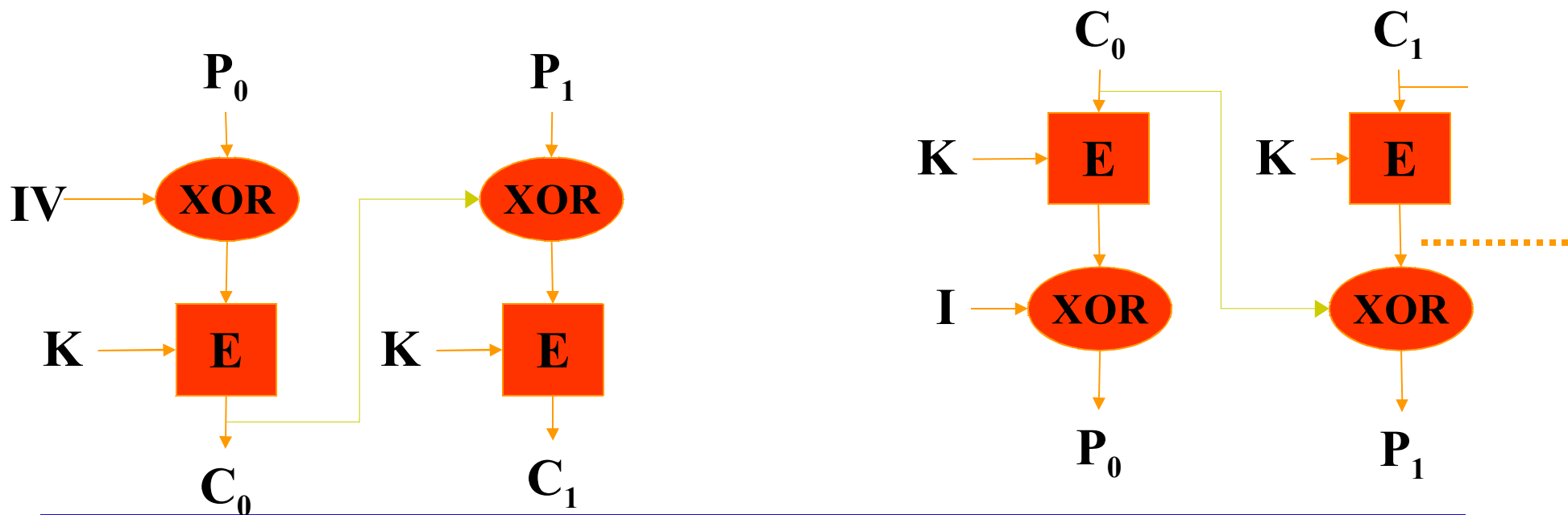
# Cifratura a chiave simmetrica

---

- Viene in genere realizzata con una sostituzione monoalfabetica:
  - Sostituisco una “lettera” (blocco di dati) con un’altra.
- Se le lettere sono quelle dell’alfabeto (blocchi di 7 o 8 bit) ho
  - 26! possibili accoppiamenti pari a circa  $10^{26}$
  - Facile usare meccanismi statistici per scardinare il codice

# Cifratura a chiave simmetrica

- Per rendere la tecnica più efficace
  - si usano “lettere” più grandi (ad es.  $n = 64$  bit) e slegate dal testo, ossia si sostituisce un blocco di bit di lunghezza fissa con un altro.
  - Si concatena il risultato di una cifratura con la successiva, ossia si esegue il concatenamento di blocchi cifrati (**Cipher Block Chaining, CBC**)



# Cifratura a chiave simmetrica

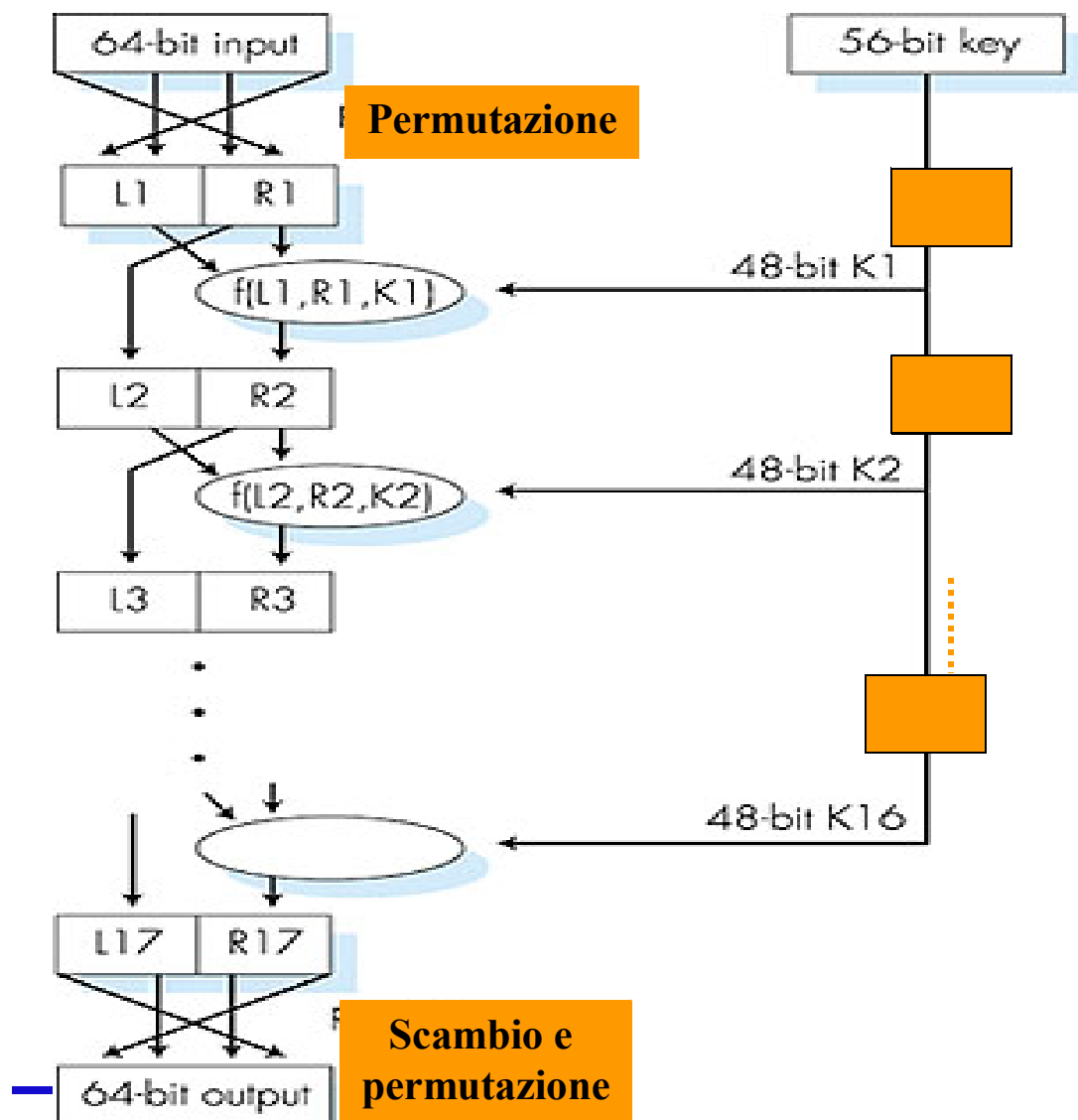
## *Data Encryption Standard (DES)*

---

- Nasce nel 1977 e viene aggiornato nel 1993,
- E' stato adottato dal U. S. *National Bureau of Standard* (oggi *National Institute for Standard and Technology*, NIST)
- L'algoritmo vero e proprio si chiama *Data Encryption Algorithm (DEA)*:
  - Opera su blocchi da 64 bit.
  - Usa una chiave da 56 bit.
  - Si compone di 19 stadi:
    - » Una prima permutazione
    - » 16 stadi parametrizzati da una variante della chiave  $K_i$ ,  $i=1,\dots,16$
    - » Uno scambio dei 32 bit destri con i sinistri
    - » Una permutazione inversa alla prima

# Cifratura a chiave simmetrica

## *Data Encryption Standard (DES)*



- In genere viene usato in unione con un concatenamento (CBC).
- La decifratura avviene con lo stesso meccanismo ma usando le chiavi in ordine inverso
- La complessità dell'algoritmo risiede nella funzione  $f(\cdot)$ .

# Cifratura a chiave simmetrica

## *Data Encryption Standard (DES)*

---

- Per quanto concerne la robustezza, sono stati indetti tre concorsi (*challenger*) per violarlo:
  - *Challenger I* (1997): scardinato in 4 mesi;
  - *Challenger II* (1998): scardinato in 56 ore
  - *Challenger II* (1999): scardinato in 22 ore e 15 min. (testate  $245 \times 10^9$  chiavi al sec.)
- Ad oggi, (nella sua forma con chiave a 56 bit) non è considerato molto sicuro.

# Cifratura a chiave simmetrica

## *Triplo-DEA (T-DEA)*

---

- Standardizzato dall'ANSI (1985) come X 9.17 e parte del DES dal 1999
- Usa 3 chiavi da 56 bit:  $K_1$ ,  $K_2$ ,  $K_3$ .
- Opera come segue:

$$C = E_{K_3}(D_{K_2}(E_{K_1}(P)))$$

- Questo significa che ha una chiave di lunghezza complessiva pari a 168 bit
- Si può cifrare e decifrare il DEA ponendo tutte le chiavi uguali
- Si può usare una chiave da 112 bit ponendo  $K_1 = K_3$

# Cifratura a chiave simmetrica: AES

---

- Il National Institute of Standards and Technology (NITS), ente governativo americano, nel 1997 ha lanciato un concorso pubblico per individuare un nuovo standard di crittografia, per uso generico del governo americano
- il nome dello standard sarebbe stato AES (Advanced Encryption Standard);
- lo scopo del concorso pubblico era quello di evitare ogni possibile sospetto sul nuovo standard.



# Cifratura a chiave simmetrica: AES

---

- I requisiti richiesti erano:
  - utilizzo di crittografia a chiave simmetrica;
  - progetto completamente pubblico;
  - chiavi di lunghezza 128, 192 e 256 bit;
  - implementazione hw e sw;
  - algoritmo doveva essere liberamente utilizzato o non avere restrizioni particolari.

# AES - Rijndael

---

- L'algoritmo prescelto è stato il Rijndael (Rijmen e Daemen)
- la selezione si è basata su criteri di sicurezza, efficienza, semplicità, flessibilità e requisiti di memoria (per sistemi embedded).
- Supporto chiavi e blocchi di cifratura da 128 a 256 bit, a passi di 32 bit
- blocchi e chiavi possono avere diversa lunghezza.
- L'utilizzo più frequente prevede chiavi di 128 o 256 bit e blocchi di 128 bit.

# Cifratura a chiave simmetrica

---

- Collocazione dei dispositivi di cifratura, due possibilità:
  - Sulle linee (il pacchetto rimane vulnerabile nei commutatori )
  - Sui dispositivi terminali (non è possibile cifrare anche le intestazioni ma solo i dati)
- L'ottimo è utilizzare ambedue i metodi.

# Cifratura a chiave pubblica

---

- Utilizza due chiavi:
  - Una chiave  $K_A$  usata per la cifratura che viene resa pubblica (chiave pubblica).
  - Una chiave  $K_B$  usata per la decifratura che viene mantenuta segreta (chiave privata).
- Si evita (ma solo parzialmente!) il problema della distribuzione della chiave.
- Deve avere tre requisiti
  - $D_{K_B}(E_{K_A}(P)) = P$
  - Non deve essere possibile dedurre  $K_B$  da  $K_A$ .
  - $K_B$  non deve poter essere dedotta tramite cifratura di testi noti

# Rivest, Shamir, Adelson (RSA)

---

## Scelta delle chiavi

- Si scelga due numeri primi grandi (ad esempio da 1024 bit):  $p$  e  $q$ .
- Si calcoli  $n = p \cdot q$ ,  $z = (p - 1)(q - 1)$ .
- Si scelga  $e$  (con  $e < n$ ) tale che non abbia fattori comuni con  $z$  ( $e$  e  $z$  sono “primi relativi”).
- Si scelga  $d$  tale che  $ed - 1$  sia esattamente divisibile per  $z$  (in altre parole  $e \cdot d \bmod z = 1$ ).
- La chiave pubblica  $K_A = (n, e)$  e la chiave privata  $K_B = (n, d)$ .

# Rivest, Shamir, Adelson (RSA)

---

- Dati  $(n, e)$  e  $(n, d)$ :
  - Per cifrare una sequenza di bit  $m$ , si calcola:  
$$c = m^e \bmod n$$
 (ossia il resto di  $m^e$  diviso  $n$ )
  - Per decifrare una sequenza di bit  $c$  ricevuta, si calcola:  
$$m = c^d \bmod n$$
 (ossia il resto di  $c^d$  diviso  $n$ )
- Ciò che accade è che  
$$m = (m^e \bmod n)^d \bmod n$$

# Rivest, Shamir, Adelson (RSA)

---

Bob sceglie  $p = 5$ ,  $q = 7$ .

Quindi  $n = 35$ ,  $z = 24$ .

$e = 5$  (così  $e$ ,  $z$  sono primi relativi).

$d = 29$  (così  $ed-1$  è divisibile esattamente per  $z$ ).

	<u>Lettera</u>	<u>m</u>	<u>m<sup>e</sup></u>	<u>c = m<sup>e</sup> mod n</u>
Cifra:	I	12	248832	17

	<u>c</u>	<u>c<sup>d</sup></u>	<u>m = c<sup>d</sup> mod n</u>	<u>Lettera</u>
Decifra:	17		12	

481968572106750915091411825223072000 I

# Rivest, Shamir, Adelson (RSA)

---

- Perché vale  $m = (m^e \bmod n)^d \bmod n$  ?
- La base è un risultato della teoria dei numeri, ossia se  $p$  e  $q$  sono primi e  $n = p q$  allora:

$$x \bmod n = x^{e \bmod (p-1)(q-1)} \bmod n$$

- $(m^e \bmod n)^d \bmod n = m^{ed} \bmod n =$   
 $= m^{ed \bmod (p-1)(q-1)} \bmod n =$   
(grazie al risultato della teoria dei numeri di cui sopra)  
 $= m^1 \bmod n =$   
(dato che si è scelto  $ed$  divisibile per  $(p-1)(q-1)$  con resto 1)  
 $= m$



# Rivest, Shamir, Adelson (RSA)

---

- Si osservi che l'algoritmo funziona anche a chiavi invertite.
- Il meccanismo è sicuro perché, al momento, non sono noti algoritmi veloci per la fattorizzazione dei numeri (altrimenti basterebbe fattorizzare  $n$ )
- Il problema della cifratura a chiave pubblica è il tempo di elaborazione, rispetto alla chiave simmetrica:
  - In software è 100 volte più lenta
  - In hardware è da 1000 a 10.000 volte più lenta
- Allora viene usato, in genere, solo per lo scambio di una chiave simmetrica di sessione.

# Integrità e firma elettronica

---

- La firma elettronica è la forma più completa di verifica di integrità. Tale tipo di firma dovrebbe far sì che:
  - L'integrità del messaggio originale sia assicurata.
  - La firma sia legata indissolubilmente al messaggio.
  - La firma sia verificabile (permette di identificare chi ha firmato).
  - La firma sia non falsificabile e non rifiutabile (solo quell'individuo deve poter fare quella firma e non deve poterla disconoscere).

# Firma elettronica

---

- Un modo per firmare il proprio documento è quello di codificarlo con la propria chiave privata.
- Dato che solo il proprietario ha la chiave privata, questo assicura che solo lui può averlo codificato, e chiunque può verificare che è stato lui a codificarlo usando la sua chiave pubblica e ritrovando il messaggio.
- Questo procedimento ha un limite:
  - La cifratura di un messaggio (con chiave pubblica) è una operazione onerosa se fatta su grandi quantità di dati. E lo stesso vale per la decifratura, obbligatoria per poter leggere il messaggio

# Firma elettronica

---

- Un meccanismo alternativo che impone un minor onere computazionale è quello del *message digest* (sunto del messaggio).
- Il principio è simile a quello dei codici a rivelazione d'errore, si applica ad un messaggio  $p$  una funzione  $H()$  il cui risultato è un blocco di dati  $d_p$  (il *digest*) con dimensioni molto minori di  $p$ . Tale *digest* deve essere legato in modo univoco la messaggio originale
- Tale funzione  $H()$  viene chiamata funzione di **hash** .

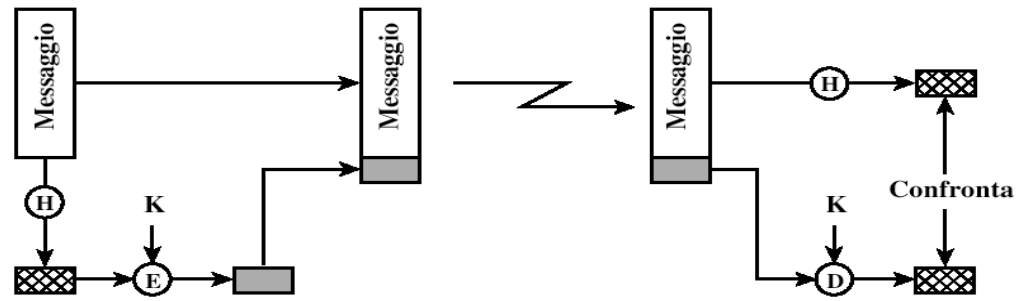
# Integrità e Firma elettronica : Digest

---

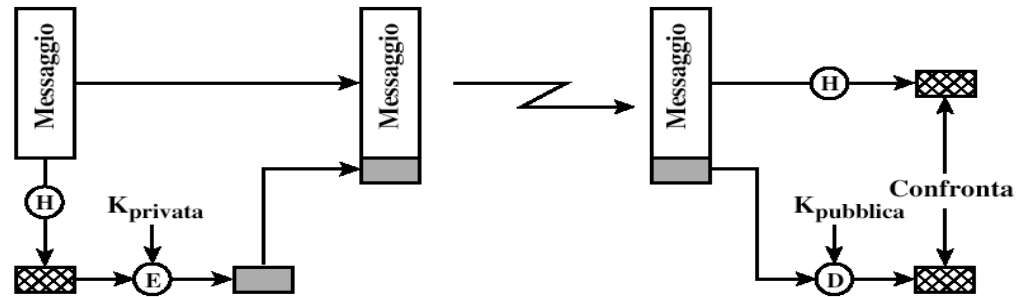
- La funzione di *hash*  $H()$  deve avere le seguenti proprietà:
  - Deve poter essere applicata a messaggi di qualunque dimensione.
  - Deve produrre un risultato di lunghezza fissa
  - Deve essere relativamente semplice da calcolare.
  - Per ogni *digest*  $d$  dato, deve essere computazionalmente impossibile trovare  $x$  tale che  $H(x) = d$  (non invertibilità).
  - Per ogni messaggio  $x$  deve essere computazionalmente impossibile trovare  $y \neq x$  tale che  $H(y) = H(x)$  (impedisce falsificazioni).
  - Deve essere computazionalmente impossibile trovare una qualsiasi coppia  $(x, y)$  tale  $H(x) = H(y)$ .

# Integrità e Firma elettronica: Digest

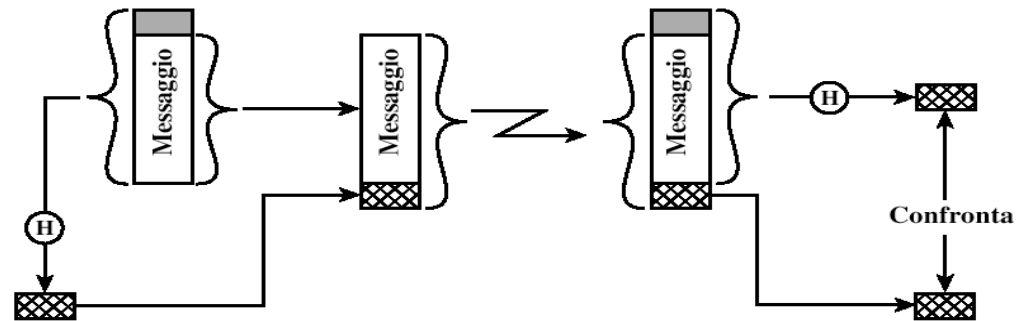
## Possibili usi del *digest* per la verifica dell'integrità



(a) Uso della crittografia convenzionale



(b) Uso della crittografia a chiave pubblica

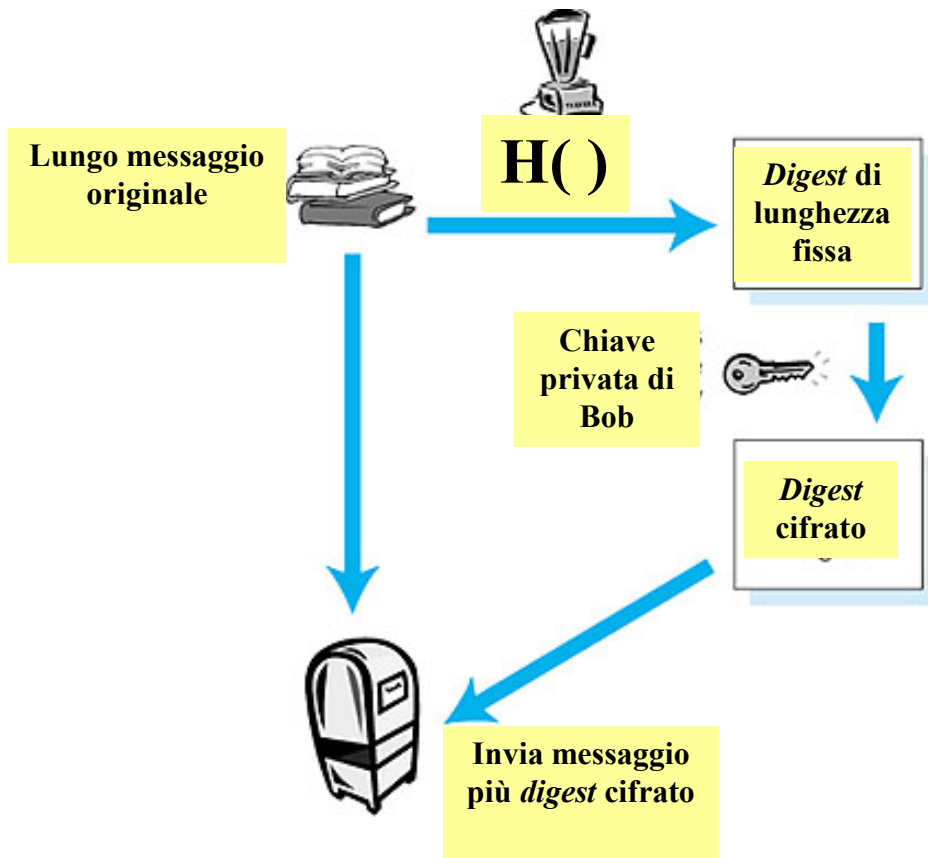


(c) Uso di un valore segreto

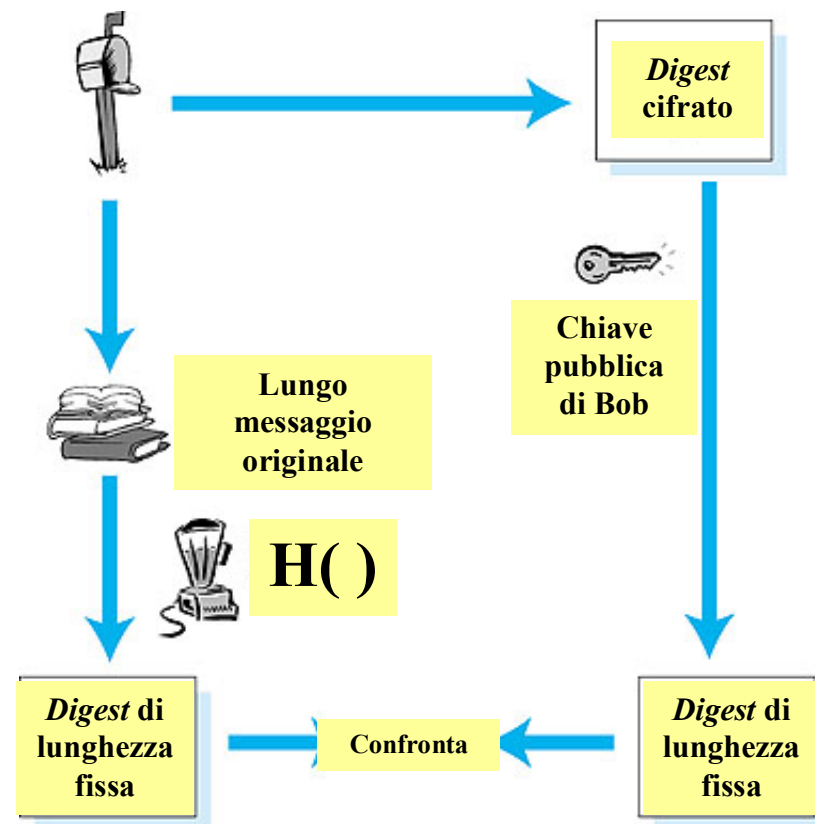
# Integrità e Firma elettronica: *Digest*

- Si può usare il *digest* cifrato con la chiave privata corrisponde a firmare il messaggio

## Bob “firma” ed invia



## Alice riceve e verifica la firma



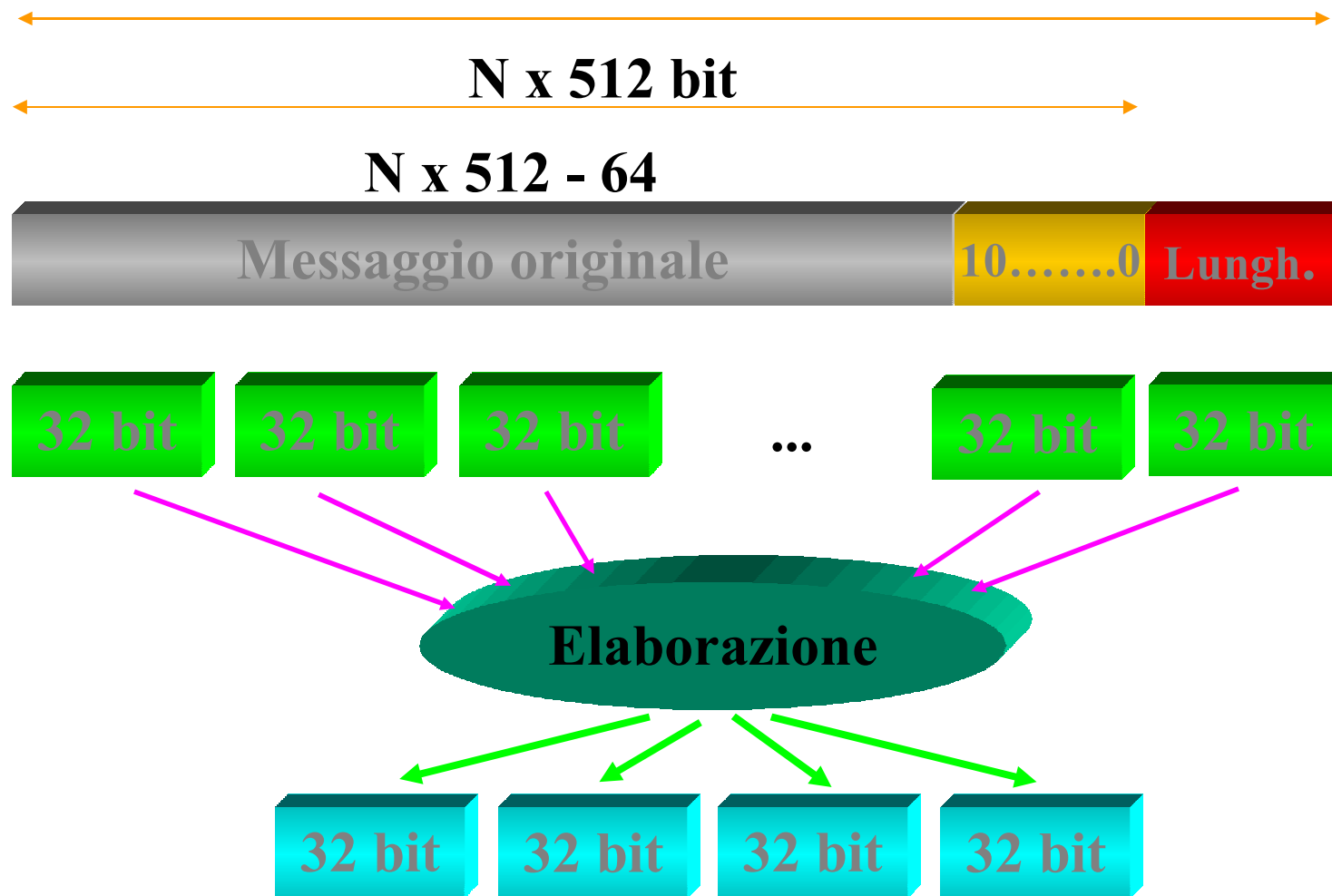
# Integrità e Firma elettronica: *Digest*

---

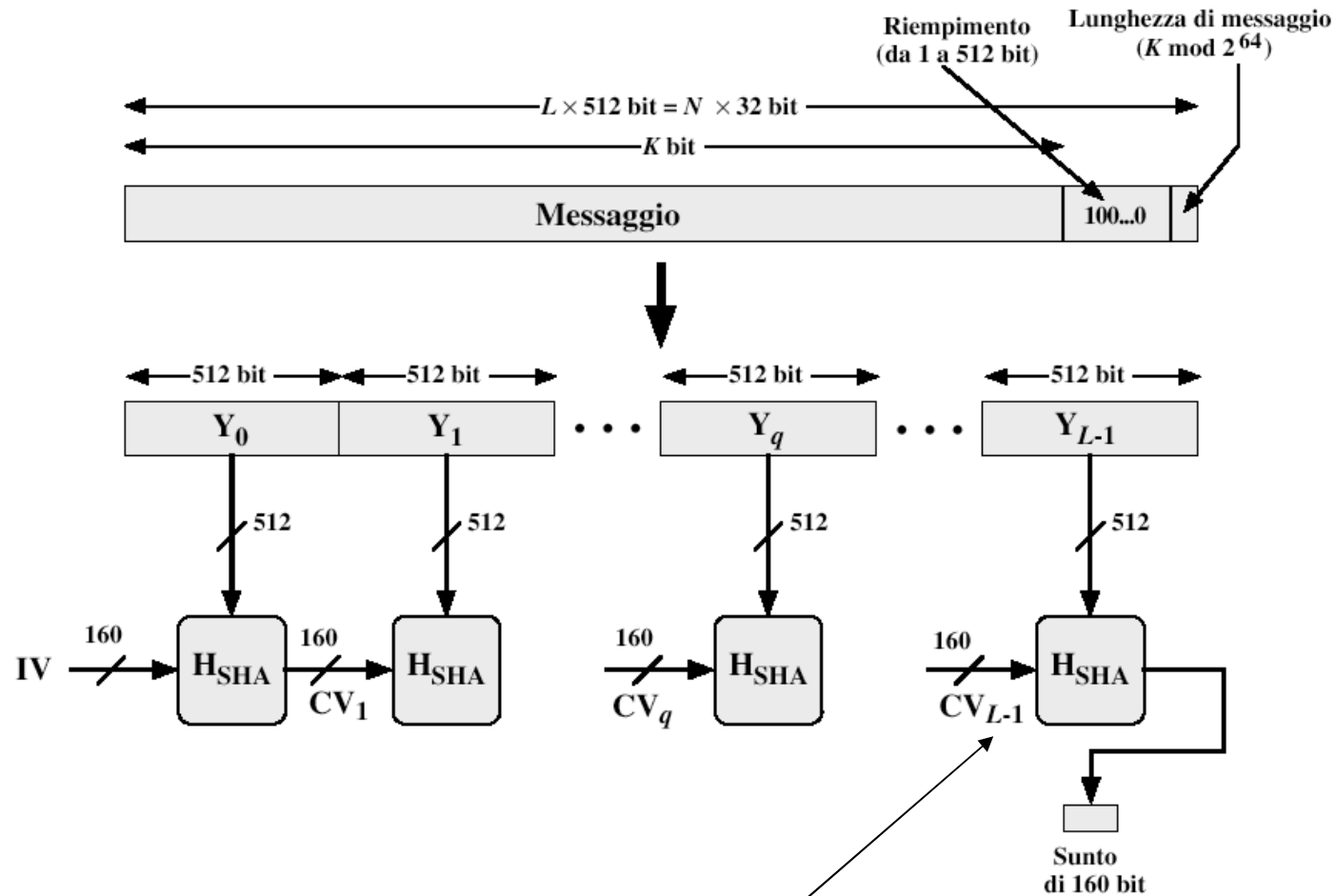
- Gli standard più usati per il *digest* attualmente sono sostanzialmente due:
  - ***Secure Hash Algorithm (SHA)***: sviluppato dal NIST e rivisto successivamente e standardizzato come FIPS PUB 180-1 noto come **SHA-1**, e usa *digest* da 160 bit.
  - **MD5** definito da Ron Rivest [RFC 1321] che usa un *digest* di 128 bit.



# Integrità e Firma elettronica: *Digest* - MD5



# Integrità e Firma elettronica: *Digest - SHA-1*



Composto da 4 cicli da 20 passi

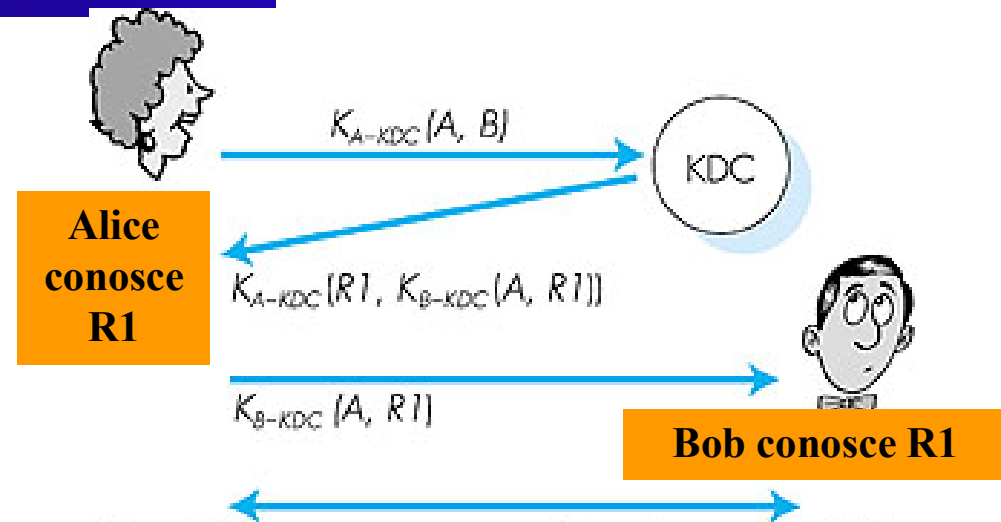
# Distribuzione delle chiavi e certificazione

---

- Due entità che voglio comunicare cifrando con chiave simmetrica, come stabiliscono una chiave segreta comune?
- La soluzione è un centro di fiducia che distribuisca le chiavi (***Key Distribution Center, KDC***).
- Per la chiave pubblica-privata, il problema è un altro: come si fa ad essere sicuri della “proprietà” di una chiave pubblica?
- Anche in questo caso bisogna avere un intermediario di fiducia detto Autorità di certificazione (***Certification Authority, CA***) che certifichi l'appartenenza di una chiave pubblica.

# Key Distribution Center

- Alice e Bob hanno bisogno di una chiave simmetrica comune.
- **KDC**: un server condivide una chiave segreta con ciascuno degli utenti registrati.
- Alice, Bob conoscono la propria chiave simmetrica,  $K_{A-KDC}$   $K_{B-KDC}$ , per comunicare con il KDC.

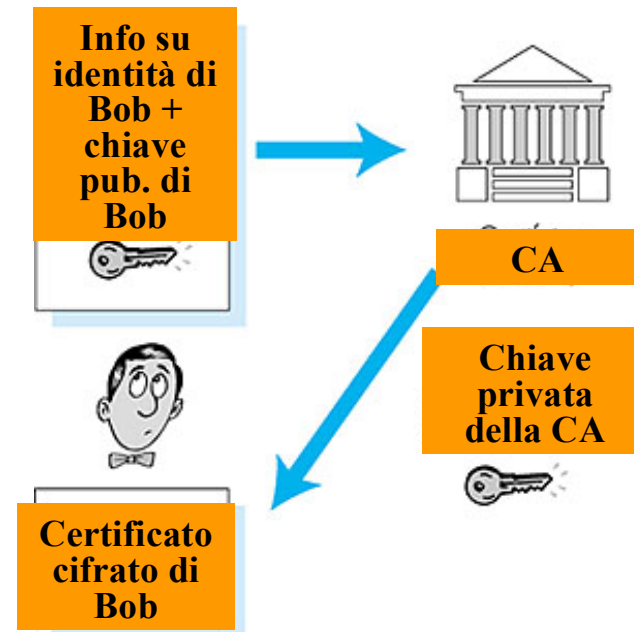


**Alice e Bob comunicano usando la chiave di sessione R1**

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

# Certification Authority (CA)

- La **Certification Authority** (CA) lega una chiave pubblica ad una entità.
- Le entità (persone, router, etc.) possono registrare le loro chiavi pubblica alla CA.
  - L'entità che si iscrive deve fornire una "prova dell'identità" alla CA.
  - La CA crea un **Certificato** che lega l'entità alla chiave pubblica.
  - Il certificato viene "firmato" dalla CA.



Quando Alice vuole la chiave pubblica di Bob:

- Prende il certificato di Bob (da Bob, dalla CA o ovunque).
- Applica la chiave pubblica del CA e ricava la chiave pubblica di Bob.

# Distribuzione delle chiavi e certificazione

---

- Si osservi che la pratica usuale è quella di:
  - Usare chiave simmetriche per la cifratura dei dati (più veloci).
  - Cambiare spesso (ogni sessione o più) la chiave simmetrica.
  - Scambiarsi la chiave simmetrica tramite una cifratura a chiave pubblica.
  - Autenticare l'identità della chiave pubblica usando una CA.

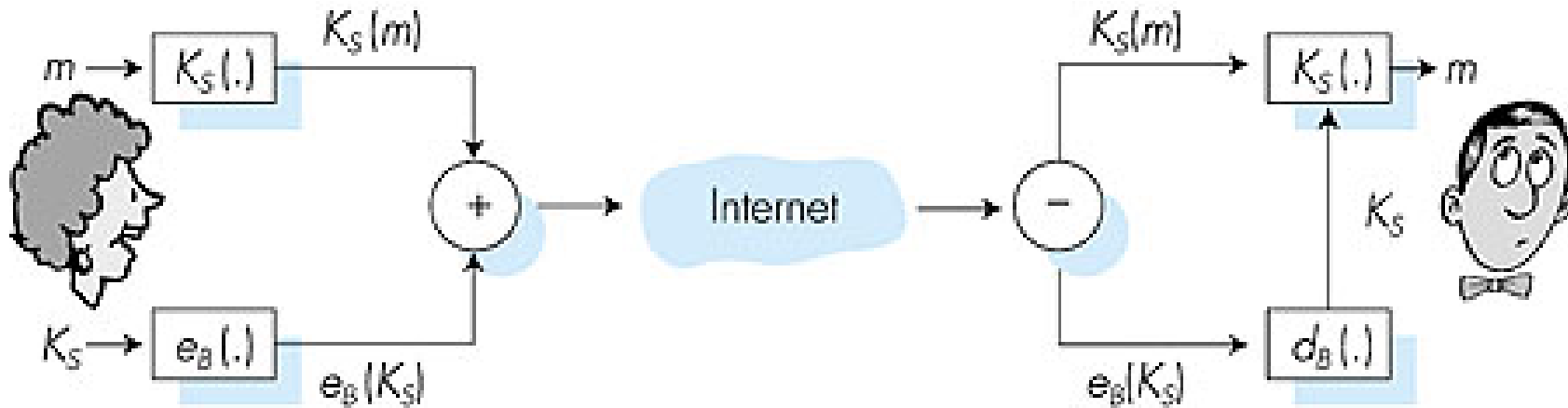
# Sicurezza - Protocolli

---

- Oltre che dal punto di vista della locazione fisica dei meccanismi di sicurezza, riveste una notevole importanza la scelta del loro posizionamento nella pila protocollare.
- I dispositivi di sicurezza possono essere implementati:
  - A livello di applicazione (ad es. email-PGP)
  - A livello di trasporto (ad es. SSL, SET)
  - A livello di rete (IPsec)
  - A livello di linea (WLAN)

# E-mail sicure - Segretezza dei dati

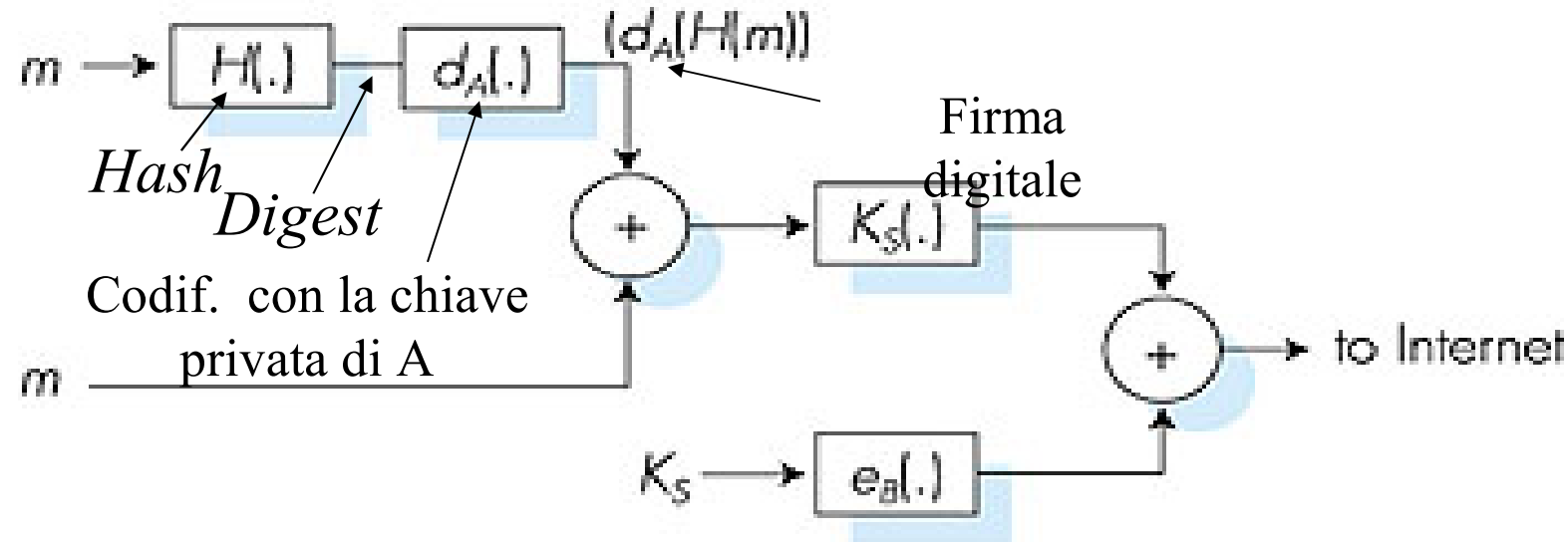
- Alice vuole inviare un messaggio  $m$  segreto a Bob



- Genera una chiave simmetrica casuale ,  $K_S$
- Cifra il messaggio con  $K_S$ ,  $K_S(m)$ .
- Cifra anche  $K_S$  con la chiave pubblica di Bob,  $e_B(K_S)$ .
- Invia sia  $K_S(m)$  che  $e_B(K_S)$  a Bob



# E-mail sicura - Segretezza, autenticazione ed integrità



- Il digest del messaggio viene cifrato con la chiave privata del mittente (firma e integrità)
- Il messaggio viene cifrato con una chiave simmetrica insieme alla firma; il tutto viene cifrato con la chiave pubblica del destinatario (segretezza)

# E-mail sicura - PGP

---

## *Pretty Good Privacy* (PGP)

- E' uno schema di di cifratura per e-mail, uno standard de facto.
- Usa la cifratura simmetrica (Triple-DES o IDEA) e a chiave pubblica (RSA), le funzioni di *Hash* (MD5 o SHA) e la firma digitale come descritto prima
- Quindi fornisce riservatezza, autenticazione del mittente e verifica dell'integrità del messaggio
- Inventato da Phil Zimmerman, oggetto per tre anni di indagini da parte federale (USA).

```
---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+l08gE4vB3mqJh  
    FEvZP9t6n7G6m5Gw2  
---END PGP SIGNATURE---
```

# *Secure Socket Layer (SSL)*

---

- SSL opera a livello di trasporto e fornisce funzioni per la sicurezza ad ogni applicazione basata su TCP
- E' utilizzato da varie applicazioni fra cui *www server* e *browser* per servizi di *e-commerce* (shttp)
- I servizi per la sicurezza di SSL sono:
  - Autenticazione del server (tramite certificato firmato da CA fidate)
  - Cifratura dei dati
  - Autenticazione dei client (opzionale)
- E' la base della ***Transport Layer Security (TSL)*** dell'IETF

# Secure Socket Layer (SSL)

---

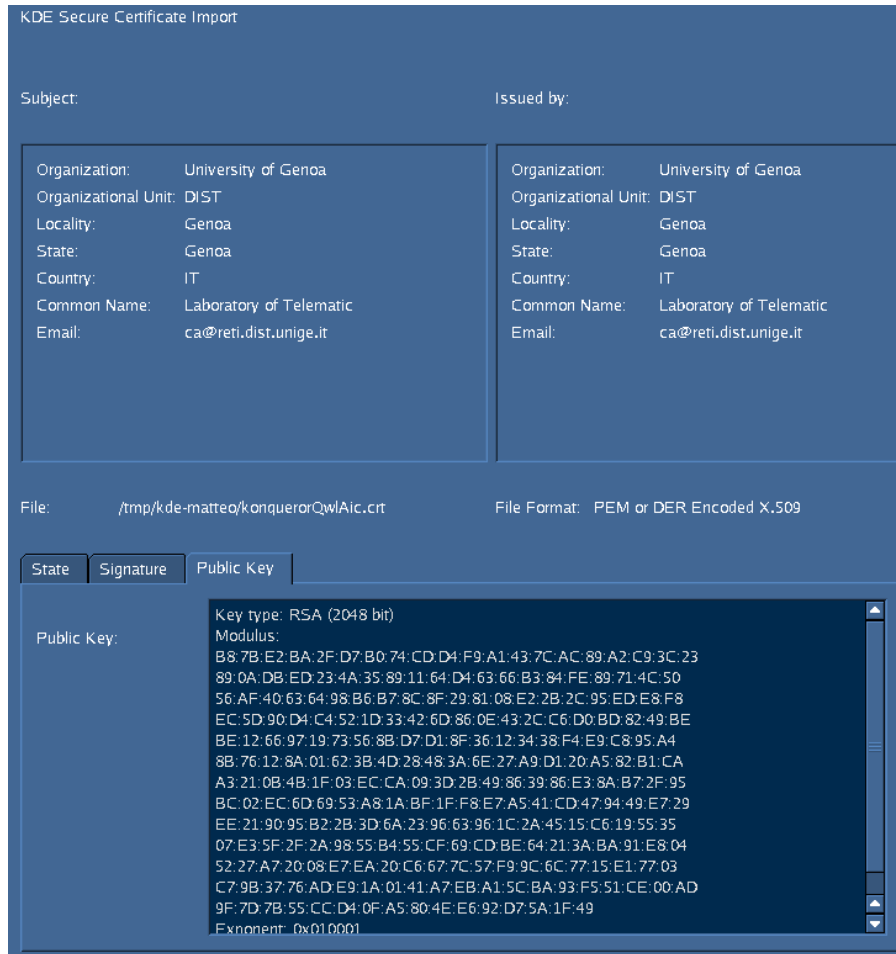
## Autenticazione del server

- Un *browser* con SSL deve possedere la chiave pubblica di una o più CA.
- Il *browser* richiede il certificato del Server secondo uno dei CA che conosce.
- Il *browser* usa la chiave pubblica del CA per estrarre la chiave pubblica del Server.

## Sessioni SSL

- Per effettuare lo scambio sicuro, SSL crea delle sessioni che possono essere usate anche da più connessioni TCP contemporaneamente
- La sessione prevede:
  - la generazione di una chiave simmetrica da parte del *browser*, cifrata con la chiave pubblica del server e ad esso inviata;
  - La decifratura della chiave simmetrica da parte del server
  - Uno scambio per definire se e come i messaggi verranno cifrati

# Distribuzione delle chiavi e certificazione: *Certificati*



**Certificato di una  
CA autofirmato**

# Distribuzione delle chiavi e certificazione: *Certificati*

🔒 Current connection is secured with SSL.

Chain:

Peer Certificate:

Organization:	University of Genoa
Organizational Unit:	Laboratory of Telematic
Locality:	Genoa
State:	Genoa
Country:	IT
Common Name:	mail.reti.dist.unige.it
Email:	webmaster@reti.dist.unige.it

Issuer:

Organization:	University of Genoa
Organizational Unit:	DIST
Locality:	Genoa
State:	Genoa
Country:	IT
Common Name:	Laboratory of Telematic
Email:	ca@reti.dist.unige.it

IP Address: 130.251.8.2  
URL: <https://mail.reti.dist.unige.it:443>  
Certificate State: **Certificate signing authority is unknown or invalid.**  
Valid from: Monday 20 May 2002 14:50:58 GMT  
Valid until: Tuesday 20 May 2003 14:50:58 GMT  
Serial Number: 2  
MD5 Digest: D1:28:B5:6B:B1:DB:1D:C9:DD:61:E6:F3:3F:C3:05:30  
Cipher in Use: RC4-MD5  
Details: RC4-MD5 SSLv3 Kx=RSA Au=RSA Enc=RC4(128) Mac=MD5  
SSL Version: TLSv1/SSLv3  
Cipher Strength: 128 bits used of a 128 bit cipher

Cryptography Configuration... Close

**Certificato di un  
server web firmato  
da una CA non  
riconosciuta dal  
browser**

# Distribuzione delle chiavi e certificazione: *Certificati*

 Current connection is secured with SSL.

Chain:

Peer Certificate:	Issuer:
Organization: University of Genoa	Organization: University of Genoa
Organizational Unit: Laboratory of Telematic	Organizational Unit: DIST
Locality: Genoa	Locality: Genoa
State: Genoa	State: Genoa
Country: IT	Country: IT
Common Name: mail.reti.dist.unige.it	Common Name: Laboratory of Telematic
Email: webmaster@reti.dist.unige.it	Email: ca@reti.dist.unige.it

IP Address: 130.251.8.2  
URL: <https://mail.reti.dist.unige.it/>  
Certificate State: The certificate is valid.  
Valid from: Monday 20 May 2003 14:50:58 GMT  
Valid until: Tuesday 20 May 2003 14:50:58 GMT  
Serial Number: 2  
MD5 Digest: D1:28:B5:6B:B1:DB:1D:C9:DD:61:E6:F3:3F:C3:05:30  
Cipher in Use: RC4-MD5  
Details: RC4-MD5 SSLv3 Kx=RSA Au=RSA Enc=RC4(128) Mac=MD5  
SSL Version: TLSv1/SSLv3  
Cipher Strength: 128 bits used of a 128 bit cipher

**Certificato di un  
server web firmato  
da una CA  
riconosciuta dal  
browser**

# Sicurezza a livello di rete: IPsec (IP security)

---

- La cifratura continua ad essere *end-to-end* ma viene effettuata nel livello di rete sui pacchetti IP e quindi diventa disponibile a tutti i protocolli che usano IP (oltre TCP, UDP, ICMP, SNMP, ...).
- Per quanto concerne l'autenticazione, in questo caso questa può avvenire anche nei confronti di indirizzi IP.
- IPsec si compone di due protocolli:
  - *Authentication Header (AH) protocol*
  - *Encapsulation Security Payload (ESP) protocol*



# Sicurezza a livello di rete: IPsec (IP security)

---

- Alcuni esempi di utilizzo di IPsec sono:
  - Interconnessione sicura di reti aziendali tramite Internet ( in sostanza permette la realizzazione di *Virtual Private Network* (VPN)).
  - Accesso remoto sicuro in Internet.
  - Interconnessione sicura fra organizzazioni diverse via Internet.
  - Migliore sicurezza nel commercio elettronico.

# Sicurezza a livello di rete: IPsec (IP security)

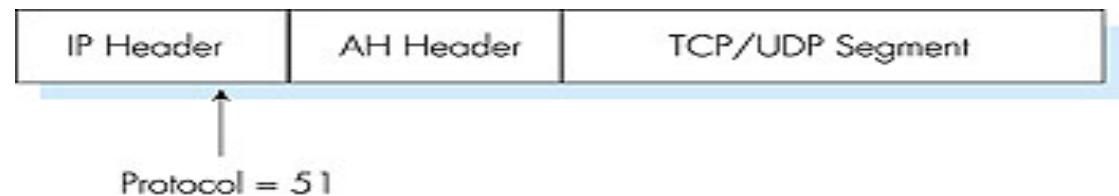
---

- Ambedue i protocolli di IPsec (ESP e AH) operano tramite una canale logico a livello di rete chiamato ***Security Association (SA)***, creato tra sorgente e destinazione con un *handshake*.
- L'SA è
  - Unidirezionale
  - Univocamente determinato da:
    - » Protocollo di sicurezza usato (ESP o AH).
    - » Indirizzo IP della sorgente.
    - » ID a 32 bit della connessione (SPI, *Security Parameter Index*).

# Sicurezza a livello di rete: IPsec - AH

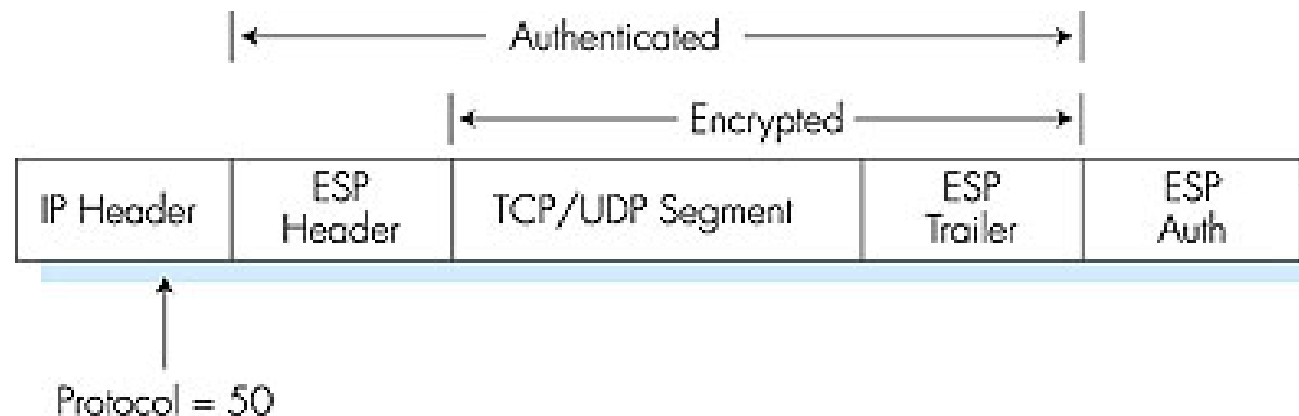
---

- Fornisce l'autenticazione dell'*host* e l'integrità dei dati ma non la riservatezza.
- L'intestazione AH viene inserita fra quella IP ed i dati
- Il numero di protocollo è il 51
- I *router* intermedi elaborano il *datagram* in modo usuale.
- L'intestazione dell'AH comprende:
  - Un identificatore di connessione
  - Un *digest* “firmato” e calcolato sul *datagram* originale
  - Un campo che specifica il tipo di dati trasportati (UDP, TCP, ICMP...)
  - Un numero di sequenza



# Sicurezza a livello di rete: IPsec - ESP

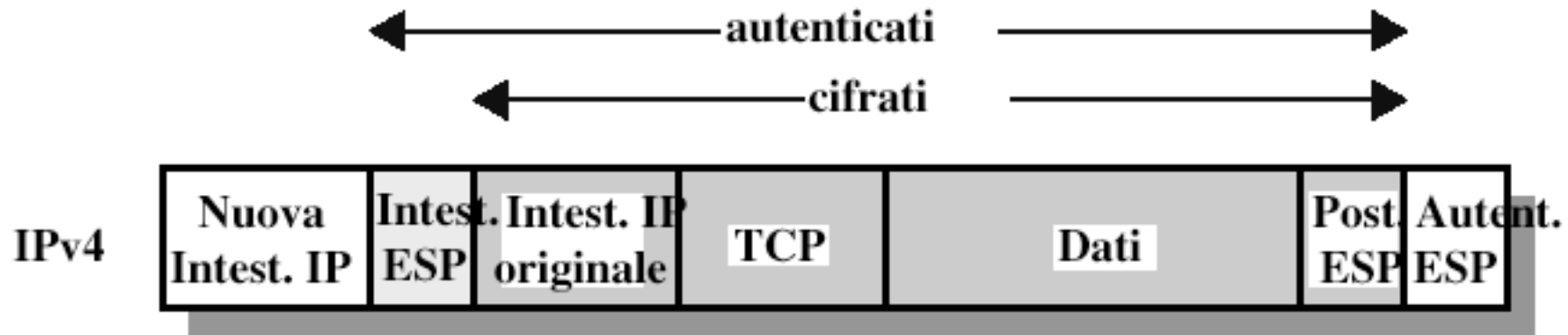
- Fornisce la riservatezza, l'autenticazione dell'host e l'integrità dei dati
- I dati e il postambolo dell'ESP sono cifrati
- L'indicazione della successiva intestazione è nel postambolo ESP.
- Il campo di autenticazione del ESP è simile ha quello dell'AH
- Il numero di protocollo contenuto nell'intestazione IP quando si usa ESP è 50



# Sicurezza a livello di rete: IPsec - Modalità di trasporto

---

- Due sono le modalità di funzionamento:
  - Modalità di trasporto
  - Modalità Tunnel
    - » applicabile se le due entità sono apparati intermedi come *router*.
    - » permette comunicazioni sicure a terminali che non usano IPsec.
    - » Permette la cifratura dell'intero pacchetto IP.



# Sicurezza a livello di rete: IPsec - SA

---

- Per il funzionamento di IPsec é necessario un meccanismo automatico per lo scambio e la gestione delle chiavi
  - *Internet Key Exchange* (IKE, RFC 2409) é il protocollo di default per lo scambio delle chiavi dell'IPsec
  - *Internet Security Association and Key Management Protocol* (ISKMP, RFC 2047 e 2048) definisce le procedure per stabilire ed interrompere gli SA. L'associazione per la sicurezza ISKMP é completamente separata dallo scambio di chiavi IKE.