Università di Genova
Facoltà di Ingegneria

12. Sicurezza nelle reti di telecomunicazioni

Prof. Raffaele Bolla

Sicurezza nelle reti

- Ci sono tre aree in cui bisogna intervenire per rendere una rete sicura
  - **Riservatezza**: il messaggio deve essere accessibile (visualizzabile o rilevabile la sua presenza) sola ad entità autorizzate.
  - **Autenticazione**: L'identità delle entità coinvolte nella comunicazioni deve poter essere verificata.
  - **Integrità** (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).

6.2

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.

6.3

Sicurezza nelle reti
Attacchi al contenuto - Esempio di sniffing

1	0.000000	abete.reti.dist.unige	wpop10.libero.it	TCP	32811 > http [SYN] Seq=2175924618 Ack=0 Win=5840 Len=0
2	0.047915	wpop10.libero.it	abete.reti.dist.unige	TCP	http > 32811 [SYN, ACK] Seq=395931055 Ack=2175924619 Win=24616 Len=0
3	0.047968	abete.reti.dist.unige	wpop10.libero.it	TCP	32811 > http [ACK] Seq=2175924619 Ack=395931056 Win=5840 Len=0
4	0.048136	abete.reti.dist.unige	wpop10.libero.it	HTTP	POST /email.php HTTP/1.1
5	0.080256	wpop10.libero.it	abete.reti.dist.unige	TCP	http > 32811 [ACK] Seq=395931056 Ack=2175925157 Win=24616 Len=0
6	0.080281	abete.reti.dist.unige	wpop10.libero.it	HTTP	Continuation
7	0.223209	wpop10.libero.it	abete.reti.dist.unige	TCP	http > 32811 [ACK] Seq=395931056 Ack=2175925329 Win=24616 Len=0
8	5.503723	wpop10.libero.it	abete.reti.dist.unige	HTTP	HTTP/1.1 200 OK
9	5.503763	abete.reti.dist.unige	wpop10.libero.it	TCP	32811 > http [ACK] Seq=2175925329 Ack=395931407 Win=6432 Len=0
10	5.575033	abete.reti.dist.unige	wpop10.libero.it	HTTP	GET /error.html HTTP/1.1
11	5.595797	wpop10.libero.it	abete.reti.dist.unige	HTTP	HTTP/1.1 200 OK
12	5.595843	abete.reti.dist.unige	wpop10.libero.it	TCP	32811 > http [ACK] Seq=2175925881 Ack=395932855 Win=8688 Len=0
13	5.598326	wpop10.libero.it	abete.reti.dist.unige	HTTP	Continuation
14	5.598367	abete.reti.dist.unige	wpop10.libero.it	TCP	32811 > http [ACK] Seq=2175925881 Ack=395934303 Win=11584 Len=0
15	5.598693	wpop10.libero.it	abete.reti.dist.unige	HTTP	Continuation
16	5.598705	abete.reti.dist.unige	wpop10.libero.it	TCP	32811 > http [ACK] Seq=2175925881 Ack=395934857 Win=14480 Len=0
17	5.638893	abete.reti.dist.unige	wpop10.libero.it	HTTP	GET /old/xam_rc/template_xam/images/spacer.gif HTTP/1.1
18	5.643669	abete.reti.dist.unige	wpop10.libero.it	TCP	32812 > http [SYN] Seq=2171723094 Ack=0 Win=5840 Len=0
19	5.667858	wpop10.libero.it	abete.reti.dist.unige	HTTP	HTTP/1.1 200 OK
20	5.667898	abete.reti.dist.unige	wpop10.libero.it	TCP	32811 > http [ACK] Seq=2175926465 Ack=395935211 Win=17376 Len=0
21	5.668389	abete.reti.dist.unige	wpop10.libero.it	HTTP	GET /old/xam_rc/template_xam/images/button_err_back.gif HTTP/1.1
22	5.677580	wpop10.libero.it	abete.reti.dist.unige	TCP	http > 32812 [SYN, ACK] Seq=1219537146 Ack=2171723095 Win=24616 Len=0
23	5.677615	abete.reti.dist.unige	wpop10.libero.it	TCP	32812 > http [ACK] Seq=2171723095 Ack=1219537147 Win=5840 Len=0
24	5.677745	abete.reti.dist.unige	wpop10.libero.it	HTTP	GET /old/xam_rc/template_xam/images/header_err.gif HTTP/1.1
25	5.699033	wpop10.libero.it	abete.reti.dist.unige	HTTP	HTTP/1.1 200 OK
26	5.707494	wpop10.libero.it	abete.reti.dist.unige	TCP	http > 32812 [ACK] Seq=1219537147 Ack=2171723683 Win=24028 Len=0
27	5.741745	abete.reti.dist.unige	wpop10.libero.it	TCP	32811 > http [ACK] Seq=2175927058 Ack=395935708 Win=17376 Len=0
28	9.096682	wpop10.libero.it	abete.reti.dist.unige	HTTP	HTTP/1.1 200 OK
29	9.096723	abete.reti.dist.unige	wpop10.libero.it	TCP	32812 > http [ACK] Seq=2171723683 Ack=1219538216 Win=7483 Len=0

6.4

Sicurezza nelle reti
Attacchi al contenuto - Esemio di sniffing

```
Frame 6 (238 on wire, 238 captured)
    Arrival Time: Dec 10, 2002 13:39:46.422800000
    Time delta from previous packet: 0.000025000 seconds
    Time relative to first packet: 0.080281000 seconds
    Frame Number: 6
    Packet Length: 238 bytes
    Capture Length: 238 bytes
Ethernet II
    Destination: 00:00:0c:03:de:0a (Cisco_03:de:0a)
    Source: 00:e0:18:a0:36:cc (Asustek_a0:36:cc)
    Type: IP (0x0800)
Internet Protocol, Src Addr: abete.reti.dist.unige.it (130.251.8.11), Dst Addr: wpop10.libero.it (193.70.192.46)
    Version: 4
    Header length: 20 bytes
  Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 224
    Identification: 0x2efe
  Flags: 0x04
    Fragment offset: 0
    Time to live: 64
    Protocol: TCP (0x06)
    Header checksum: 0xfe9e (correct)
    Source: abete.reti.dist.unige.it (130.251.8.11)
    Destination: wpop10.libero.it (193.70.192.46)
Transmission Control Protocol, Src Port: 32811 (32811), Dst Port: http (80), Seq: 2175925157, Ack: 395931056
    Source port: 32811 (32811)
    Destination port: http (80)
    Sequence number: 2175925157
    Next sequence number: 2175925329
    Acknowledgement number: 395931056
    Header length: 32 bytes
  Flags: 0x0018 (PSH, ACK)
    Window size: 5840
    Checksum: 0x24ab (correct)
  Options: (12 bytes)
Hypertext Transfer Protocol
    Content-Type: application/x-www-form-urlencoded\r\n
    Content-Length: 100\r\n
    \r\n
    Data (100 bytes)
```

6.5

Sicurezza nelle reti
Attacchi al contenuto - Esempio di sniffing

```
0000  00 00 0c 03 de 0a 00 e0  18 a0 36 cc 08 00 45 00   ....Þ..à . 6Ì..E.
0010  00 e0 2e fe 40 00 40 06  fe 9e 82 fb 08 0b c1 46   .à.þ@.@. þ..û..ÁF
0020  c0 2e 80 2b 00 50 81 b1  fb a5 17 99 6d b0 80 18   À..+.P.± û¥..m°..
0030  16 d0 24 ab 00 00 01 01  08 0a 00 0e d7 79 22 87   .Ð$«.... ....×y".
0040  96 9d 43 6f 6e 74 65 6e  74 2d 54 79 70 65 3a 20   ..Conten t-Type:
0050  61 70 70 6c 69 63 61 74  69 6f 6e 2f 78 2d 77 77   applicat ion/x-ww
0060  77 2d 66 6f 72 6d 2d 75  72 6c 65 6e 63 6f 64 65   w-form-u rlencode
0070  64 0d 0a 43 6f 6e 74 65  6e 74 2d 4c 65 6e 67 74   d..Conte nt-Lengt
0080  68 3a 20 31 30 30 0d 0a  0d 0a 64 6f 6d 69 6e 69   h: 100.. ..domini
0090  6f 3d 6c 69 62 65 72 6f  2e 69 74 26 4c 4f 47 49   o=libero .it&LOGI
00a0  4e 3d 75 74 65 6e 74 65  26 50 41 53 53 57 44 3d   N=utente &PASSWD=
00b0  70 61 73 73 26 63 68 6f  69 63 65 3d 6c 69 62 65   pass&cho ice=libe
00c0  72 6f 26 41 63 74 5f 4c  6f 67 69 6e 2e 78 3d 35   ro&Act_L ogin.x=5
00d0  26 41 63 74 5f 4c 6f 67  69 6e 2e 79 3d 39 26 41   &Act_Log in.y=9&A
00e0  63 74 5f 4c 6f 67 69 6e  3d 45 6e 74 72 61        ct_Login =Entra
```

Login: **utente**

Password: **pass**

6.6

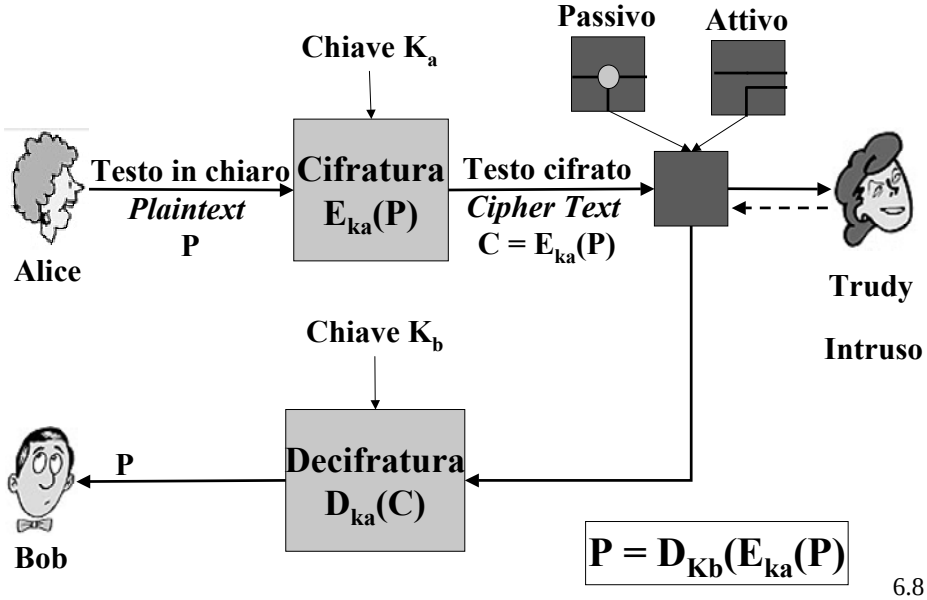
Sicurezza nelle reti
Attacchi

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

6.7

Riservatezza: Cifratura



6.8

Cifratura a chiave Simmetrica

- E' una tecnica antica (Giulio Cesare)
- $K_A = K_B = \mathbf{K}$: 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.

6.9

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.

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

6.10

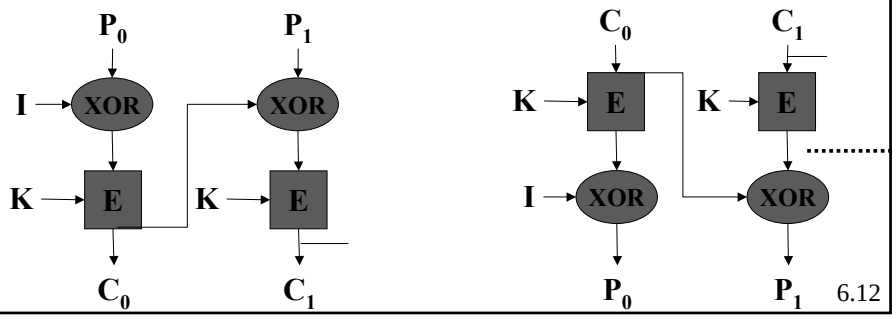
Cifratura a chiave simmetrica

- 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

6.11

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)



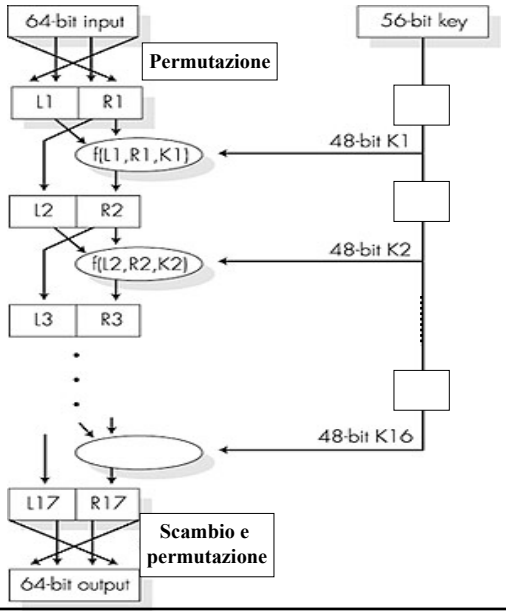
6.12

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,..,16
    - » Uno scambio dei 32 bit destri con i sinistri
    - » Una permutazione inversa alla prima

6.13

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

6.14

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 245×10^9 chiavi al sec.)
- Ad oggi, (nella sua forma con chiave a 56 bit) non è considerato molto sicuro.

6.15

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: $\mathbf{K_1}$, $\mathbf{K_2}$, $\mathbf{K_3}$.
- Opera come segue:

$$\mathbf{C = E_{K3}(D_{K2}(E_{K1}(P))}$$

- 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 $\mathbf{K_1 = K_3}$

6.16

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 e utilizzare ambedue i metodi.

6.17

Cifratura a chiave pubblica

- Utilizza due chiavi:
  - Una chiave $\mathbf{K_A}$ usata per la cifratura che viene resa pubblica (chiave pubblica).
  - Una chiave $\mathbf{K_B}$ 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
  - $\mathbf{D_{KB}(E_{KA}(P)) = P}$
  - Non deve essere possibile dedurre $\mathbf{K_B}$ da $\mathbf{K_A}$.
  - K_B non deve poter essere dedotta tramite cifratura di testi noti

6.18

Cifratura a chiave pubblica
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 ·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 ·d mod z = 1.**
- La chiave pubblica $\mathbf{K_A = (n, e)}$ e la chiave privata $\mathbf{K_B = (n, d)}$.

6.19

Cifratura a chiave pubblica
Rivest, Shamir, Adelson (RSA)

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

$$\mathbf{m = (m^e \bmod n)^d \bmod n}$$

6.20

Cifratura a chiave pubblica
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**).

	Lettera	m	m^e	$c = m^e \bmod n$
Cifra:	l	12	248832	17

	c	c^d	$m = c^d \bmod n$	Lettera
Decifra:	17	481968572106750915091411825223072000	12	l

6.21

Cifratura a chiave pubblica
Rivest, Shamir, Adelson (RSA)

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

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

- $\mathbf{(m^e \bmod n)^d \bmod n = m^{ed} \bmod n =}$

$$\mathbf{= m^{ed \bmod (p-1)(q-1)} \bmod n =}$$

(grazie al risultato della teoria dei numeri di cui sopra)

$$\mathbf{= m^1 \bmod n =}$$

(dato che si è scelto **ed** divisibile per **(p-1)(q-1)** con resto 1)

$$\mathbf{= m}$$

6.22

Cifratura a chiave pubblica
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.

6.23

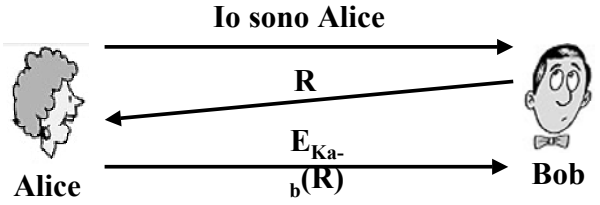
Autenticazione

- Obiettivo: Bob vuole che Alice provi la sua identità
- Una prima serie di Autentication Protocol (AP) semplici potrebbero essere:
  - **AP1:** Alice invia un pacchetto dicendo "Sono Alice".
  - **AP2:** Alice invia un pacchetto dicendo "Sono Alice" ed allega il suo numero IP.
  - **AP3:** Alice invia un pacchetto dicendo "Sono Alice" ed allega una password.
  - **AP3.1:** Alice invia un pacchetto dicendo "Sono Alice" ed allega una password cifrata.

6.24

Autenticazione

- **AP4** Per evitare un "*playback attack*", si usa un *nonce* (*only once*) R, ossia un numero usato una volta sola generato a caso da Bob).

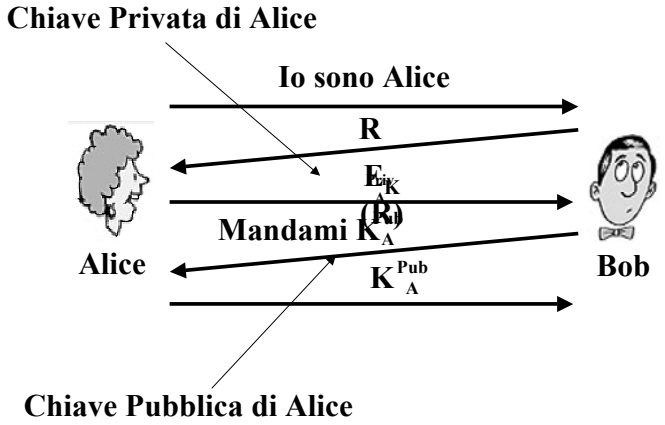


Come si scambiano la chiave K_{a-b} ?

6.25

Autenticazione

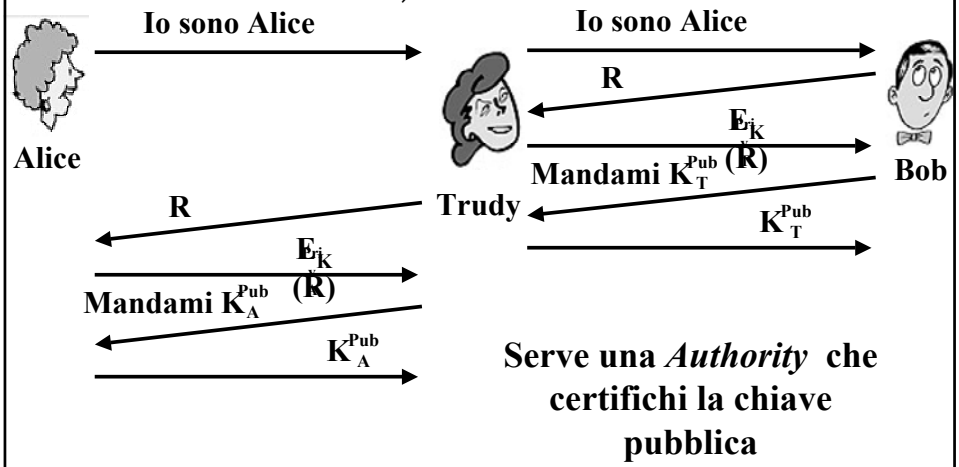
- **AP5**: l'uso della chiave pubblica:



6.26

Autenticazione

- AP5 è soggetto all'attacco nel mezzo (*in the middle attack*)



6.27

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).

6.28

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

6.29

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** .

6.30

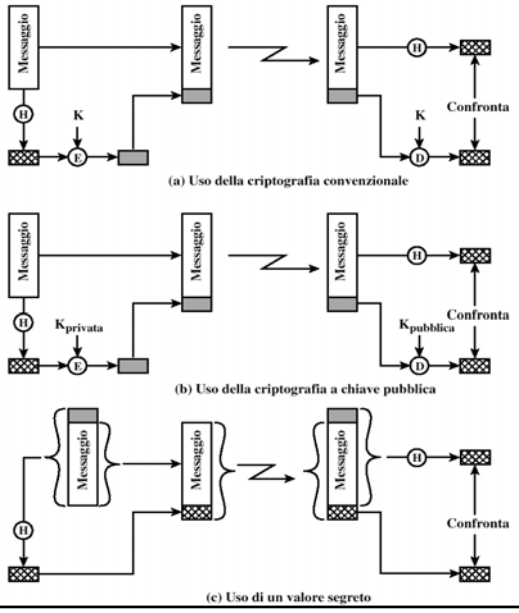
Integrità e Firma elettronica
Digest

- 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 ≠ x tale che H(y) = H(x) (impedisce falsificazioni).
  - Deve essere computazionalmente impossibile trovare una qualsiasi coppia (x, y) tale H(x) = H(y).

6.31

Integrità e Firma elettronica
Digest

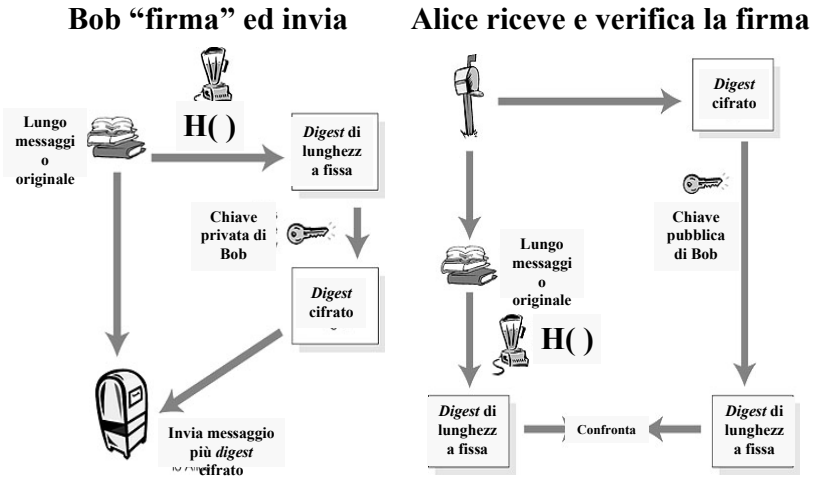
Possibili usi del *digest* per la verifica dell'integrità



6.32

Integrità e Firma elettronica
Digest

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



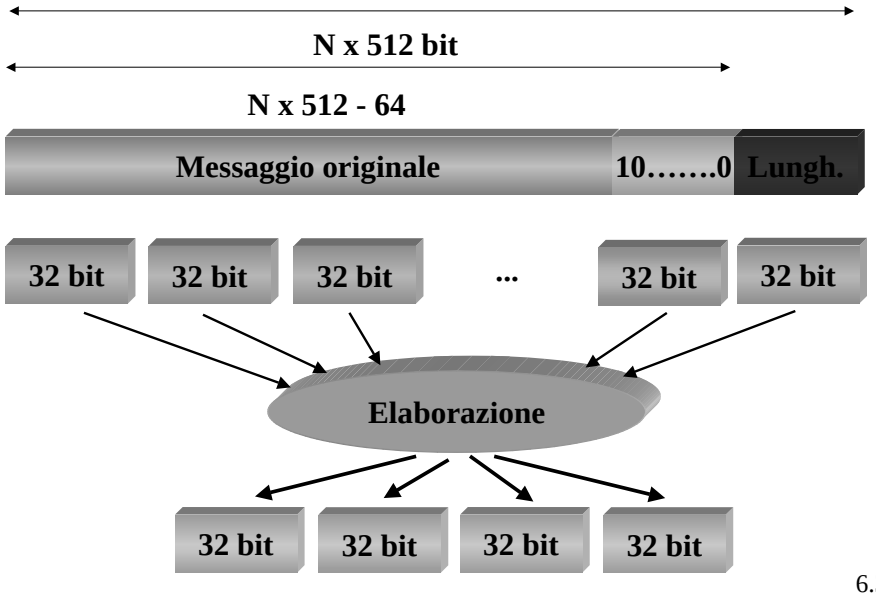
6.33

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.

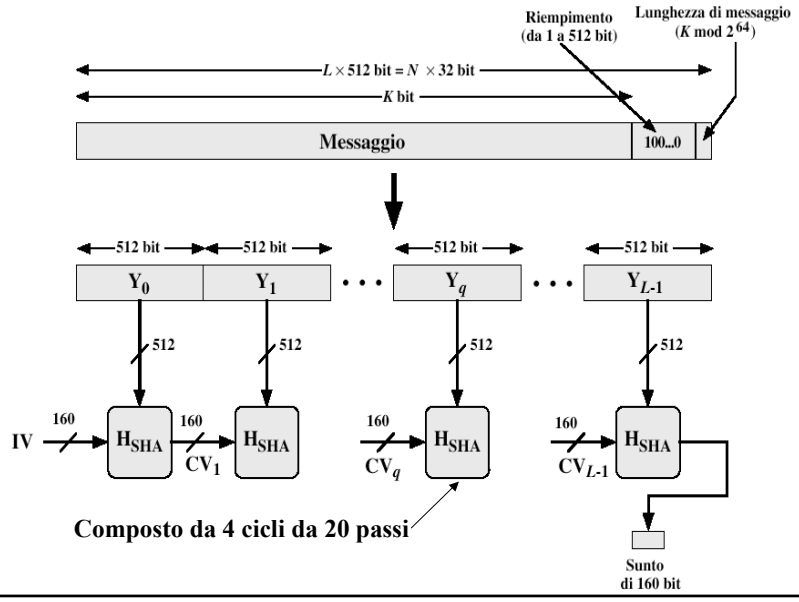
6.34

Integrità e Firma elettronica
Digest - MD5



6.35

Integrità e Firma elettronica
Digest - SHA-1



6.36

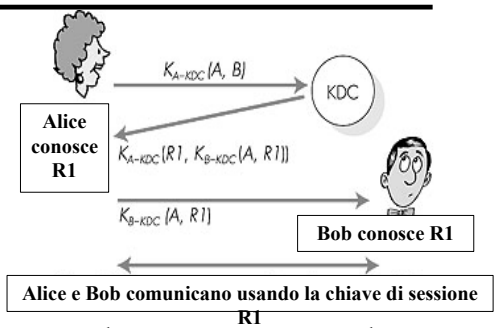
Distribuzione delle chiavi e certificazione

- 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.

6.37

Distribuzione delle chiavi e certificazione
Key Distibution 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, $K_{A\text{-}KDC}$ $K_{B\text{-}KDC}$, per comunicare con il KDC.

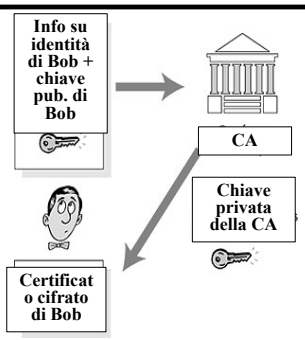


- Alice comunica con il KDC, acquisisce la chiave di sessione R1, e $K_{B\text{-}KDC}(A,R1)$
- Alice invia a Bob $K_{B\text{-}KDC}(A,R1)$ e Bob estrae R1
- Alice e Bob ora condividono la chiave simmetrica R1.

6.38

Distribuzione delle chiavi e certificazione
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.

6.39

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.

6.40

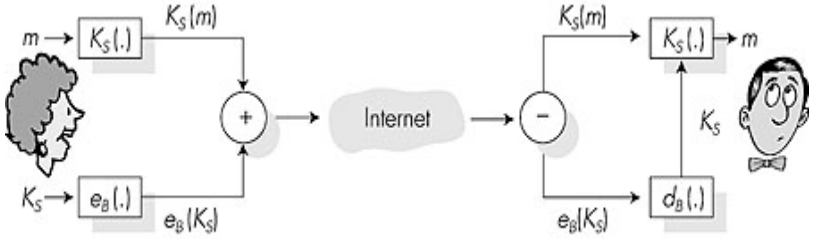
Sicurezza - Protocolli

- 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)

6.41

E-mail sicure - Segretezza dei dati

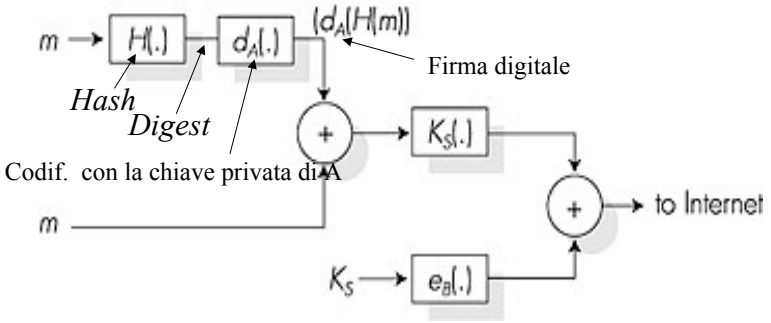
- Alice vuole inviare un messaggio m segreto a Bob



- 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

6.42

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)

6.43

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+lo8gE4vB
   3mqJhFEvZP9t6n7G6m5Gw2
---END PGP SIGNATURE---
```

6.44

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

6.45

Secure Socket Layer (SSL)

Autenticazione del server

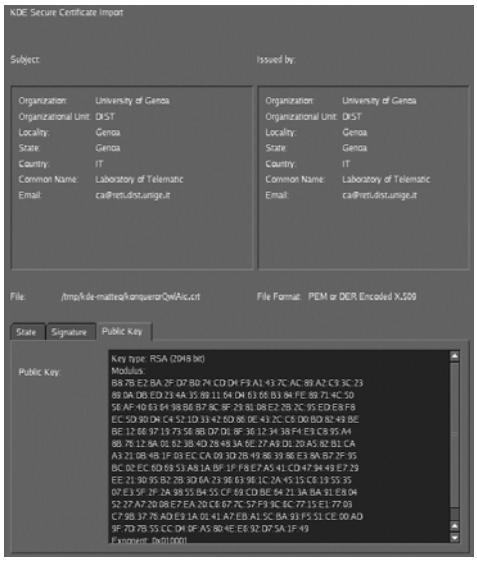
- Un *browser* con SSL deve possedere la chiave pubblica di una o più CA.
- Il *browser* richiede il certificato del Server secondo uno dei CA che conosce.
- Il *browser* usa la chiave pubblica del CA per estrarre la chiave pubblica del Server.

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

6.46

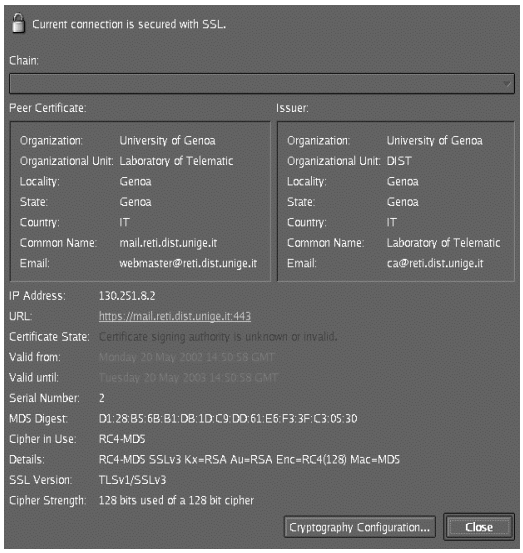
Distribuzione delle chiavi e certificazione
Certificati



Certificato di una CA autofirmato

6.47

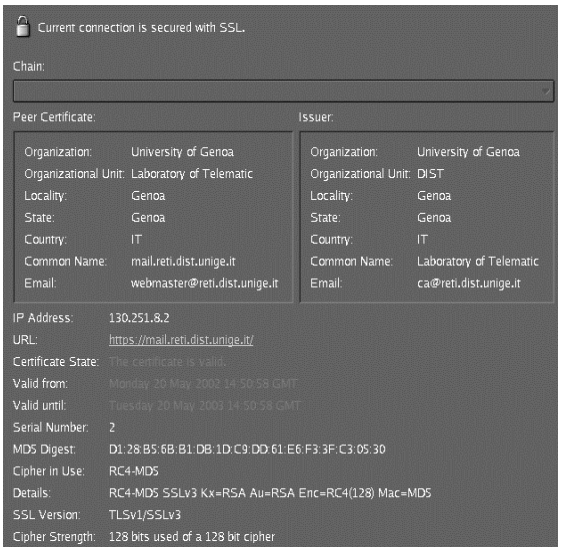
Distribuzione delle chiavi e certificazione
Certificati



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

6.48

Distribuzione delle chiavi e certificazione
Certificati



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

6.49

Secure Electronic Transaction (SET)

- E' stato progettato per realizzare i pagamenti via carta di credito e le transazioni via Internet (VISA e Mastercard);
- Prevede la presenza di tre attori che devono essere tutti certificati:
  - Cliente (certificato dalla propria banca)
  - Venditore (certificato dalla propria banca)
  - Banca (del venditore)
- Dà significato legale ai certificati;
- Il numero di carta di credito del cliente passa alla banca del venditore senza che quest'ultimo lo possa vedere.
- Tre componenti software:
  - *Browser wallet* (portafoglio)
  - *Merchant server*
  - *Acquirer Gateway*

6.50

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***

6.51

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.

6.52

Sicurezza a livello di rete

IPsec (IP security)

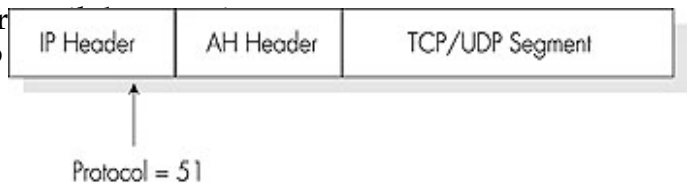
- Ambedue i protocolli di IPsec (ESP e AH) operano tramite una canale logico a livello di rete chiamato ***Security Association*** (**SA**), creato tra sorgente e destinazione con un *handshake*.
- L'SA è
  - Unidirezionale
  - Univocamente determinato da:
    - » Protocollo di sicurezza usato (ESP o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

6.53

Sicurezza a livello di rete

IPsec - AH

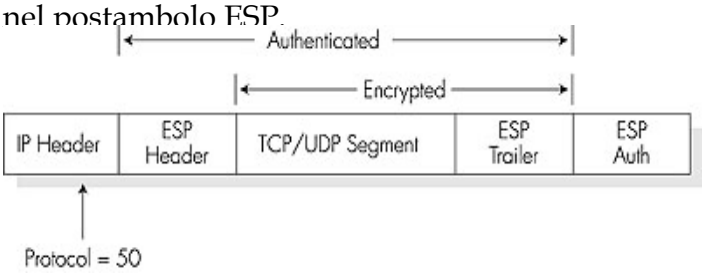
- Fornisce l'autenticazione dell'*host* e l'integrità dei dati ma non la riservatezza.
- L'intestazione AH viene inserita fra quella IP ed i dati
- Il numero di protocollo è il 51
- I *router* intermedi elabor… modo
- L'intestazione dell'AH comprende:
  - Un identificatore di connessione
  - Un *digest*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



6.54

Sicurezza a livello di rete
IPsec - ESP

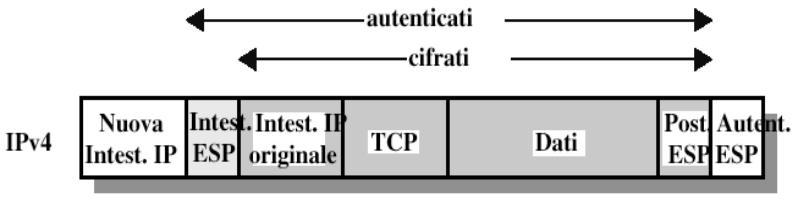
- Fornisce la riservatezza, l'autenticazione dell'host e l'integrità dei dati
- I dati e il postambolo dell'ESP sono cifrati
- L'indicazione della successiva intestazione è nel postambolo ESP.
- Il campo di autenticazione del ESP è simile ha quello dell'AH
- Il numero di protocollo contenuto nell'intestazione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



6.55

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.



6.56

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.

6.57