Firma digitale

La firma di un documento digitale è una sequenza di bit che dipende dal documento firmato e dalla persona che firma

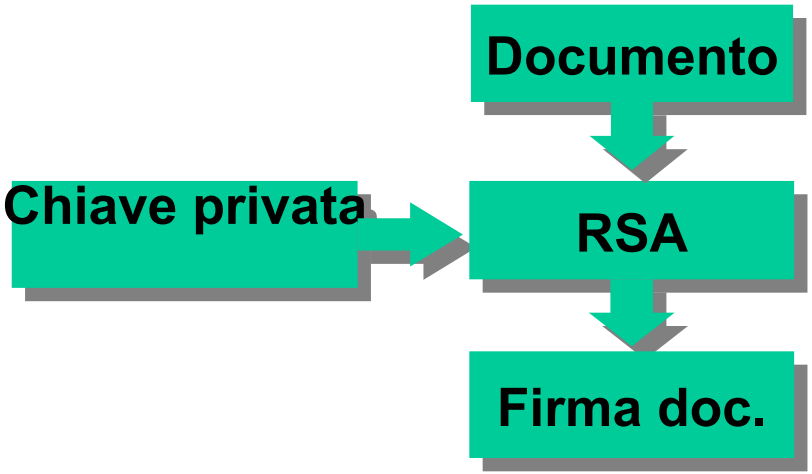
- **Chiunque deve poter verificare la validità della firma**
- **La firma non deve essere falsificabile** (le firme apposte dalla stessa persona su documenti diversi sono diverse)
- **La firma non deve essere ripudiabile** (il firmatario non puo' negare la sua firma)

Firma digitale con RSA

Firma con RSA:

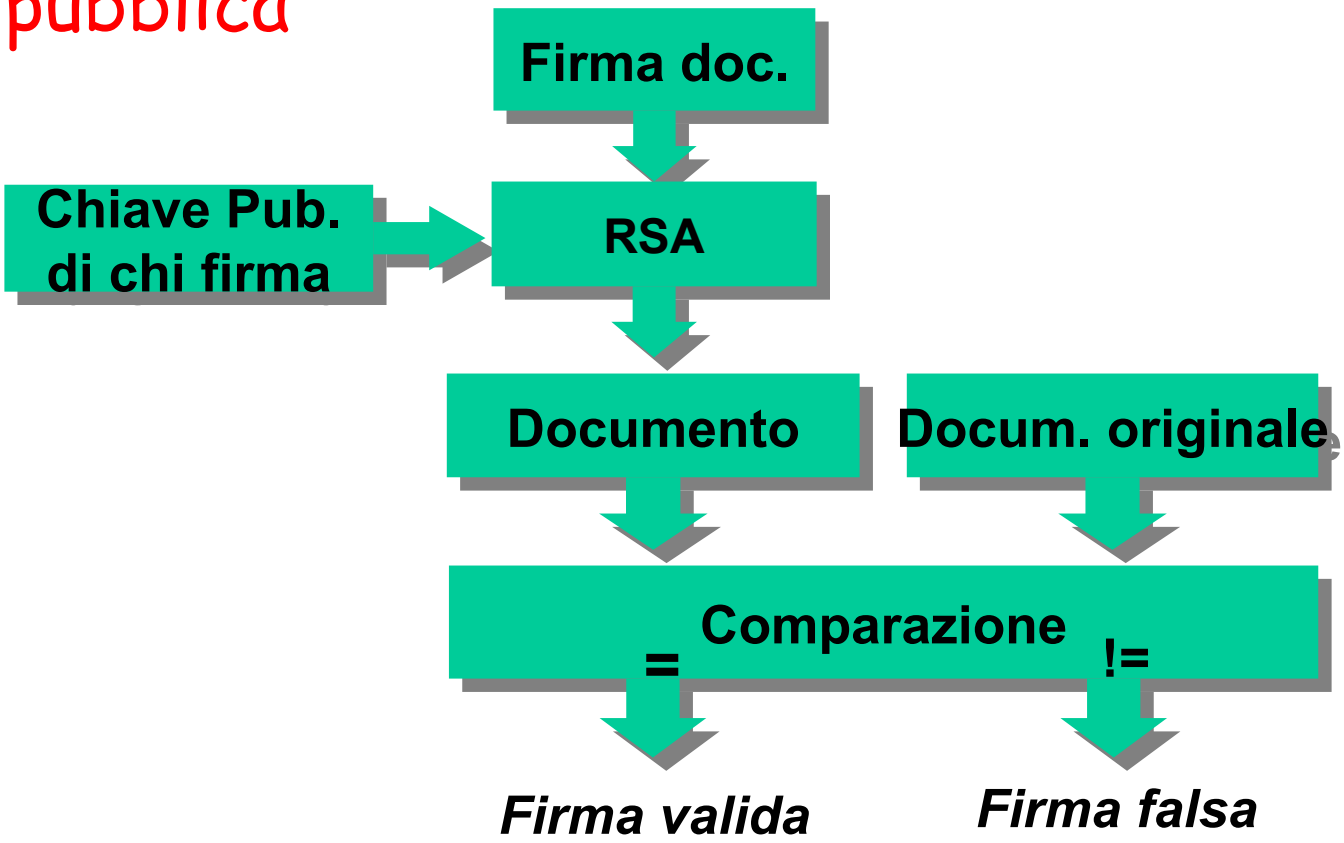
Si codifica il documento da firmare con la chiave privata

Nota bene: il messaggio è mandato in chiaro



Firma digitale con RSA (cont.)

Verifica: si codifica la firma con chiave pubblica



Firma con RSA: perché funziona

Ricordiamo che $m = (m^e \bmod n)^d \bmod n$

Ma d e e sono intercambiabili, quindi

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

- Chiunque può verificare la validità della firma
- La firma può essere apposta solo da chi conosce la chiave privata
  - Non falsificabile
  - Non ripudiabile

Impronta di un documento

Il metodo è lento sia per la firma che per la verifica (specie su documenti lunghi)

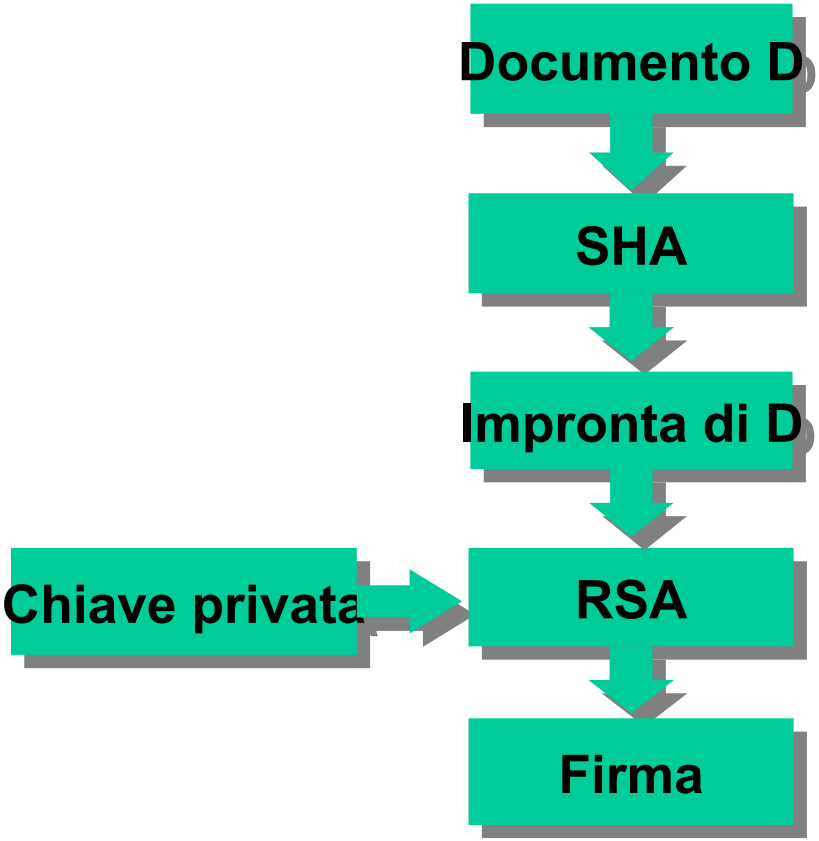
- richiede la codifica di tutto il messaggio con RSA (o altro metodo a chiave pubblica)

Idea: Firmare con la chiave segreta non tutto il documento ma una sua impronta digitale

- Calcolo impronta digitale documento (es. 512 bit)
- Codifico con la chiave segreta solo l'impronta

Un Algoritmo A per trovare l'impronta digitale di un documento D calcola una sequenza di bit A(D) strettamente correlata a D e di lunghezza fissa.

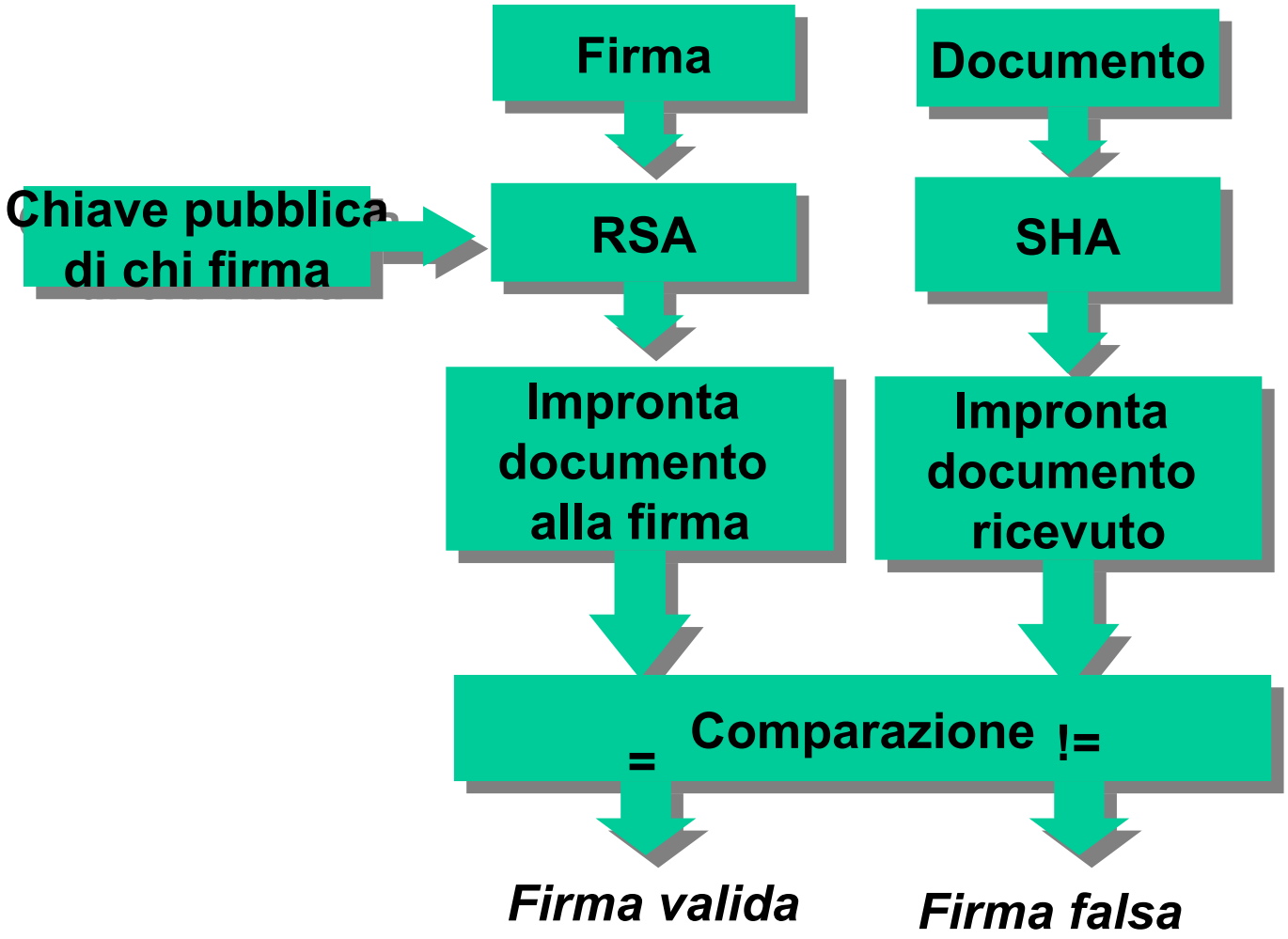
Firma di un documento



Esistono molti algoritmi per il calcolo dell'impronta

(SHA è uno dei più usati)

Verifica di autenticita' della firma



Proprieta' dell'impronta

Se documenti diversi hanno impronte diverse:

- le firme apposte dalla stessa persona su documenti diversi sono diverse
- conoscere la firma di una persona su un documento non permette di falsificare la firma di un altro documento

Esistono documenti diversi con la stessa impronta? Sì; però

I metodi usati garantiscono che dato un documento trovarne un altro con la stessa impronta è possibile solo per tentativi.

Proprieta' dell'impronta digitale

Proprietà richieste:

- Dato un documento è facile (veloce) ottenere una sequenza di lunghezza fissa (es. 128, 160) l' impronta
- Data un'impronta I e' difficile (solo per tentativi) ottenere un documento con impronta I
- E' difficile (solo per tentativi) trovare due documenti diversi con la stessa impronta

Conservando ***in modo sicuro*** l'impronta di un documento e' possibile verificare se il documento ha subito alterazioni (non e' necessario conservare tutto il documento in modo sicuro).

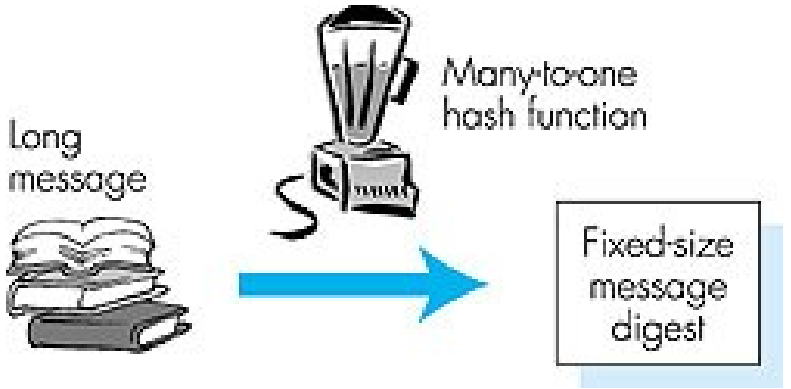
Algoritmi per calcolare un'impronta

Non vanno bene

- Internet checksum
  - Troppo facile trovare un altro documento con la stessa impronta
- DES
  - Calcola un'impronta di 64 bit; con 2^{33} tentativi si trovano due documenti con la stessa impronta

Paradosso del compleanno

Algoritmi proposti



- MD5
  - impronta di 128-bit
- SHA-1
  - US standard
  - Impronta di 160-bit
  - Molto veloce

CHECKSUM

- Messaggio

I O U 1 → 9
0 0 . 9 → 1
9 B O B

- ASCII

49 4F 55 31 → 39
30 30 2E 39 → 31
39 42 4F 42

B2 C1 D2 AC
(checksum)

900.19
"IOU900.19BOB"

100.99
"IOU100.99BOB"

I due messaggi hanno
la medesima checksum,
ma un costo decisamente superiore!

Un possibile attacco

Attacco su impronta piccola (es. 64 bit):
Trudy vuole far firmare ad Alice un contratto svaforevole

- Trudy prepara 2 versioni del contratto: M favorevole a Alice e M' sfavorevole ad Alice
- M' viene alterato con piccoli cambiamenti (es. aggiunta di spazi) fino a che Trudy ottiene un documento con la stessa impronta
- La firma che Alice ha apposto su M è la stessa firma per M': Trudy ha raggiunto il suo scopo!

Trudy deve generare 2^{64} messaggi

Un possibile attacco (cont.)

Attacco su impronta piccola (es. 64 bit):
Trudy vuole far firmare ad Alice un contratto svaforevole

- Trudy prepara 2 versioni del contratto: M favorevole a Alice e M' sfavorevole
- M' viene alterato con piccoli cambiamenti
- M viene alterato con piccoli cambiamenti
- Trudy deve ottenere una versione di M e una di M' con la stessa impronta

Quanti messaggi deve generare Trudy affinchè

M ed M' abbiano la stessa impronta?

Un possibile attacco (cont.)

Paradosso del compleanno

- Quante persone bisogna scegliere a caso affinchè con prob.>0.5 ci sia una persona con lo stesso mio compleanno? Risposta 183
- Quante persone bisogna scegliere a caso affinchè con prob.>0.5 ci siano almeno due persone con lo stesso compleanno? Risposta solo 23.

Prob(scelgo t elementi diversi fra n)=

$(1- (t-1)/n)\ (1- (t-2)/n) \ldots (1- 2/n)\ (1- 1/n)=$

$\sim e^{(-t(t-1)/2n)}$

t 2 1

con t =23 e n=365 si ottiene prob. > 0.5

Si suppongono gli n valori equamente probabili.

La formula generale è:

$$P(n,k)=1-\frac{n!}{(n-k)!\,n^{k}}$$

Con le dovute approssimazioni ottengo

$$P(n,k)>1-e^{\frac{-k(k-1)}{2n}}$$

Eguagliando questa quantità a 0,5 scopro per che valore di k ho una probabilità del 50% di avere due elementi uguali

$$k=\sqrt{\ln(2)*2n}=1{,}18\sqrt{n}\approx\sqrt{n}=\sqrt{2^{m}}=2^{\frac{m}{2}}$$

dimensione del messaggio = m

Un possibile attacco (cont.)

Attacco su impronta piccola (es. 64 bit): Trudy vuole far firmare ad Alice un contratto svaforevole

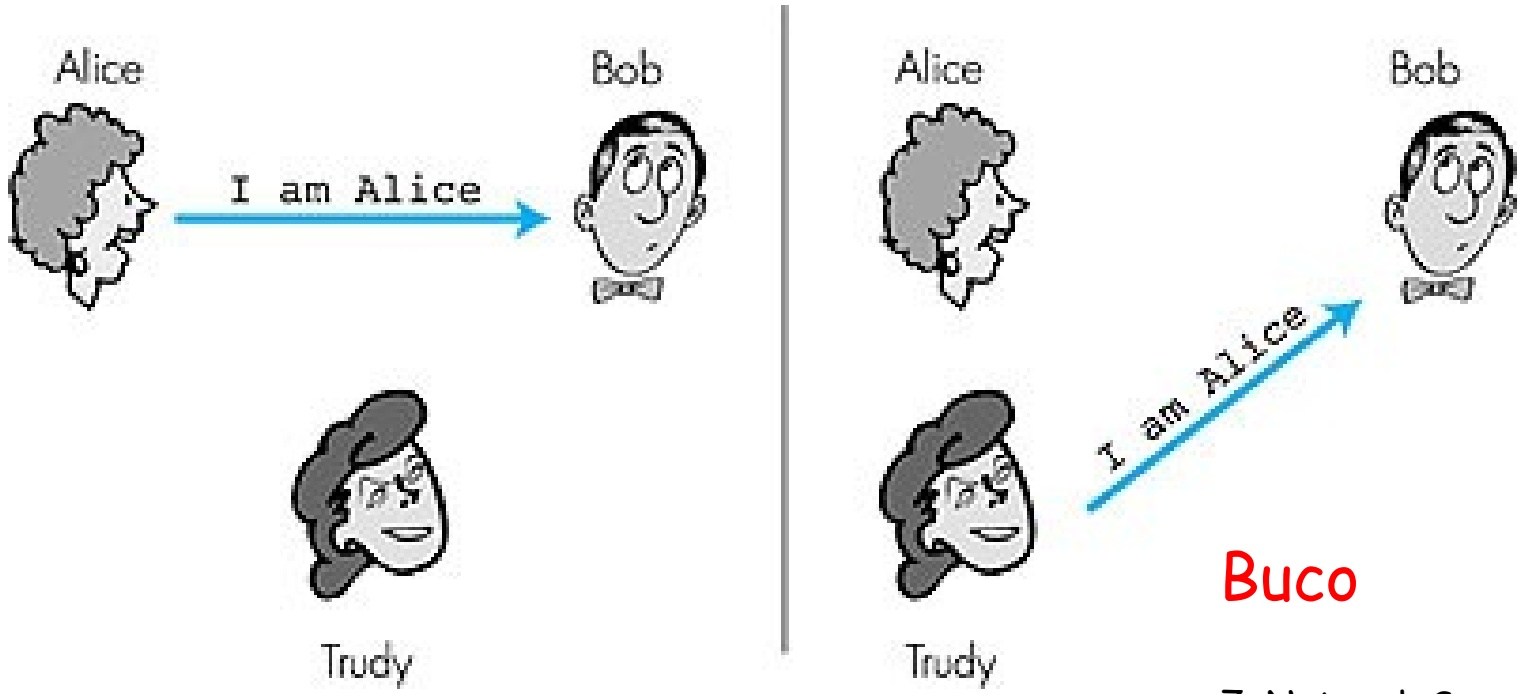
- Trudy prepara 2 copie del contratto: M favorevole a Alice e M' sfavorevole
- Trudy deve generare 2^{32} versioni di M e 2^{32} versioni di M' per avere una probabilità di successo maggiore di 0.5; in totale 2^{33} ; troppo facile!

L'impronta di un testo deve essere lunga (es. 160 bit)

Autenticazione

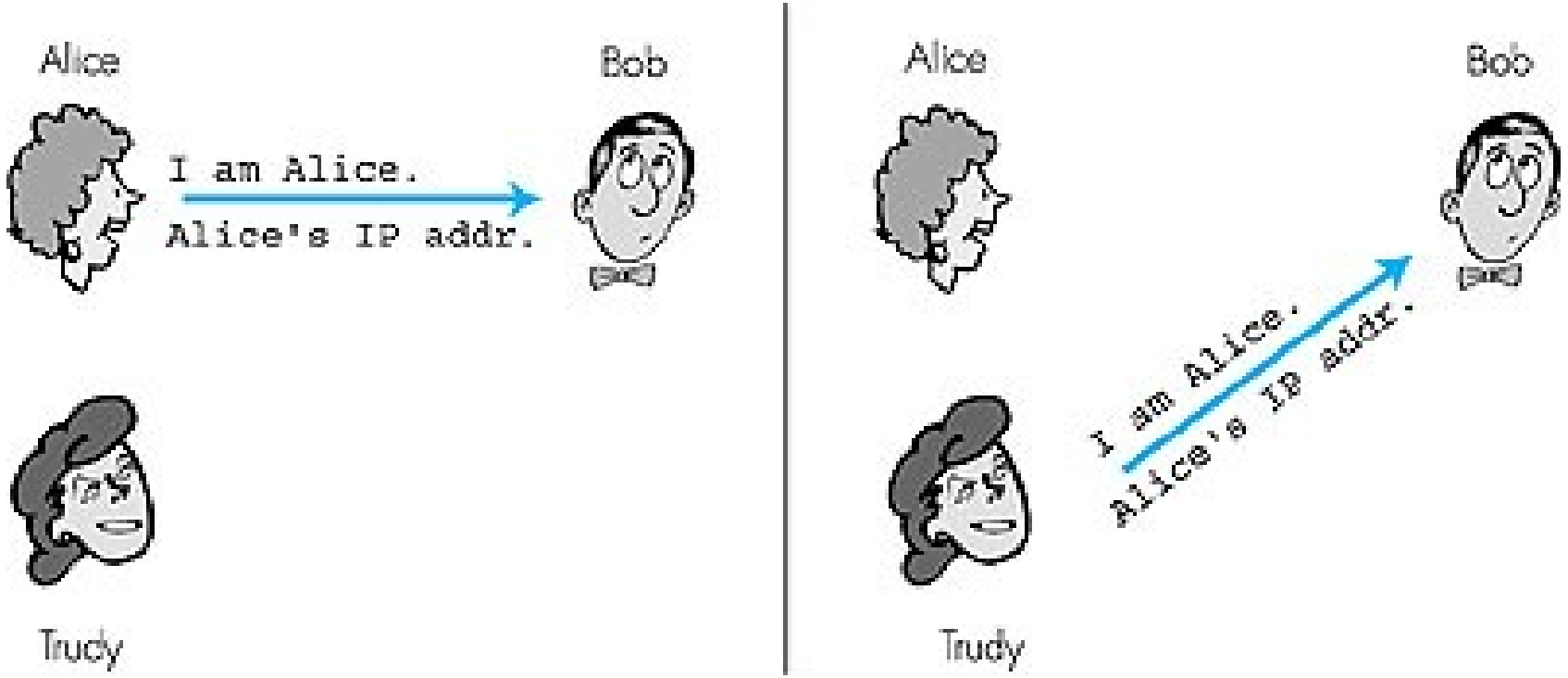
Obiettivo: Bob vuole che Alice "dimostri" la propria identità

Protocollo ap1.0: Alice dice "Sono Alice"



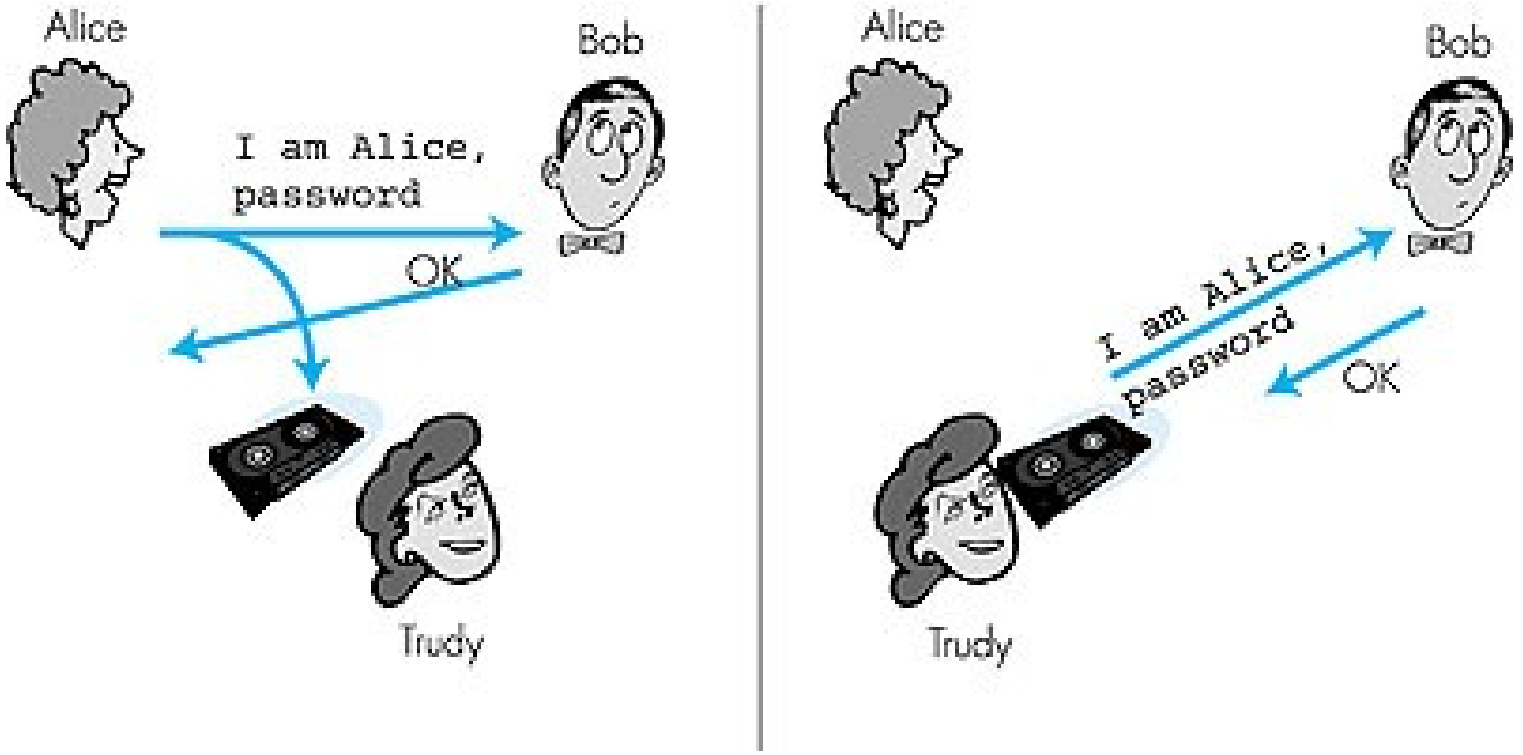
Autenticazione (cont.)

Protocollo ap2.0: Alice dice "sono Alice" e invio il Proprio indirizzo IP per "dimostrarlo".



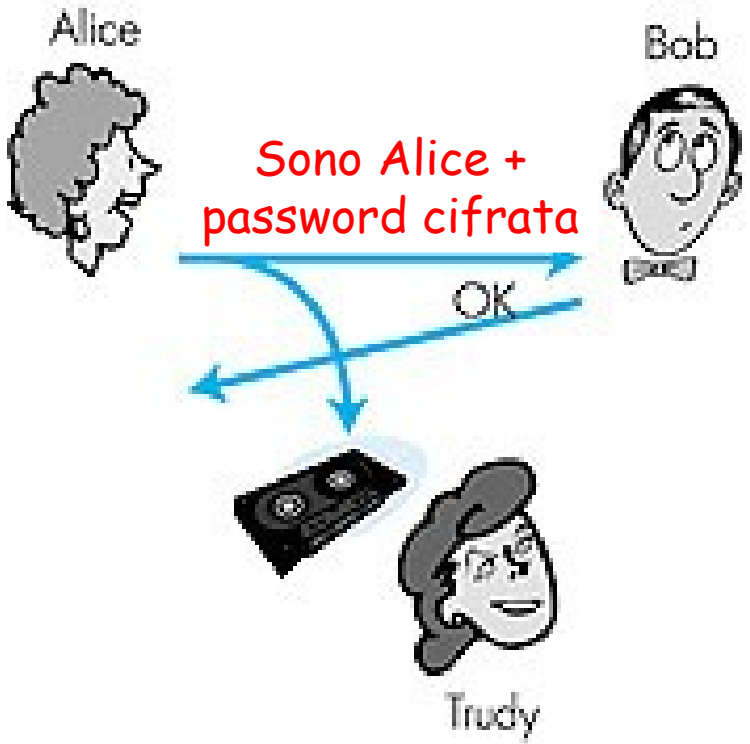
Autenticazione (cont.)

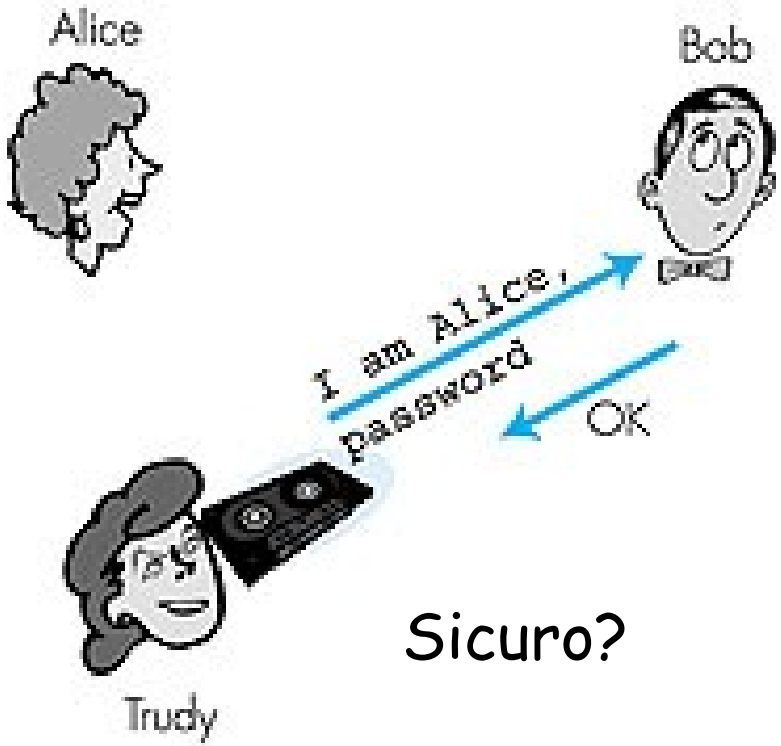
Protocollo ap3.0: Alice dice "sono Alice" e invia la password per "dimostrarlo".



Autenticazione (cont.)

Protocollo ap3.1: Alice dice "sono Alice" e invia la password *cifrata* per "dimostrarlo".





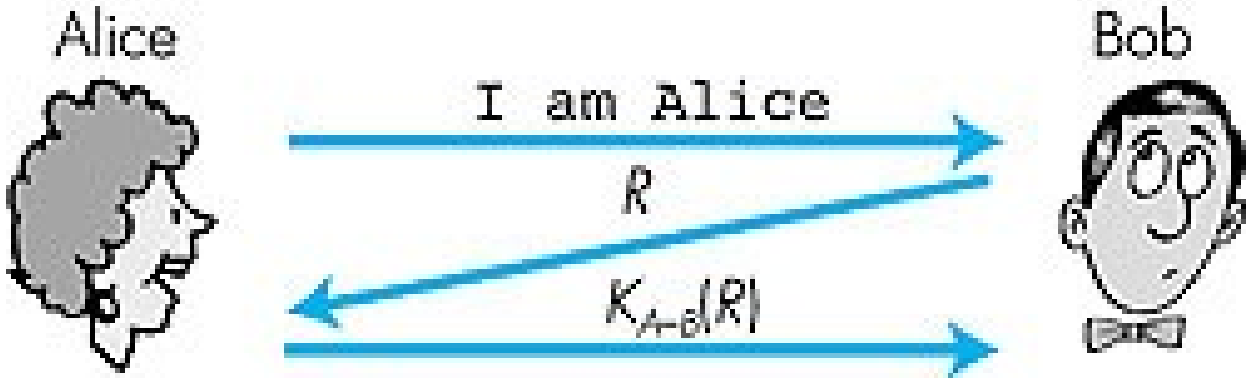
Sicuro?

Autenticazione (cont.)

Obiettivo: evitare registrazione

Nonce: numero R scelto casualmente (usato una sola volta)

ap4.0: Bob invia il nonce R ad Alice. Alice deve restituire R, cifrata con la chiave segreta condivisa



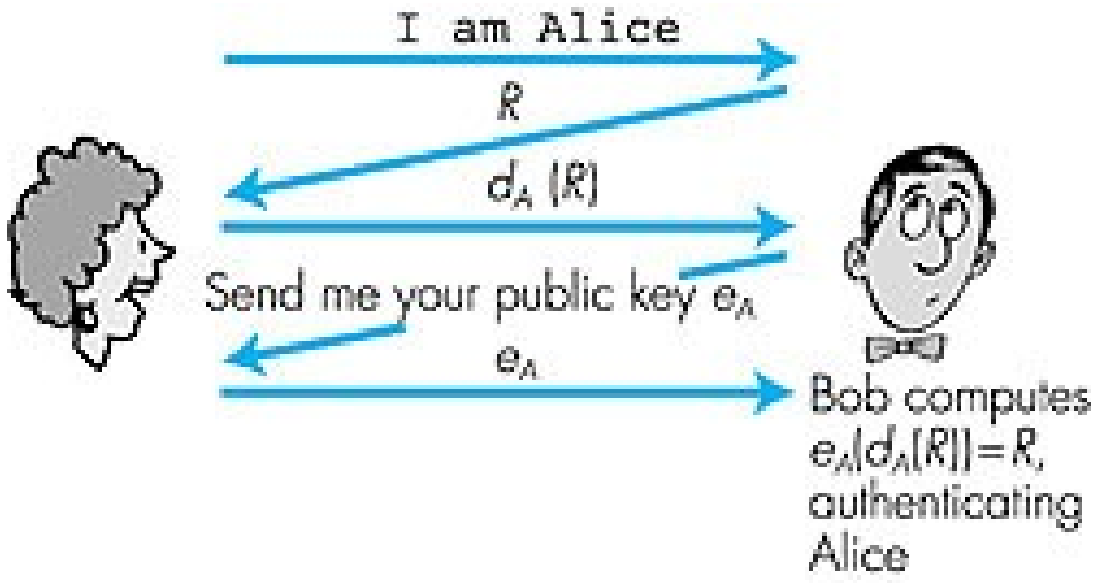
Problemi? Svantaggi?

Autenticazione (cont.)

ap4.0 richiede una chiave simmetrica condivisa

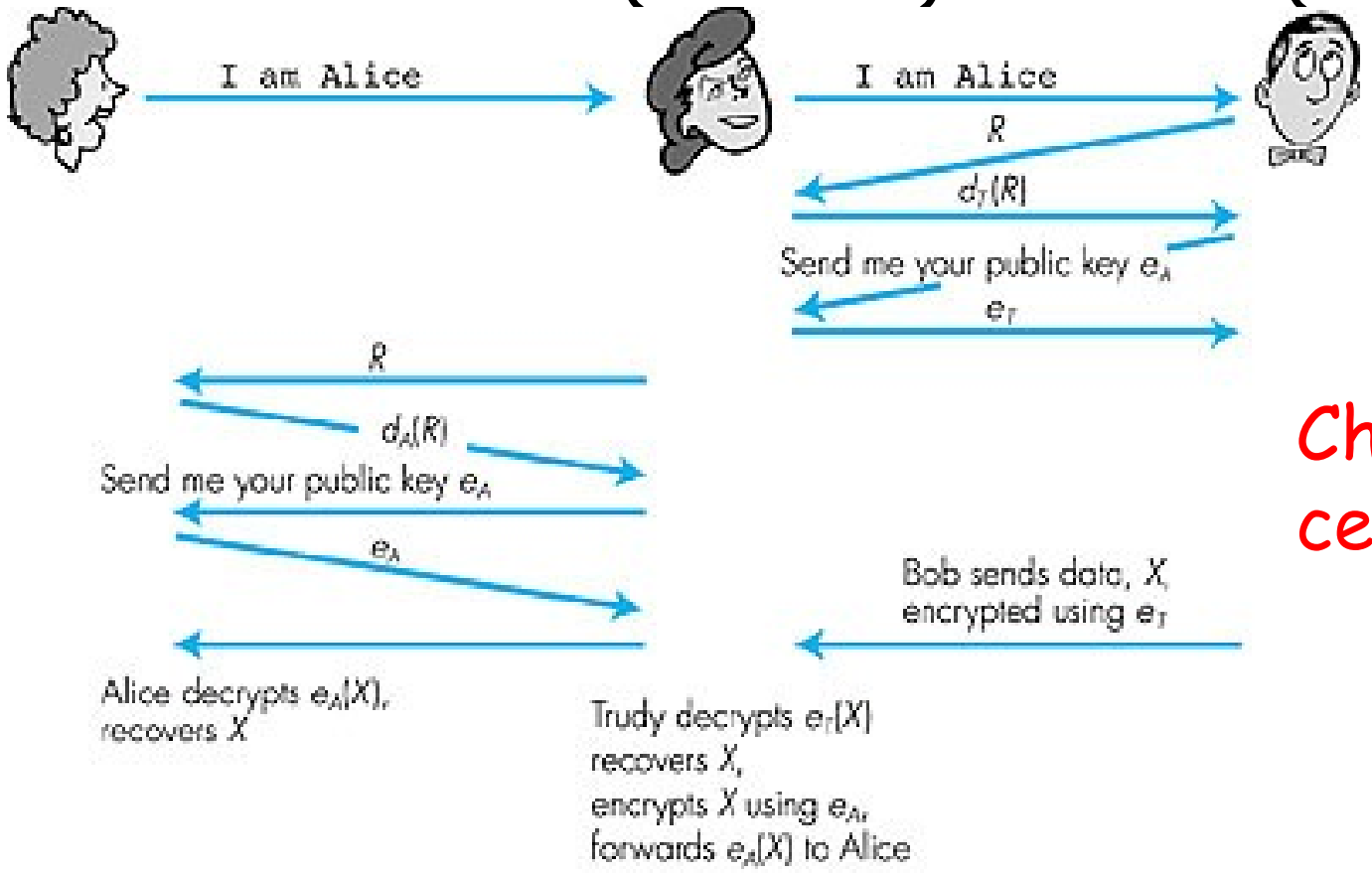
- Bisogna raggiungere accordo sulla chiave prima
- Alternativa: usare chiavi publiche

ap5.0: uso di nonce e chiave pubblica



Autenticazione (cont.)

Attacco "Man in the middle" Trudy finge di essere sia Alice (con Bob) che Bob (con Al.)



Chiavi certificate

Autenticazione: X.509

Y{I} rappresenta la firma di I da parte di Y

t timestamp, r nonce (num. casuale)

A ⟶ B: $A\{t_A, r_A, B\}$

B ⟶ A: $B\{t_B, r_B, A, r_A\}$

Se Trudy usa messaggi di autentica usati precedentemente (replay di messaggi) per sostituirsi a A allora il time stamp è cambiato

Firma di r_B da parte di A fornisce ulteriore prova a B sulla sua identità

Autenticazione: X.509 (cont.)

Autenticazione tridirezionale

$A \longrightarrow B: \quad A\{t_A, r_A, B\}$

$B \longrightarrow A: \quad B\{t_B, r_B, A, r_A\}$

$A \longrightarrow B: \quad A\{r_B\}$

Il terzo messaggio rappresenta ulteriore prova per B che sta parlando con A

Autenticazione: X.509 (cont.)

Autenticazione tridirezionale con timestamp facoltativo (versione vecchia di X.509)

$A \longrightarrow B: \quad A\{r_A, B\}$

$B \longrightarrow A: \quad B\{r_B, A, r_A\}$

$A \longrightarrow B: \quad A\{r_B\}$

La sicurezza si basa sul fatto che i nonce non sono utilizzati più volte

Sufficiente?

Autenticazione: X.509 (cont.)

A --> B: $A\{r_A, B\}$

B --> A: $B\{r_B, A, r_A\}$

A --> B: $A\{r_B\}$

Autenticazione: X.509 (cont.)

Trudy si autentica con Bob:

$T \longrightarrow B: A\{r_A, B\}$ vecchio mess. aut.

$B \longrightarrow T: B\{r_B, A, r_A\}$

Come fa T a conoscere $A\{r_B\}$?

$A \longrightarrow T: A\{r_{A'}, T\}$ T forza A a iniziare

$T \longrightarrow A: T\{r_B, A, r_{A'}\}$ connessione con T

$A \longrightarrow T: A\{r_B\}$

$T \longrightarrow B: A\{r_B\}$

Uso dei time stamp necessario!!

Autenticazione: X.509 (cont.)

Trudy si autentica con Bob:

$T \longrightarrow B: A\{r_A, B\}$ vecchio mess. aut.

$B \longrightarrow T: B\{r_B, A, r_A\}$

Come fa T a conoscere $A\{r_B\}$?

$A \longrightarrow T: \quad A\{r_{A'}, T\}$ T forza A a iniziare

$T \longrightarrow A: T\{r_B, A, r_{A'}\}$ connessione con T

$A \longrightarrow T: \quad A\{r_B\}$

$T \longrightarrow B: \quad A\{r_B\}$

Uso dei time stamp necessario!!

$A \longrightarrow B: A\{r_A, B\}$

$B \longrightarrow A: B\{r_B, A, r_A\}$

$A \longrightarrow B: A\{r_B\}$

Chiavi garantite?

Come raggiungere l'accordo su una chiave segreta?

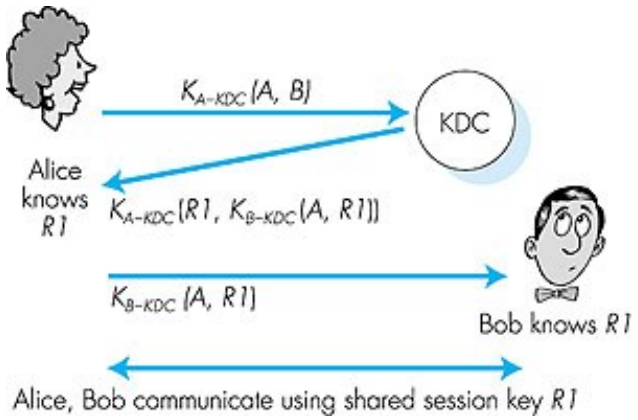
- Metodo di Diffie-Hellman
- Uso di un centro distribuzione chiavi (Key Distribution Center (KDC)) fidato

Come essere sicuri dell'autenticità di una chiave pubblica ottenuta via Web, e-mail ecc.?

- Uso di autorità di certificazione (CA) fidata

Autorità di certificazione (Key Distribution Center (KDC))

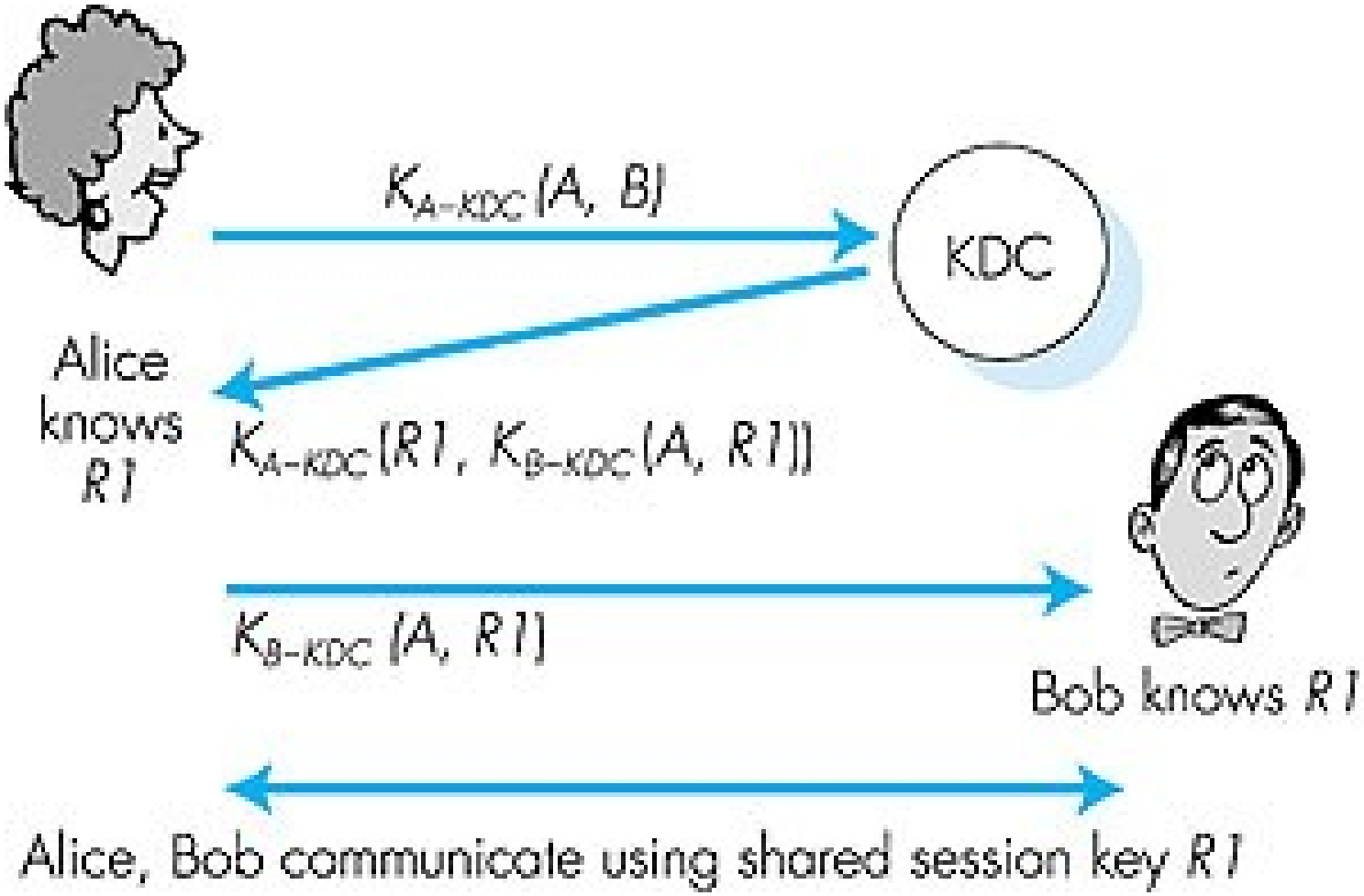
- Alice e Bob vogliono condividere una chiave segreta R (di sessione)
- KDC condivide chiavi segrete con ogni utente ($K_{A\text{-}KDC}$ [$K_{B\text{-}KDC}$] con A [B])



- Alice comunica con KDC, prende la chiave di sessione R e $K_{B\text{-}KDC}(A,R)$ (chiave, utile per comunicare con A, codificata con $K_{B\text{-}KDC}$)
- Alice comunica a Bob $K_{B\text{-}KDC}(A,R1)$
- Bob estrae R da $K_{B\text{-}KDC}(A,R)$

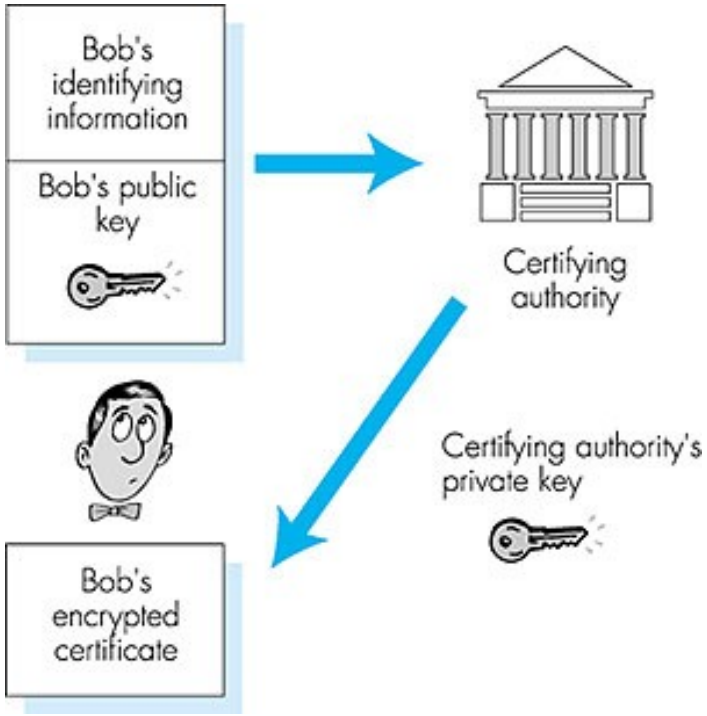
Alice e Bob condividono la chiave segreta R

KDC


$K_{A\text{-}KDC}[A, B]$
KDC
Alice knows $R1$
$K_{A\text{-}KDC}[R1, K_{B\text{-}KDC}[A, R1]]$
$K_{B\text{-}KDC}[A, R1]$
Bob knows $R1$
Alice, Bob communicate using shared session key $R1$

Autorità di Certificazione

- Autorità di certificazione (CA) garantisce chiavi pubbliche di ogni utente
  - Utente conosce la chiave pubblica della CA
- Un utente (persona, router, etc.) registra la sua chiave pubblica con CA
  - Utente fornisce a CA "garanzia di identità"
  - CA crea certificato collega utente a chiave pubblica
  - Certificato è firmato dalla CA



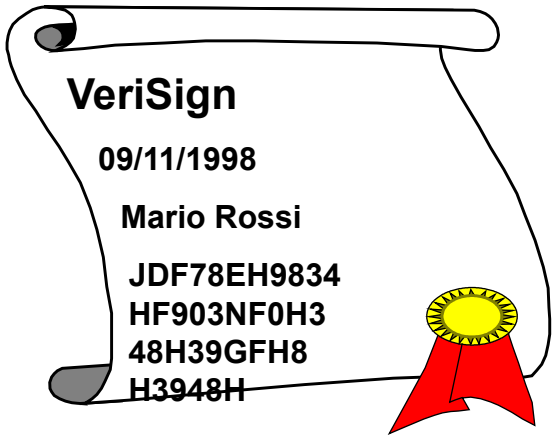
Alice vuole conoscere la chiave pubblica di Bob:

- Chiede a CA certificato di Bob
- Verifica autenticità del certificato (verifica firma CA)

Autorità di certificazione (cont.)

L'Autorità di certificazione

- garantisce la effettiva corrispondenza di una chiave pubblica con il soggetto che la espone.
- pubblica, in un apposito registro, certificati firmati con la propria chiave privata che specificano:
  - Il nome dell'Autorità
  - La data di emissione del certificato
  - La data di scadenza del certificato
  - Il nominativo del soggetto
  - La chiave pubblica del soggetto



Autorità di certificazione (cont.)

Le chiavi pubbliche possono essere sospese o revocate (ad es. furto o smarrimento)

- L'Autorità di certificazione gestisce un registro storico delle chiavi pubbliche revocate,

I certificati vanno chiesti al momento della verifica della firma

Esempio: verifica di un firma

- Si chiede alla CA la chiave pubblica del firmatario *al momento della firma.*
- Tale sequenza di operazioni viene svolta in modo automatico dal software

Autorità di certificazione (cont.)

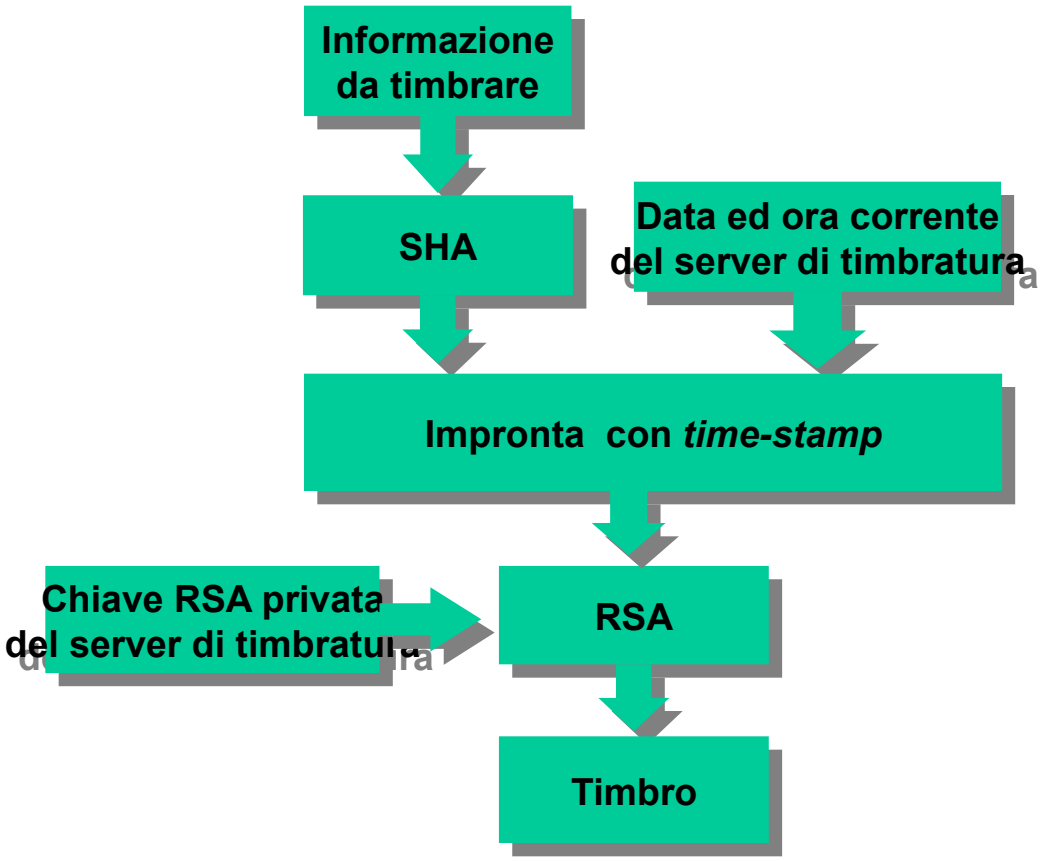
- Certificati X.509
  - standard supportato da molti protocolli (es.SSL, PKCS)
  - i campi di un certificato X.509 includono: versione, no. Seriale, tipo algo. Usato per firma, nome di chi lo rilascia, periodo di validità,nome dell'intestatario, firma dei precedenti campi.
- Legislazione italiana (1997) legge "Bassanini":

  "Gli atti, i dati e i documenti formati dalla pubblica amministarzione e dai privati con strumenti informatici e telematici, i contratti stipulati nelle medesime forme, nonché la loro archiviazione e trasmissione con strumenti informatici e telematici, sono validi e rilevanti ad ogni effetto di legge."

Certificazione temporale

- Servizio **timbratura** (time-stamp):
  - la CA garantisce il momento in cui un documento è stato creato
- Esempio
  - Il cliente sottopone al servizio di timbratura un **digest** del documento da timbrare.
  - Il servizio di timbratura ottiene un **time-stamp** da un orologio gestito dall'Autorità e riconosciuto valido.
  - Il servizio di timbratura allega il time-stamp al digest, cifra il tutto con la chiave privata dell'Autorità e lo restituisce al cliente.

Certificazione temporale (cont.)



Quanto è fidata l'autorità fidata?

Metodo di Diffie-Helmann

Alice e Bob vogliono trovare una chiave segreta comunicando su un canale non sicuro

Diffie-Hellman (1976) hanno proposto un metodo semplice e robusto:

- Dato un numero primo p sia g generatore del gruppo Z_p :
  - g è generatore se la sequenza

$g, g^2 \bmod p, g^3 \bmod p \ldots., g^{p-1} \bmod p$

permette di ottenere tutti i numeri tra 1 e p

  - Esempio: p = 11, g= 2

Metodo di Diffie-Helmann (cont.)

- Alice e Bob concordano su un primo **p** e un generatore **g** (**p** e **g** possono essere noti a Trudy)
- Alice sceglie un numero **x** a caso; invia a Bob $g^x \bmod p$
- Bob sceglie un numero **y** a caso; invia a Alice $g^y \bmod p$
- Alice calcola $(g^y \bmod p)^x \bmod p = g^{yx} \bmod p$
- Bob calcola $(g^x \bmod p)^y \bmod p = g^{yx} \bmod p$

Metodo di Diffie-Helmann (cont.)

Perché funziona:

- Se g è un generatore allora

  $(g^y \bmod p)^x \bmod p = g^{yx} \bmod p$

  $= (g^x \bmod p)^y \bmod p$

- Trudy conosce **p,g, $g^x \bmod p$, $g^y \bmod p$** ma per conoscere x deve calcolare il logaritmo discreto di **$g^x \bmod p$**

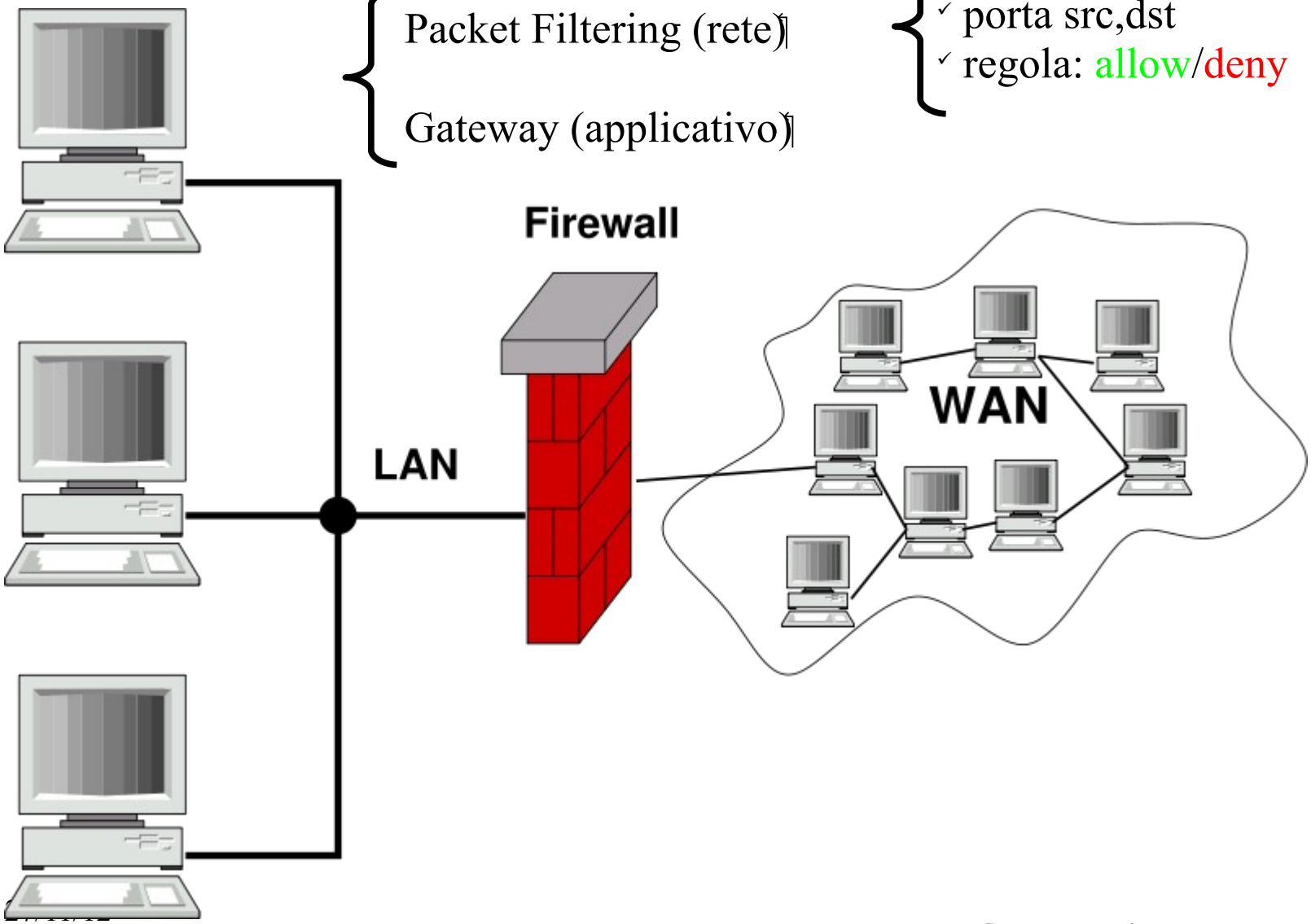
Logaritmo discreto:

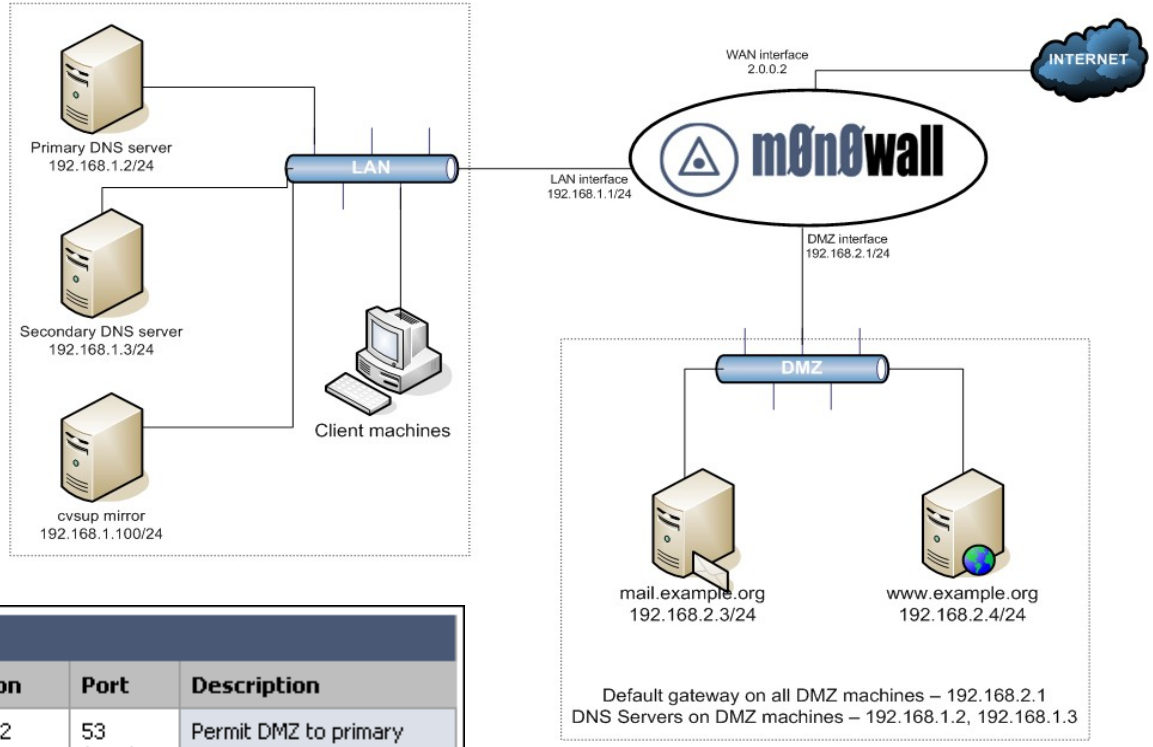
- dati a,p,g calcolare x t.c. **$g^x \bmod p = a$**
- Log. discreto è un problema computazionalmente difficile

Firewall

- Packet Filtering (rete)
- Gateway (applicativo)

- ✓ IP src,dst
- ✓ porta src,dst
- ✓ regola: allow/deny





DMZ interface

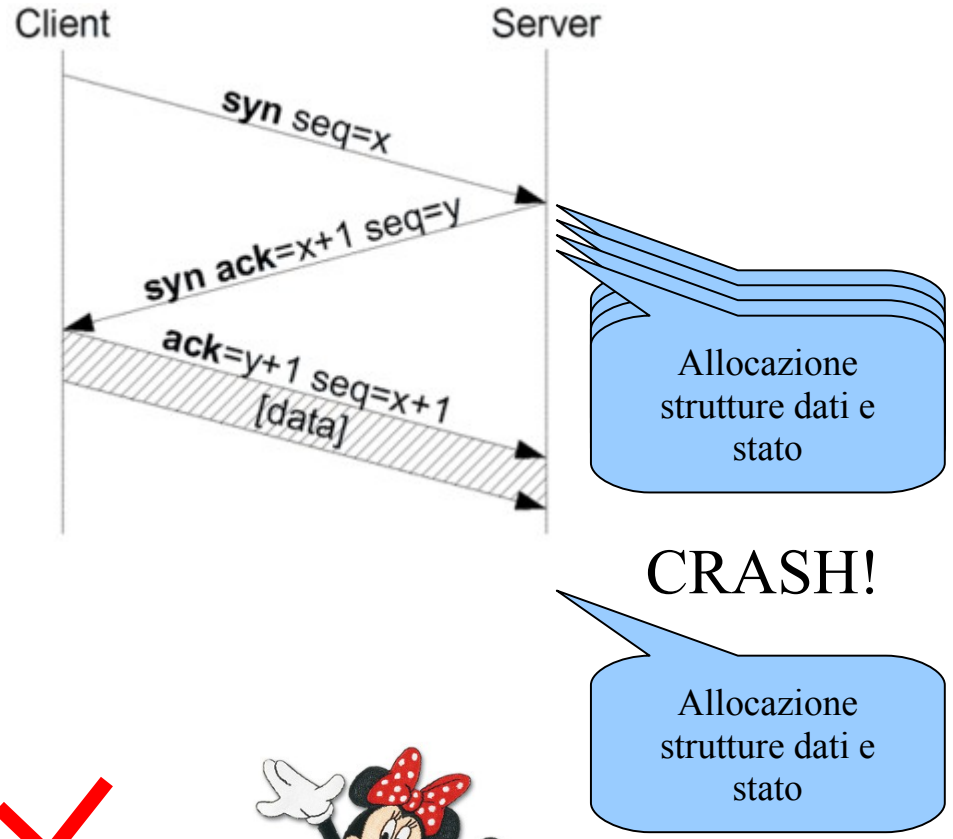
	Proto	Source	Port	Destination	Port	Description
↑	UDP	DMZ net	*	192.168.1.2	53 (DNS)	Permit DMZ to primary DNS server
↑	UDP	DMZ net	*	192.168.1.3	53 (DNS)	Permit DMZ to secondary DNS server
↑	TCP	DMZ net	*	192.168.1.100	5999	permit DMZ to cvsup on cvsup mirror server
↑	UDP	DMZ net	*	192.168.1.100	123	permit DMZ to NTP on cvsup mirror server
✗	*	*	*	LAN net	*	Reject DMZ traffic to LAN
↑	*	DMZ net	*	! LAN net	*	permit DMZ to any *BUT* LAN

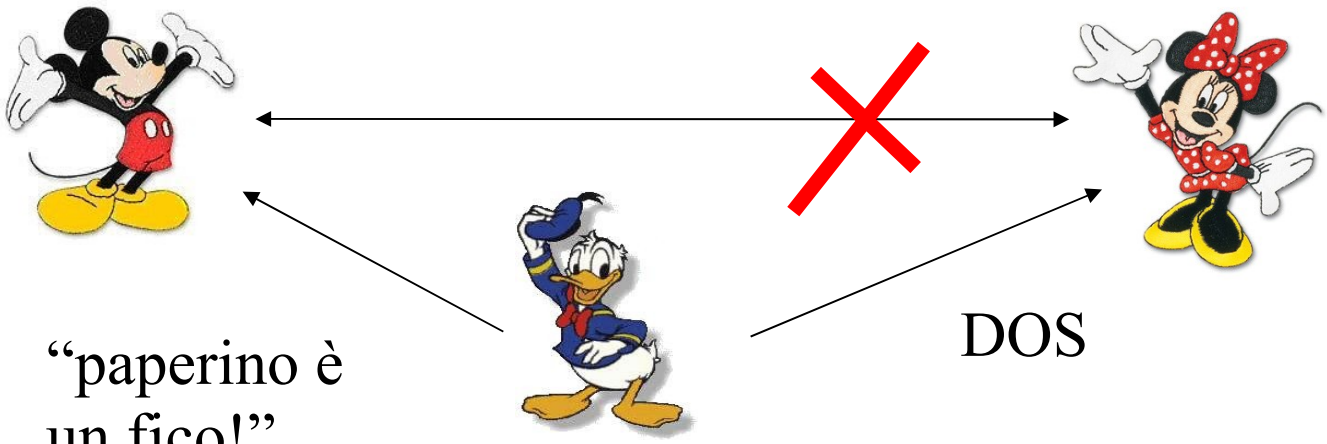
Attacchi

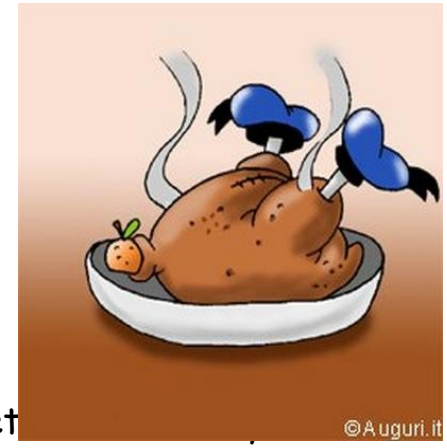
- Mapping
  - colleziono informazioni. es: ping
- Sniffing
  - Modalità promiscua
  - Wireshark
  - catturo informazioni rilevanti. es: password
- Spoofing
  - cambio IP src. es: Software Open Source
  - IP appartiene alle interfacce raggiungibili? (filtraggio)

Attacchi

- Denial of Service (DOS)
  - SYN flooding
  - Distributed DOS
- Dirottamento (hijack)



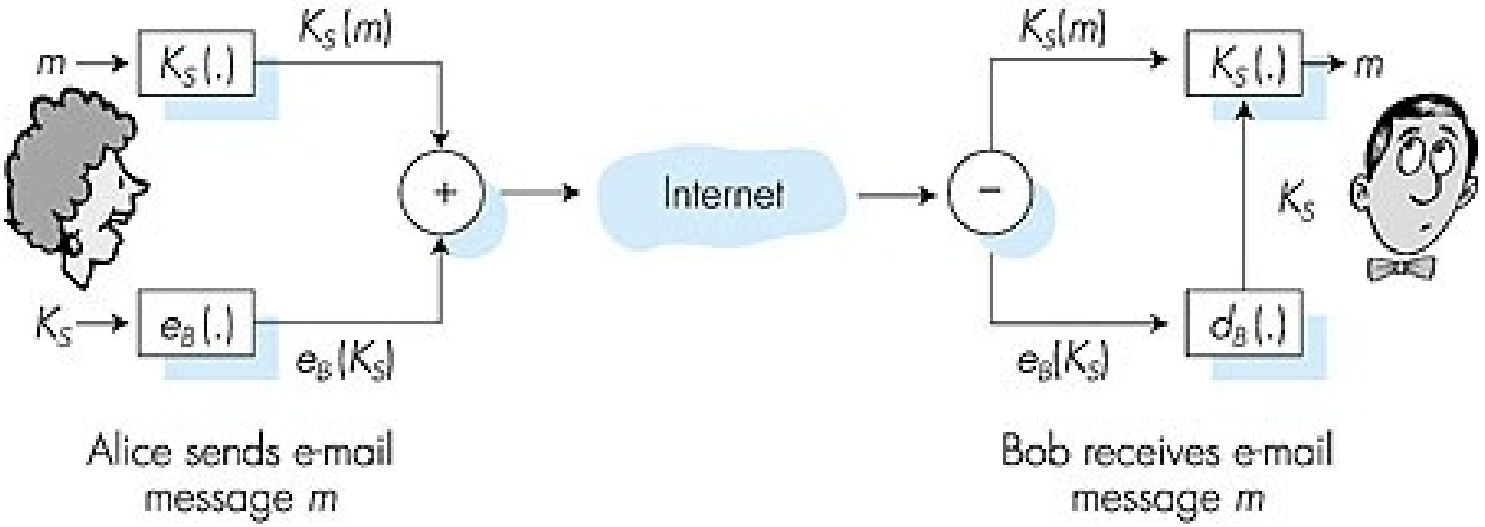




La sicurezza ai diversi livelli

e-mail sicura

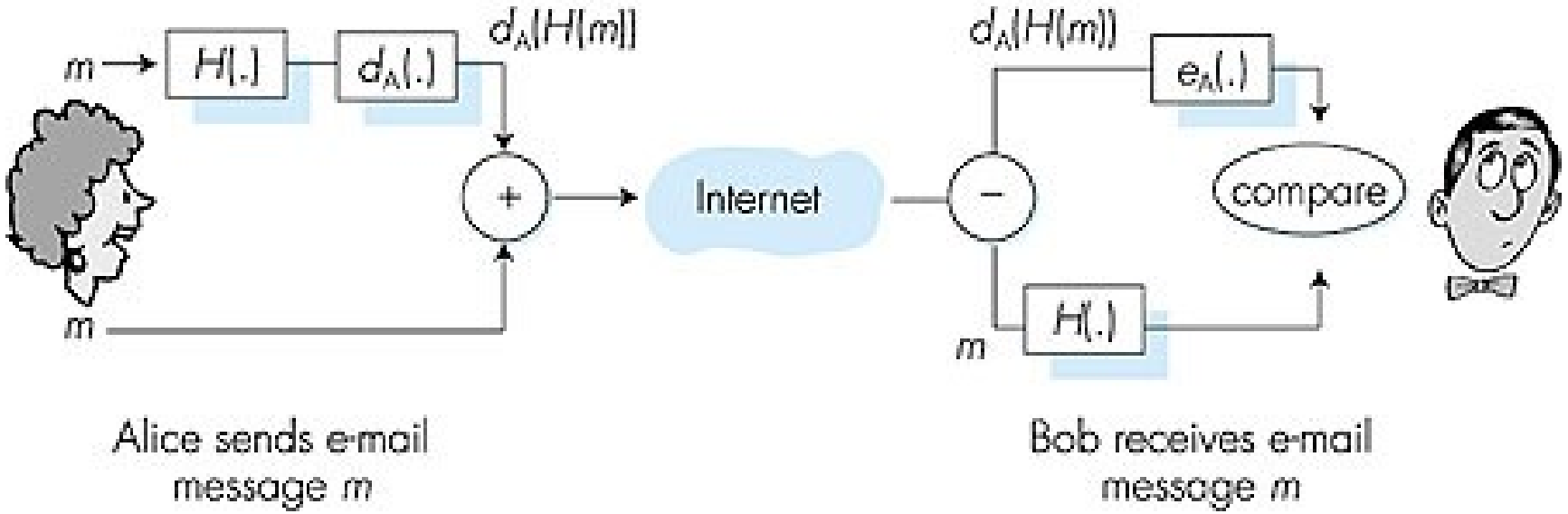
Alice invia messaggio m e-mail segreto a Bob:



- genera chiave simmetrica casuale K_S
- codifica messaggio con K_S
- codifica K_S con la chiave pubblica di Bob
- invia $K_S(m)$ e $e_B(K_S)$ a Bob.

e-mail sicura (cont.)

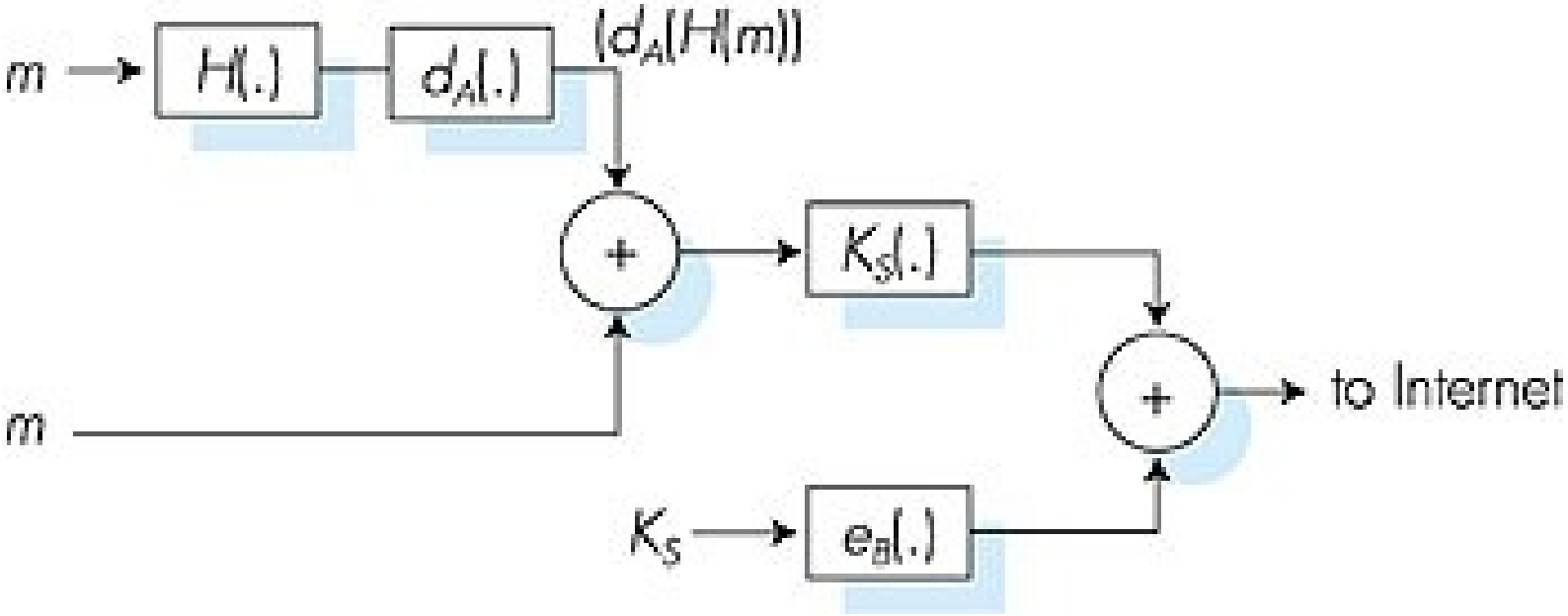
Alice vuole fornire autenticazione da parte del mittente dell'integrità del messaggio



- Alice appone la sua firma digitale al messaggio
- invia il messaggio (in chiaro) e la firma digitale

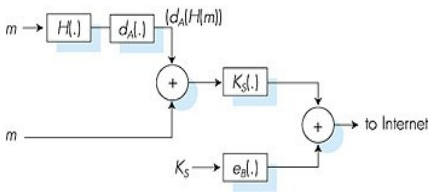
e-mail sicura (cont.)

Alice vuole fornire confidenzialità, integrità e autentica da parte del mittente



Alice usa sia la sua chiave privata che la chiave pubblica di Bob

e-mail sicura: Pretty good privacy (PGP)



- Schema di codifica per e-mail, standard de-facto in Internet
- Usa crittografia a chiave simmetrica e a chiave pubblica, calcolo impronta e firma digitale
- Fornisce confidenzialità, autentica del mittente e integrità dei messaggi
- Inventore: Phil Zimmerman, sotto inchiesta per 3 anni per esportazione illegale di materiale bellico

Un messaggio PGP firmato:

```
---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+lo8gE4vB3
   mqJhFEvZP9t6n7G6m5Gw2
---END PGP SIGNATURE---
```

MIME

Secure Socket Layer (SSL)

- PGP fornisce sicurezza per una specifica applicazione
- SSL opera sullo strato di trasporto; fornisce sicurezza ad ogni applicazione TCP
- SSL: usato fra WWW browsers, servers per e-commerce (https).
- SSL servizi offerti:
  - Autenticazione server
  - Codifica dati
  - Autenticazione client (opzionale)

Autenticazione Server:

- browser con SSL include chiavi pubbliche per CA fidate
- Browser richiede certificato del server
- Browser usa la chiave pubblica della CA per estrarre e verificare la chiave pubblica del server dal certificato

Secure Socket Layer (cont.)

Sessione SSL crittata:

- Browser genera chiave simmetrica di sessione, la codifica con la chiave pubblica del server e invia la codifica al server
- Server decodifica la chiave di sessione con la sua chiave pubblica
- Browser e server concordano che messaggi futuri saranno crittati
- I dati inviati nel socket TCP sono codificati con la chiave di sessione

- SSL: base del livello di trasporto sicuro (Transport Layer Security, TLS).
- SSL può essere usato anche per altre applicazioni (non Web) ad es., IMAP.
- L'autentica del cliente può essere fatta in modo analogo (usando certificati del cliente)
- Uso di chiavi di sessione generate ad hoc permette di limitare l'uso della chiave pubblica (+ veloce, + sicuro)

Secure electronic transact. (SET)

Limiti di SSL in applicazioni e-commerce. Garanzie su

- Negozio: La carta di credito è valida(rubata?)
- Cliente: L'azienda è autorizzata alla transazione?

SET

- Progettato per transazioni con carta di credito
- Fornisce sicurezza fra:
  - cliente
  - negoziante
  - banca negoziante
- Tutti hanno certificati
- Il numero della carta di credito è fornito in modo crittato (negoz. non vede il numero in chiaro)
  - Previene frodi da parte del negoziante
- Tre componenti software :
  - Browser
  - Server negoziante
  - Gateway cliente

Ipsec: Sicurezza allo strato di rete

- Sicurezza allo strato di rete:
  - Mittente invia i dati crittati in datagramma IP
  - Segmenti TCP e UDP; messaggi ICMP e SNMP
- Autenticazione:
  - Host destinazione può autenticare indirizzo IP sorgente
- Due protocolli principali:
  - Autenticazione dell'header + integrità messaggio (protocollo AH)
  - Confidenzialità dei dati (protocollo ESP)
- In AH e ESP, mittente e destin. eseguono handshake:
  - Creano un canale logico a livello di rete (Security Association, SA)
  - Determinano chiave segreta (Diffie-Hellman)
- Ogni SA è unidirezionale
- Unicamente determinata da:
  - Protocollo di sicurezza (AH or ESP)
  - Indirizzo Ip sorgente
  - ID connessione (32-bit)

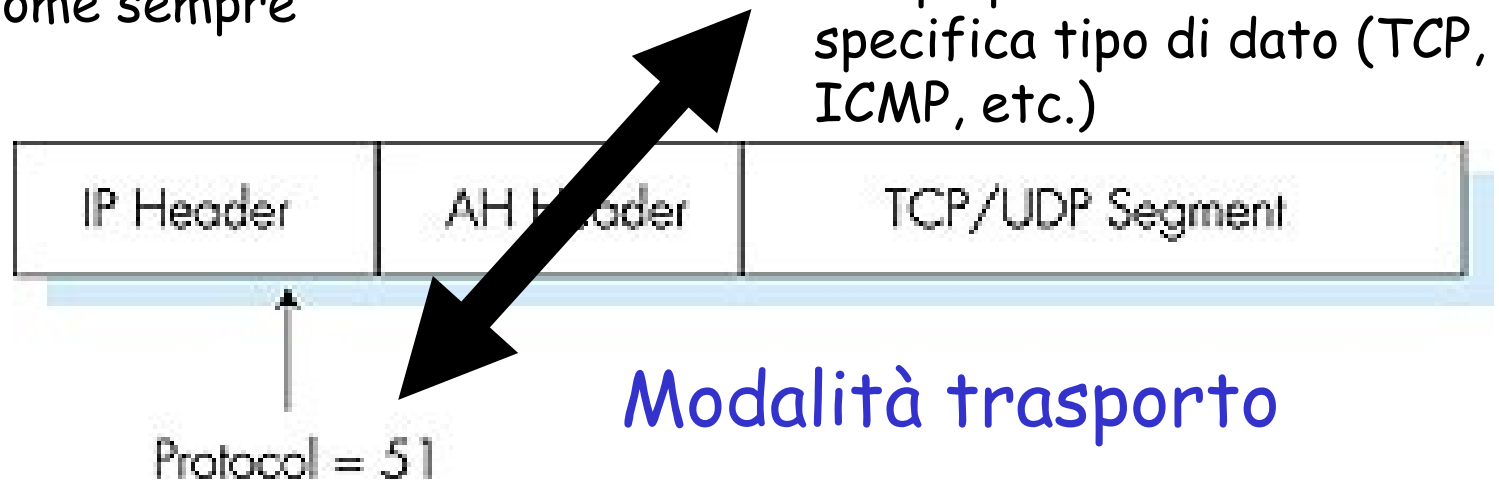
Due modalità: Trasporto e Tunnel

Protocollo Authentication Header (AH)

- Fornisce autenticazione sorgente, integrità dei dati, non segretezza
- Header AH inserito fra header IP e campo dati IP
- Protocol field = 51.
- Router Intermedi processano i datagrammi come sempre

Header AH include:

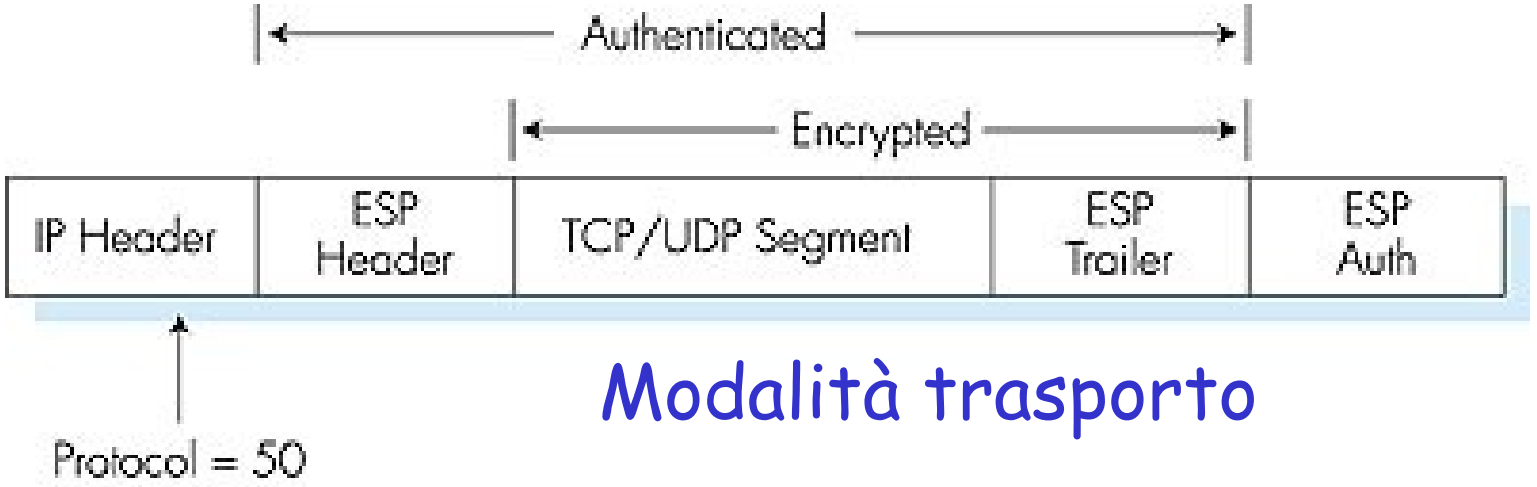
- Identificatore di connessione
- Autenticazione dei dati: firma impronta messaggio, calcolata su datagramma IP originale- parte non modificabile (fornisce autenticazione mittente e integrità dei dati)
- Campo prossimo header: specifica tipo di dato (TCP, UDP, ICMP, etc.)



Modalità trasporto

Protocollo ESP

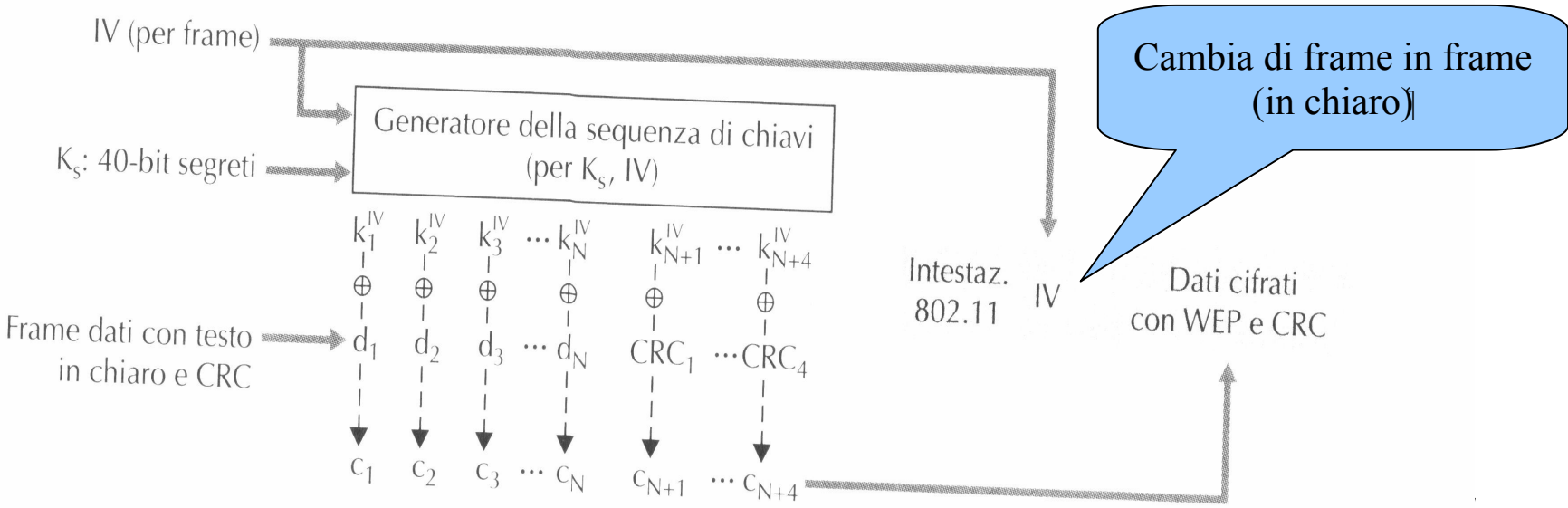
- Fornisce segretezza, autenticazione host, integrità dei dati
- Dati , ESP trailer encrypted.
- Next header field is in ESP trailer.
- Autenticazione ESP simile a autenticazione AH
- Campo protocollo = 50.



Modalità trasporto

WEP Wireless

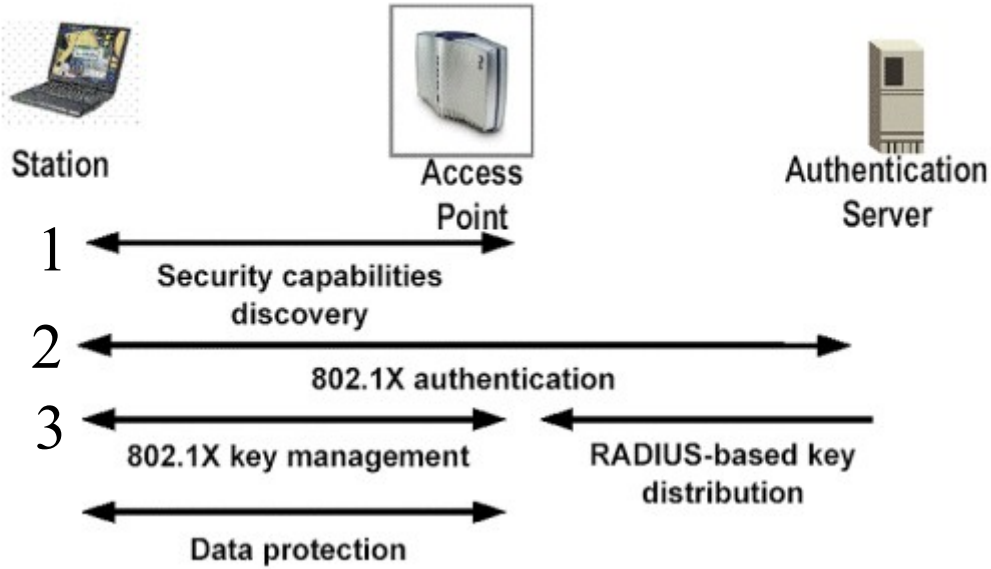
- Chiave simmetrica (40 bit) condivisa
- Autenticazione con nounce
- IV variabile 24 bit
- Fissata la Ks, ci sono solo 2^24 chiavi univoche



WEP

- Fissata la Ks, ci sono solo 2^24 chiavi univoche
- Chiavi a caso → prob riusare chiave (setsso IV) > 99% dopo 12000 frame
- Paccketto 1K, freq 11Mbps → pochi secondi
- Poichè IV in chiaro, facile identificare duplicati
- Noti di e Ci → facile trovare Ki
- Noto IV e Ki → decifrare è semplice
- 802.11i

802.11i



- AP comunica capabilities e Station seleziona (ancora no autenticazione)
- STA e AS si autenticano e generano MK (Master Key), AP si comporta da "punto di passaggio"
- MK → PMK (Pairwise Master Key)
- AS comunica PMK ad AP
- PMK → session Key

La controversia crittografica

Come i governi devono bilanciare interessi in conflitto?

- Privatezza dei cittadini: la crittografia è lo strumento che garantisce i cittadini da intrusioni
- Rispetto della legge: la crittografia è usata da criminali e terroristi

Semplificando

- In God we trust, all others we monitor

(motto (?) della National Security Agency, USA)

- Se la crittografia è fuori legge solo i fuorilegge useranno la crittografia (Phil Zimmerman)

Esercizi

1. Descrivere graficamente un protocollo di autenticazione basato su chiave pubblica che risulta vulnerabile rispetto ad un attacco del tipo "uomo nel mezzo".

   Il protocollo è più sicuro se l'autenticazione è richiesta da entrambi le parti?

   Discutere brevemente inoltre le modalita' con cui è possibile rendere robusto il protocollo rispetto a questo tipo di attacchi.

Esercizi

2. Descrivere graficamente un metodo per il calcolo dell'impronta di un documento che utilizza DES (e calcola impronte di 64 bit). Infine spiegare in dettaglio perche' impronte di 64 bit non sono considerate sicure (indipendentemente dal metodo utilizzato per il calcolo).

Esercizi

3. Assumete che un KDC server o un CA server si guasti. Chi può comunicare in modo sicuro e chi no nei due casi?