Università di Ferrara
Corso di laurea specialistica in Informatica

Security

Alberto Gianoli

Premessa

Tradizionalmente la sicurezza dei computer era legata a fattori pratici (pochi computer perchè costosi) e amministrativi (difficile accesso a un computer)

L'aumento degli utilizzatori ha richiesto la creazione di meccanismi per tenere separati gli utenti e i loro dati.

L'introduzione della rete e dei collegamenti tra computer richiede misure per proteggere i dati durante la trasmissione

Definizioni

Computer security: è il nome generico con cui ci si riferisce all'insieme di tools usati per proteggere i dati e tenere lontani gli hackers

Network security: sono i meccanismi usati per proteggere le informazioni quando vengono trasmesse da un computer ad un altro all'interno di una rete.

Internet security: estensione del problema precedente quando si ha a che fare con Internet.

Definizioni (2)

Security attack: qualsiasi azione che compromette la sicurezza dei dati (o del computer)

Security mechanism: un meccanismo ideato per rilevare, prevenire o eliminare un security attack. Non esiste un unico meccanismo che protegge da tutto. Però hanno quasi tutti in comune l'uso della crittografia.

Security service: un servizio che migliora la sicurezza di un sistema o della trasmissione di dati. I security services fanno uso di uno o più security mechanisms.

Security attacks

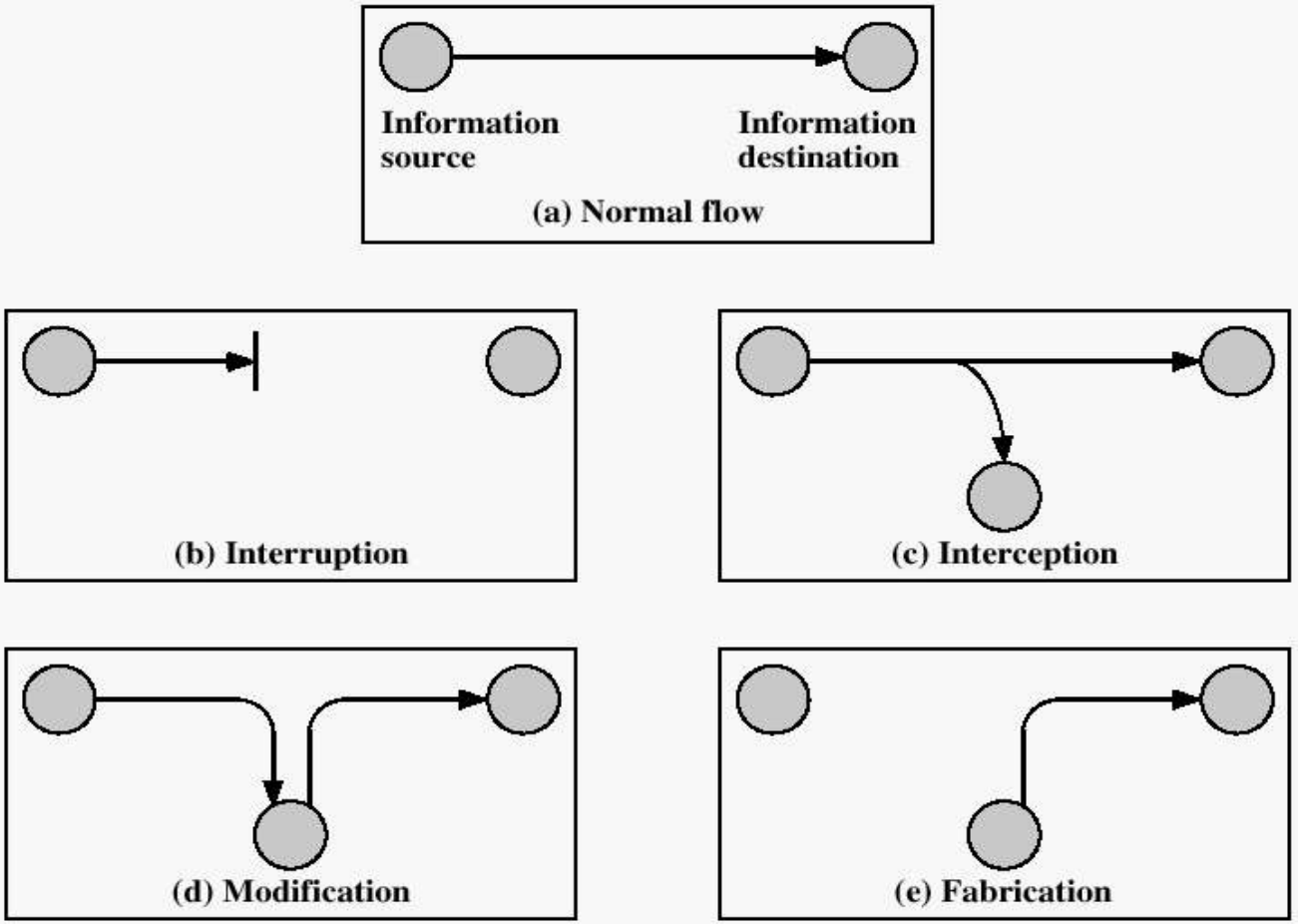


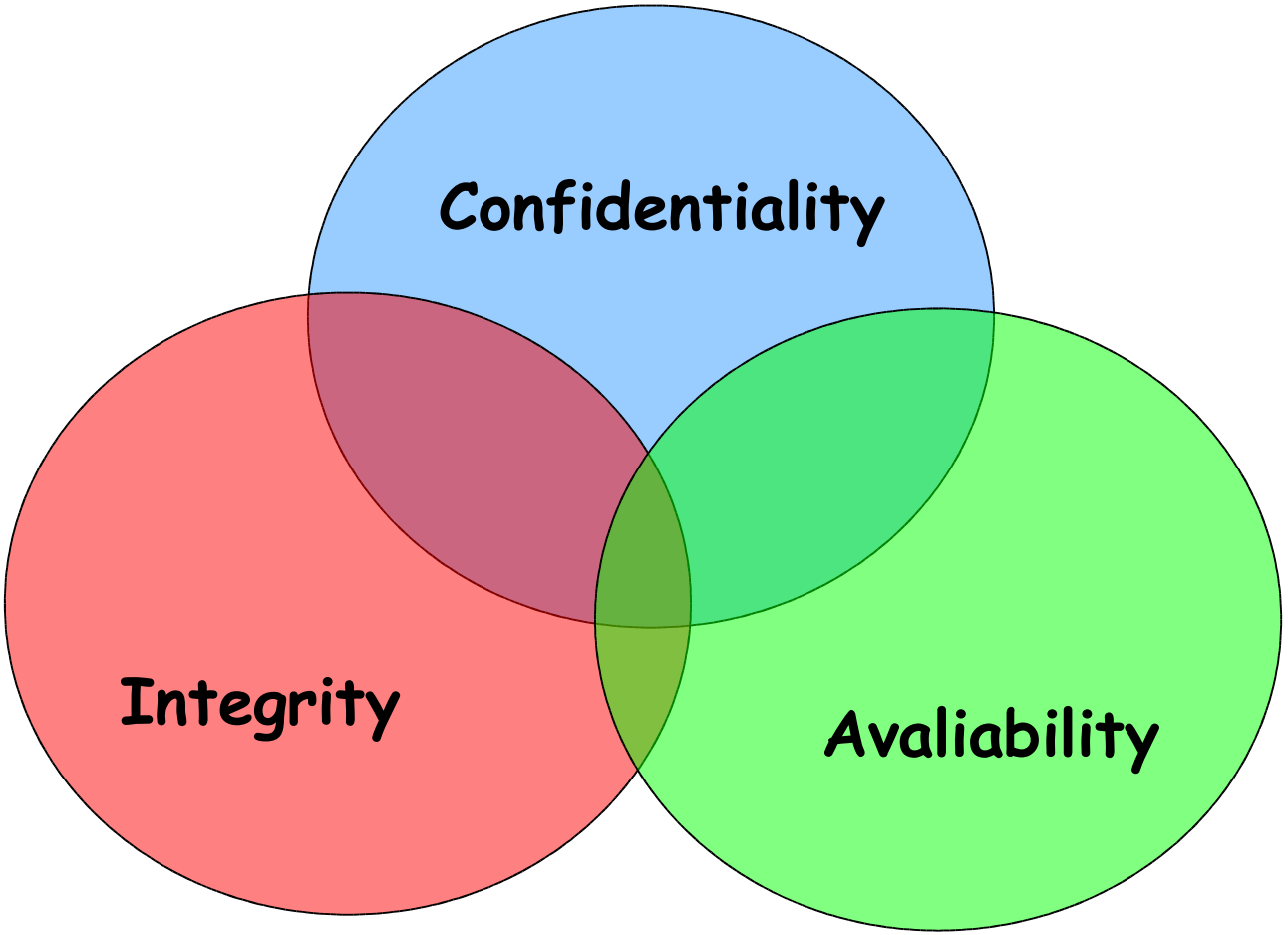
Figure 1.1 Security Threats

Security attacks (2)

- Interruzione: è un attacco che colpisce l'availability di un sistema
- Interception: colpisce la confidentiality
- Modification: altera l'integrità dei dati
- Fabrication: compromette l'autenticità di un messaggio

Il secondo attacco è considerato un attacco passivo (più difficile da scoprire), mentre gli altri sono classificati come attacchi attivi.

Security goals



Security services

Confidenzialità: privacy

Autenticazione: chi ha creato / mandato dei dati è veramente chi dice di essere

Integrità: i dati non sono stati manomessi

Non ripudiabilità: far finta di non aver mandato un email

Access control: prevenire l'uso improprio delle risorse

Availability: garantiscono l'accesso a un servizio / ai dati

Metodi di difesa 1

- Minimi privilegi: ogni oggetto (utente o programma) deve avere solo i minimi privilegi necessari al suo lavoro

- Difesa in profondità: non fare affidamento su un solo meccanismo di difesa, per forte che possa essere/sembrare

- Minimizzare i punti di accesso: dato che vanno controllati, ridurne il numero aiuta il lavoro

- Cercare l'anello debole: bisogna essere consci di qual'è il punto più debole del sistema, e se necessario rafforzarlo fino ad un livello accettabile

Metodi di difesa 2

- Fail-safe: quando si verifica un errore il sistema non deve permettere l'accesso a intrusi, anche a costo di rifiutare utenti legittimi

- Ciò che non è espressamente permesso è proibito:
  - Fail-safe
  - Di soito l'utente pensa l'opposto
  - Attivare i servizi sulla base delle reali necessità; gli altri sono disattivati

- Partecipazione universale: non è possibile avere una macchina sicura senza la partecipazione degli utenti

Metodi di difesa 3

- Criptare le informazioni
- Controlli software: metodi per ridurre l'accesso ai dati o ai servizi a seconda dell'utente. Si basano sulla conoscenza di una password.
- Controlli hardware: come sopra ma si basano su una pezzo hardware (es. smartcard)
- Policies: per esempio obbligare gli utenti a cambiare le password con una certa frequenza oppure imporre password opportunamente scelte.
- Controlli antropometrici: fa molto film di spionaggio, ma un portatile capace di riconoscere l'impronta digitale è già esistito.....

Crittografia

I metodi di crittografia sono caratterizzati da tre parametri indipendenti:

- Il tipo di operazione usata per passare dal plaintext al ciphertext

- Il numero di chiavi usate
  - Chiave simmetrica (chiave unica)
  - Chiave asimmetrica (due chiavi, public key encryption)

- Il modo in cui il plaintext viene processato.

Crittografia (2)

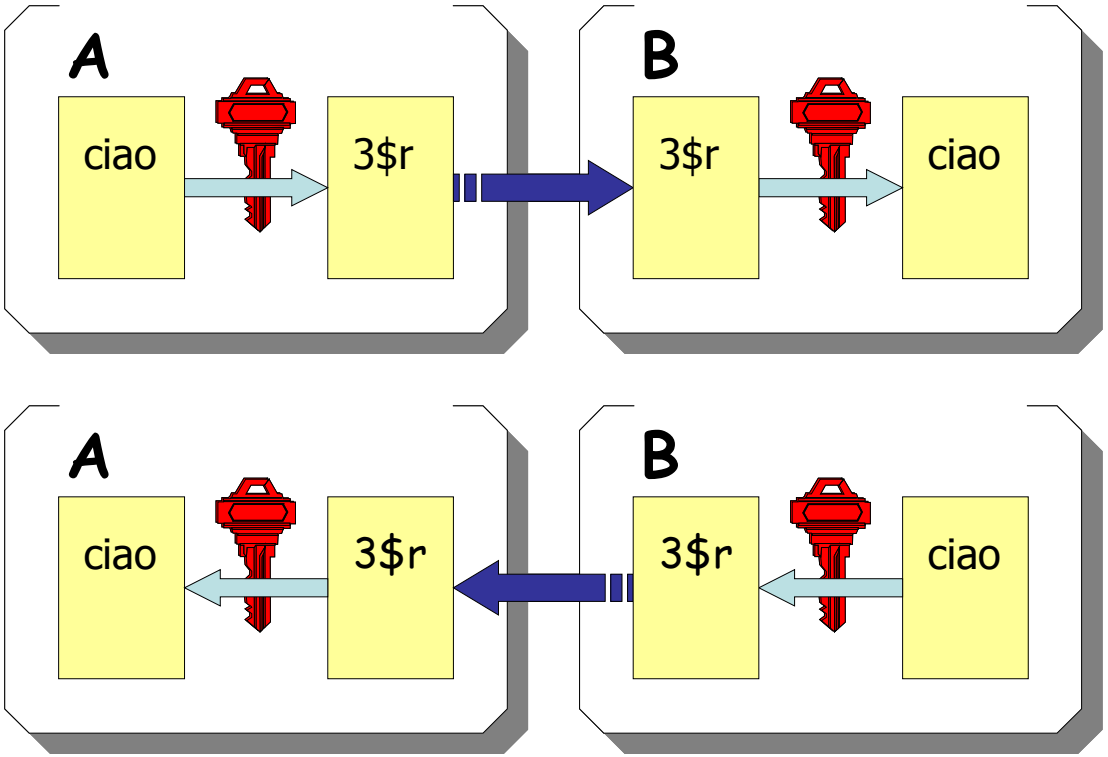
Key Size (bits)	Numero di chiavi alternative	Tempo necessario a 10^6 decritt./μs
32	$2^{32} = 4.3 \times 10^9$	2.15 millisecondi
56	$2^{56} = 7.2 \times 10^{16}$	10 ore
128	$2^{128} = 3.4 \times 10^{38}$	5.4×10^{18} anni
168	$2^{168} = 3.7 \times 10^{50}$	5.9×10^{30} anni

Crittografia (3)

Il sistema più usato è quello a chiave pubblica: facilita la distribuzione della chiave, mantenendo il livello di sicurezza.

Come si fa a scambiarsi la chiave?

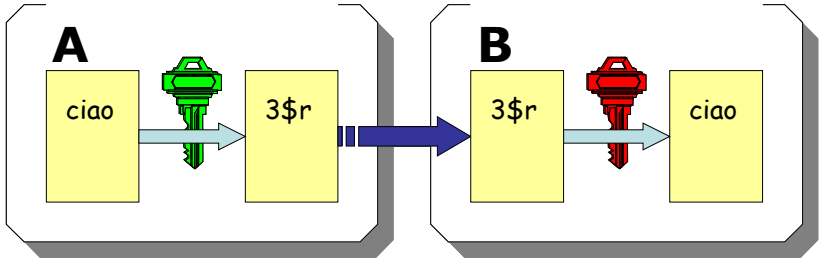
Se ci sono *n* utenti il numero delle chiavi è $O(n^2)$



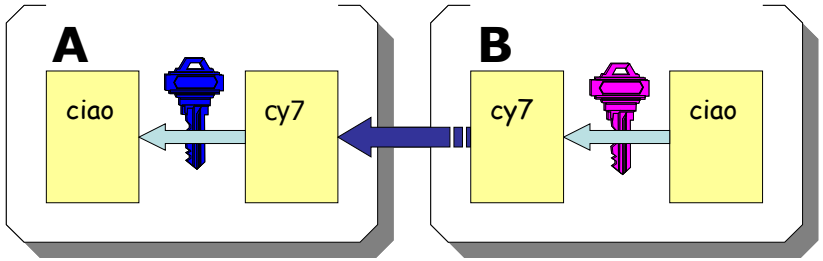
Crittografia (4)

Nel meccanismo detto “a chiave pubblica” ogni utente ha due chiavi (pubblica e privata):

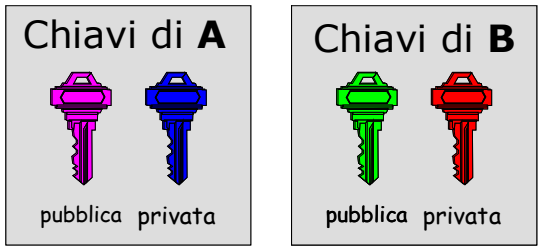
- Da una è impossibile risalire all'altra
- Quello che si cifra con una si decifra con l'altra



Non è necessario lo scambio: il mittente cifra con la pubblica del destinatario. Questi decifra con la sua chiave privata.



Per *n* utenti il numero di chiavi è $O(n)$



Funzioni di hashing

Sono funzioni che accettano in input un messaggio di lunghezza variabile e producono una stringa di lunghezza fissa (*hash*)

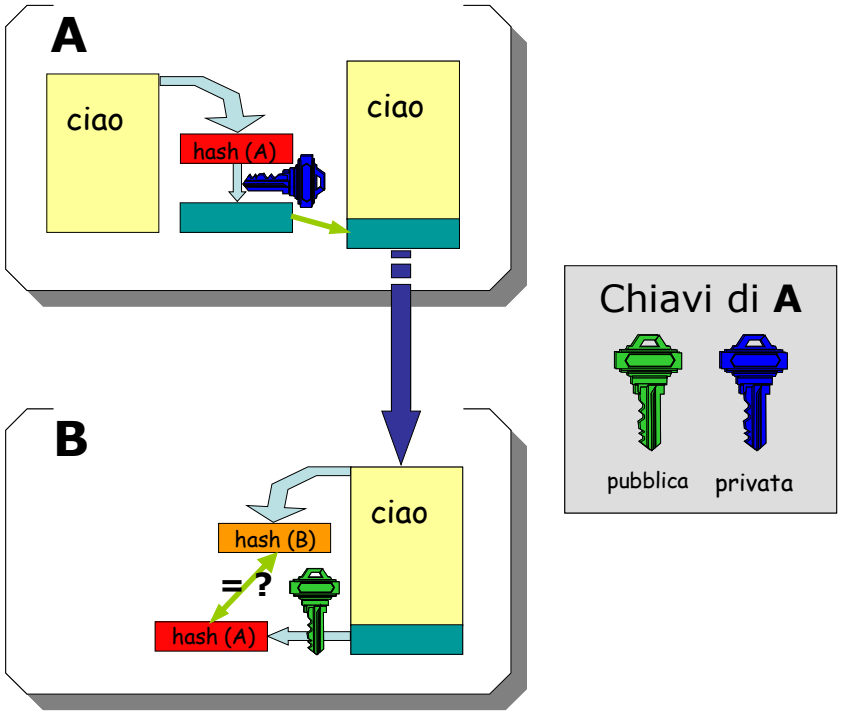
Ovviamente due messaggi diversi possono avere lo stesso hash, ma:

- minime modifiche a un messaggio cambiano radicalmente il suo hash
- la possibilità che due messaggi diversi che abbiano senso e che siano simili/correlati producano lo stesso hash è praticamente nulla
- dato un blocco di dati x, trovare un $y \neq x$ per cui $\mathcal{H}(y)=\mathcal{H}(x)$ è "molto complicato"

Firma digitale

Possiamo mettere insieme hash e cifratura per produrre una firma digitale:

- A calcola l'hash del messaggio e lo cifra con la sua chiave privata; l'hash cifrato è la firma digitale
- A invia messaggio e hash a B
- B ricalcola l'hash e lo confronta con quello ricevuto da A
- Se i due hash sono uguali, il messaggio non è stato modificato e A non può ripudiarlo



Certificati

Un messaggio dotato di firma digitale è “sicuro” se:
- la chiave di A non è stata compromessa
- B conosce la “vera” chiave pubblica di A

La convalida delle chiavi pubbliche viene effettuata attraverso i certificati: una autorità esterna (per es. Certification Authority, CA) garantisce l'autenticità

Esistono due modelli basati su questi presupposti:
- PGP: “web of trust”
- X.509: organizzazione gerarchica

PGP (1)

- Firma digitale e/o criptazione del messaggio: mediante chiave asimmetrica (DSS/SHA e/o RSA/SHA)
- Compressione: comprime (zip) il messaggio dopo la firma ma prima di criptarlo
- Compatibilità con e-mail: usa conversione radix-64 (anche se la dimensione del messaggio aumenta di 1/3)
- Segmentation e reassembly: spesso c'è un limite a 50kB. PGP si occupa automaticamente di spezzare l'informazione in più mail quando si manda. In ricezione basta concatenare i vari mail (senza il loro header)

Come “certificare” l'identità delle persone (scambiarsi le chiavi pubbliche)?

Web of trust (catena di S. Antonio)

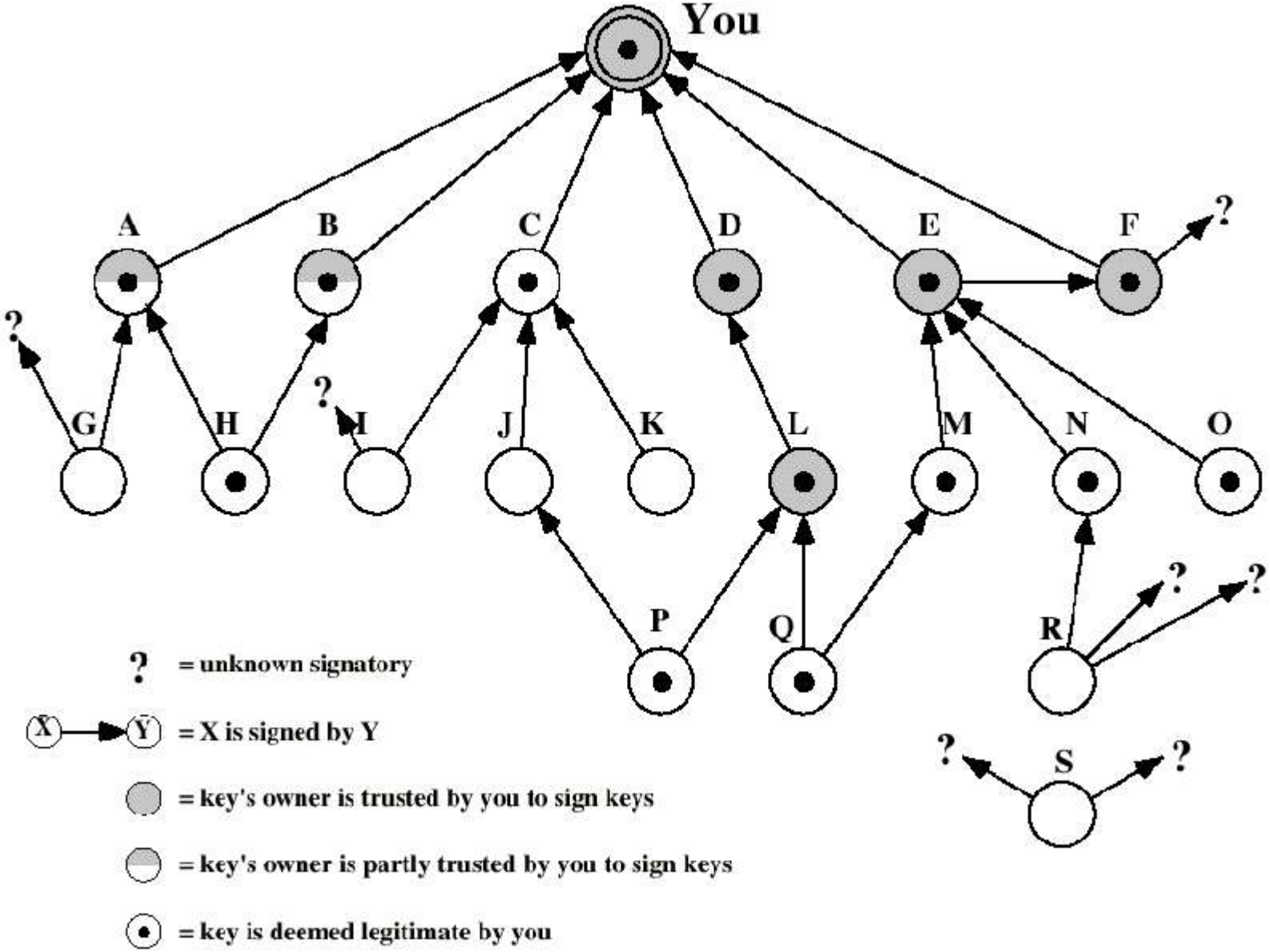
PGP (2)

- Key Ring: privato e pubblico per tener traccia delle chiavi (vostre e degli altri).
- Vari livelli di “trust”: key legimitacy, signature trust, owner trust.

Le chiavi si scambiano di solito mediante incontri tra le persone (colleghi, amici, ecc.) e esiste un meccanismo di controfirma che può essere usato per riconoscimenti a catena.

Es.: A crea una propria chiave e la pubblica su un server di chiavi. B conosce A e controfirma con la sua chiave la chiave pubblica di A. Se C conosce B e accetta chiavi controfirmate da lui, accetta làidentità di A anche se non lo conosce

PGP (3)


You
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
?
= unknown signatory
X
Y
= X is signed by Y
= key's owner is trusted by you to sign keys
= key's owner is partly trusted by you to sign keys
= key is deemed legitimate by you

X.509

Un certificato X.509 contiene sicuramente (altri campi sono opzionali, esistono 3 versioni backward compatibili):

- alcune informazioni sul proprietario
- la data di scadenza
- la chiave pubblica del proprietario
- informazioni sul garante (la CA)

Il certificato è firmato dalla CA

- cioè contiene anche un hash del suo contenuto, cifrato con la chiave privata della CA

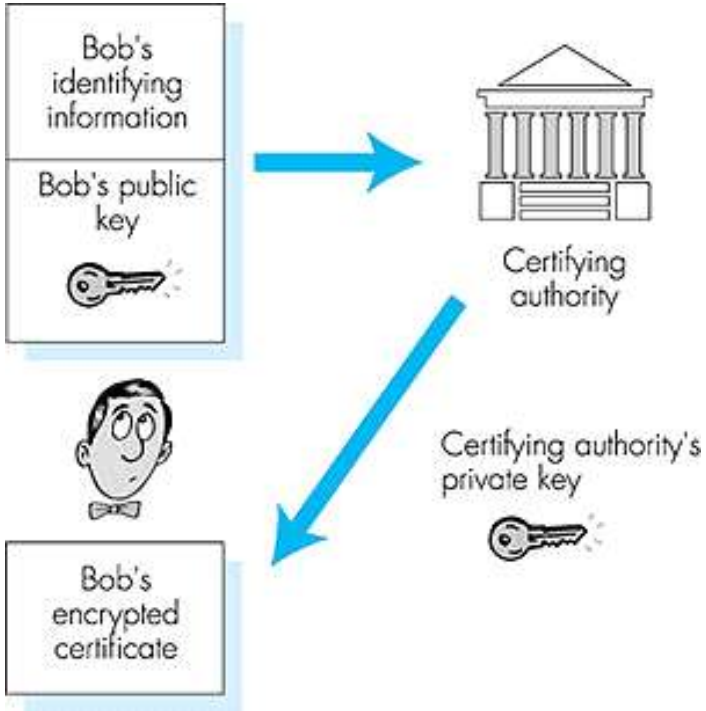
Autorità di certificazione (1)

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



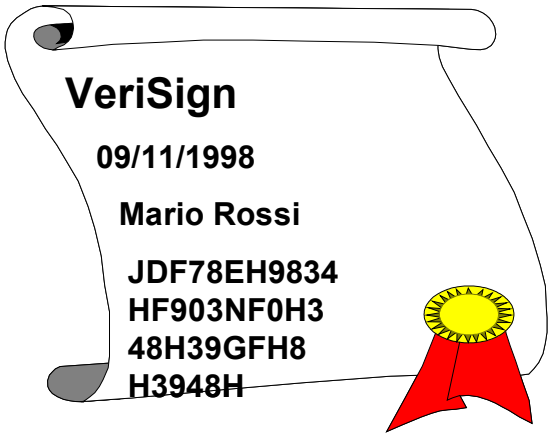
- Alice vuole conoscere la chiave pubblica di Bob:
- Chiede a CA certificato di Bob
- Verifica autenticità del certificato (verifica firma CA)

Autorità di certificazione (2)

L'Autorità di certificazione:

- garantisce la effettiva corrispondenza di una chiave pubblica con il soggetto che la espone

- pubblica, in un apposito registro, <u>certificati firmati con la propria chiave privata</u> 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 (3)

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

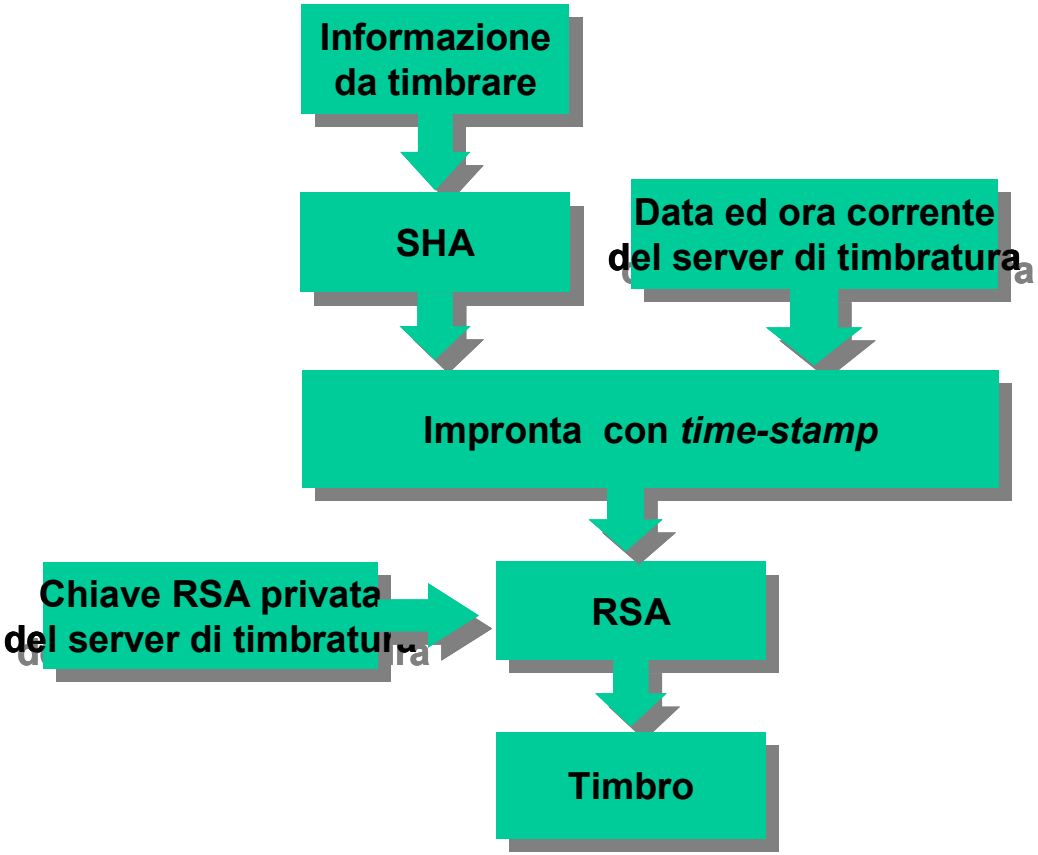
I certificati vanno chiesti al momento della verifica della firma

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 (4)

Allo stesso modo una CA può anche fornire una certificazione temporale (quando è stato creato un documento)

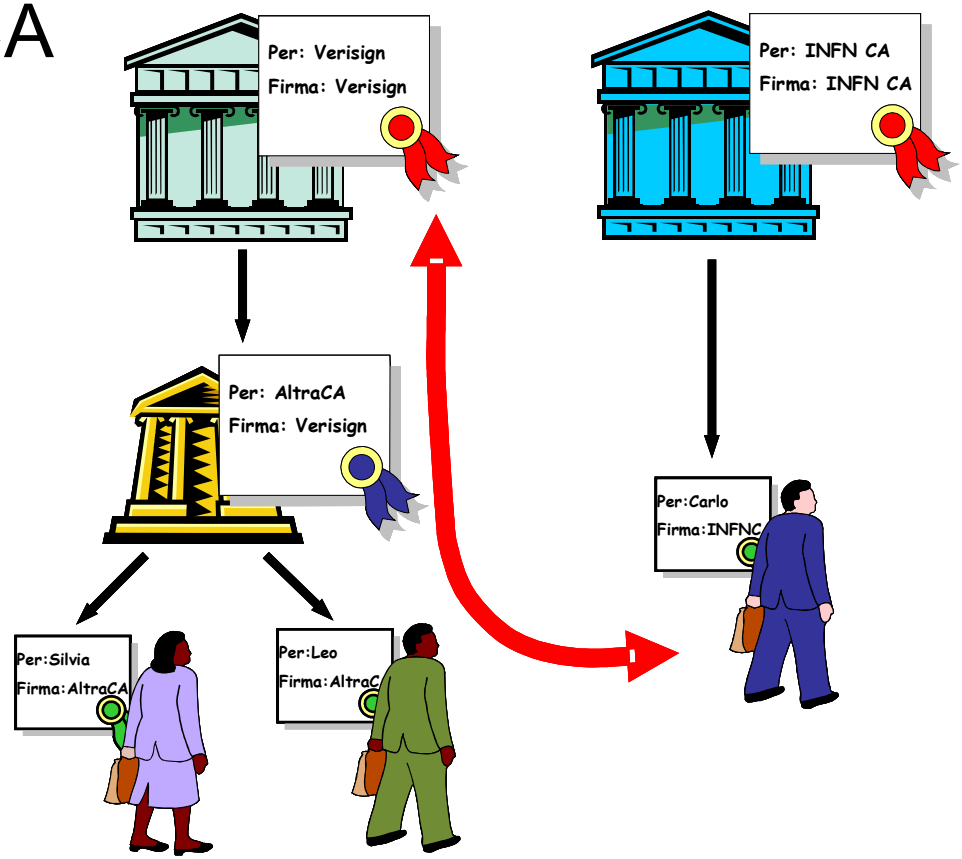


Autorità di certificazione (5)

Anche le CA hanno un proprio certificato

Una CA può garantire un'altra CA (livello inferiore), formando una catena gerarchica di certificati

All'origine della catena c'è una **Root CA** che ha un certificato auto-firmato (root certificate)

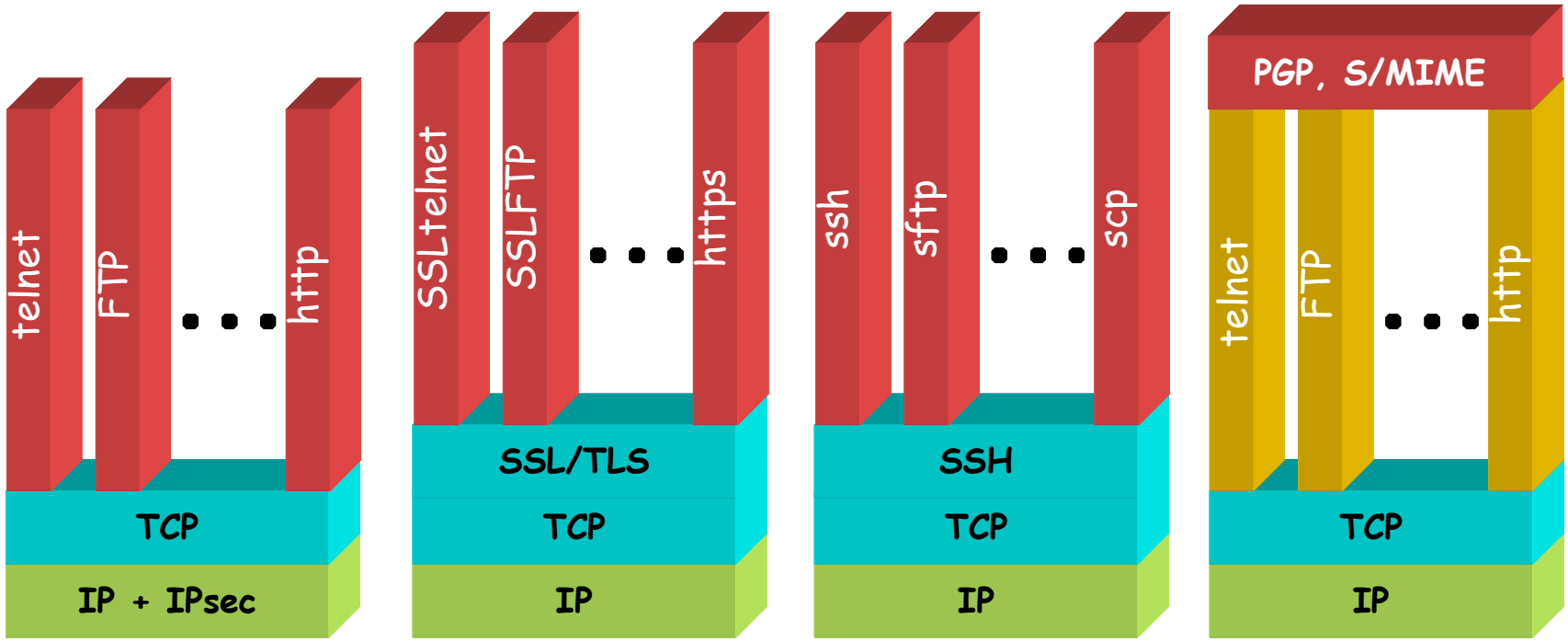


Public Key Infrastructure (PKI)

- Una o più CA organizzate gerarchicamente o via “web of trust”
  - Politica di emissione dei certificati: Certificate Practice Statement (CPS)
  - Emette certificati per server e utenti finali (o altre CA)
  - Mantiene una Certificate Revocation List (CRL)
  - Gestisce servizi WWW e LDAP (pubblicazione cert.)
- Gestione delle chiavi private
  - Generazione
    - via browser
    - via hardware: ad es. SmartCard
  - Custodia
    - Hard disk (insicuro!)
    - SmartCard

Protocolli (1)

La criptazione vista finora (PGP, X.509) fornisce sicurezza per una specifica applicazione. Come generalizzare?



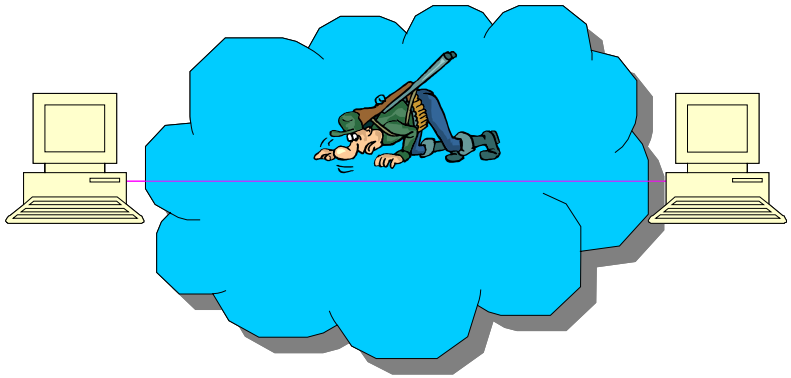
Protocolli (2)

- **Network level**
  - trasparenti per le applicazioni, che non devono essere modificate
  - il network layer deve essere modificato
- **Transport level**
  - viene fornita una libreria di funzioni che può essere utilizzata dai progammi applicativi
  - necessaria la modifica dei programmi applicativi
- **Application level**
  - trasparenti per la rete
  - i servizi di sicurezza devono essere individualmente inclusi in ogni applicazione
  - necessaria la modifica dei programmi applicativi

Ipsec (1)

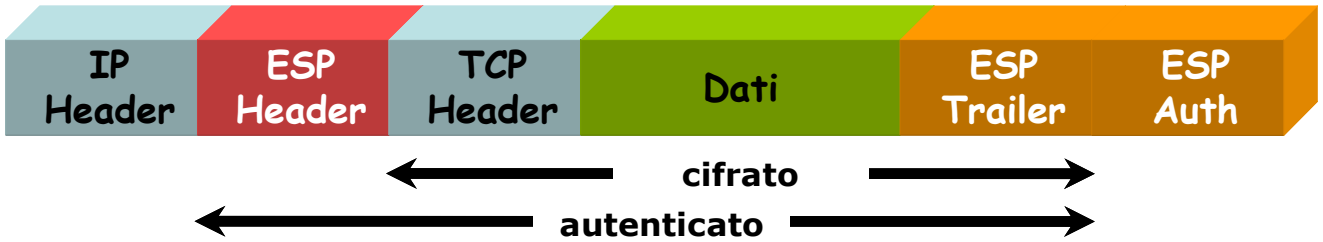
- Autenticazione e cifratura al livello Network
  - Authentication Header (AH)
- autentifica ogni pacchetto;
  - Encapsulating Security Payload (ESP)
- cifra i dati in ogni pacchetto;
  - Internet Key Exchange (IKE)
- protocollo di negoziazione
  - metodi di autentificazione
  - metodi di cifratura
- consente scambio sicuro di chiavi

Ipsec (2)



Authentication Header

Encapsulated Security Payload



AH

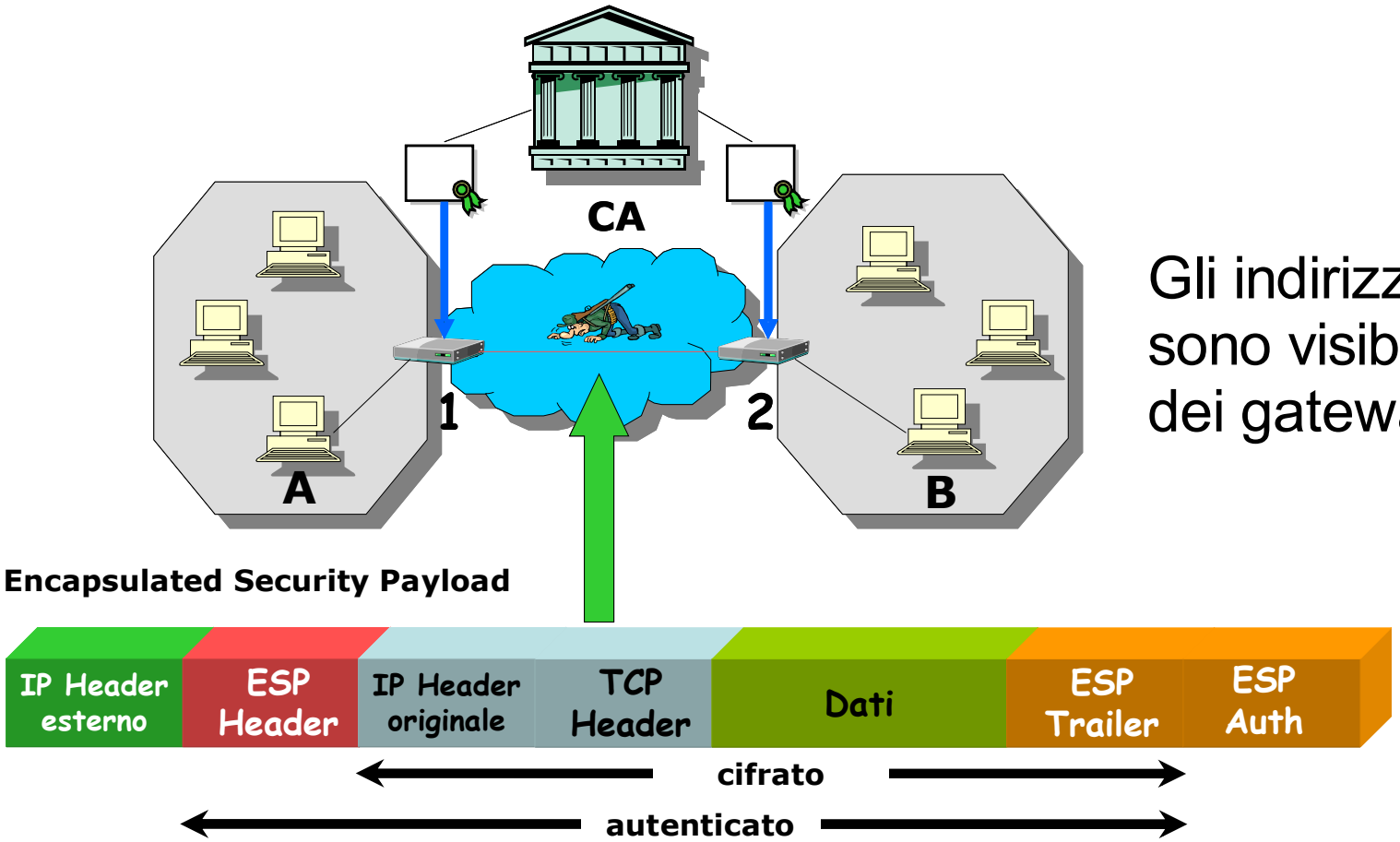
- campi variabili: TOS, Frg offset, TTL, …

ESP

- autentificazione facoltativa
- l'IP header non è protetto

Ipsec (3)

Ipsec Tunnel



Gli indirizzi di A e B non sono visibili all'esterno dei gateway 1 e 2

SSH

- Rimpiazza telnet e i comandi “r” (rlogin, rshell)
  - Versione 1 e Versione 2 (IETF SECSH Working Group);
    - nella versione 1 richiede una distribuzione “manuale” delle chiavi di host e utenti;
  - connessione crittografata: protegge da:
    - IP spoofing;
    - IP source routing;
    - DNS spoofing;
    - sniffing;
    - man-in-the-middle;
  - tunneling traffico tcp e X11;
  - compressione dei dati (facoltativa, utile per connessioni lente).
    - Molto apprezzato anche dagli hacker, che spesso lo installano sulle macchine compromesse, perché impedisce agli amministratori di capire quali dati vengano trasferiti e per il meccanismo di port forwarding.

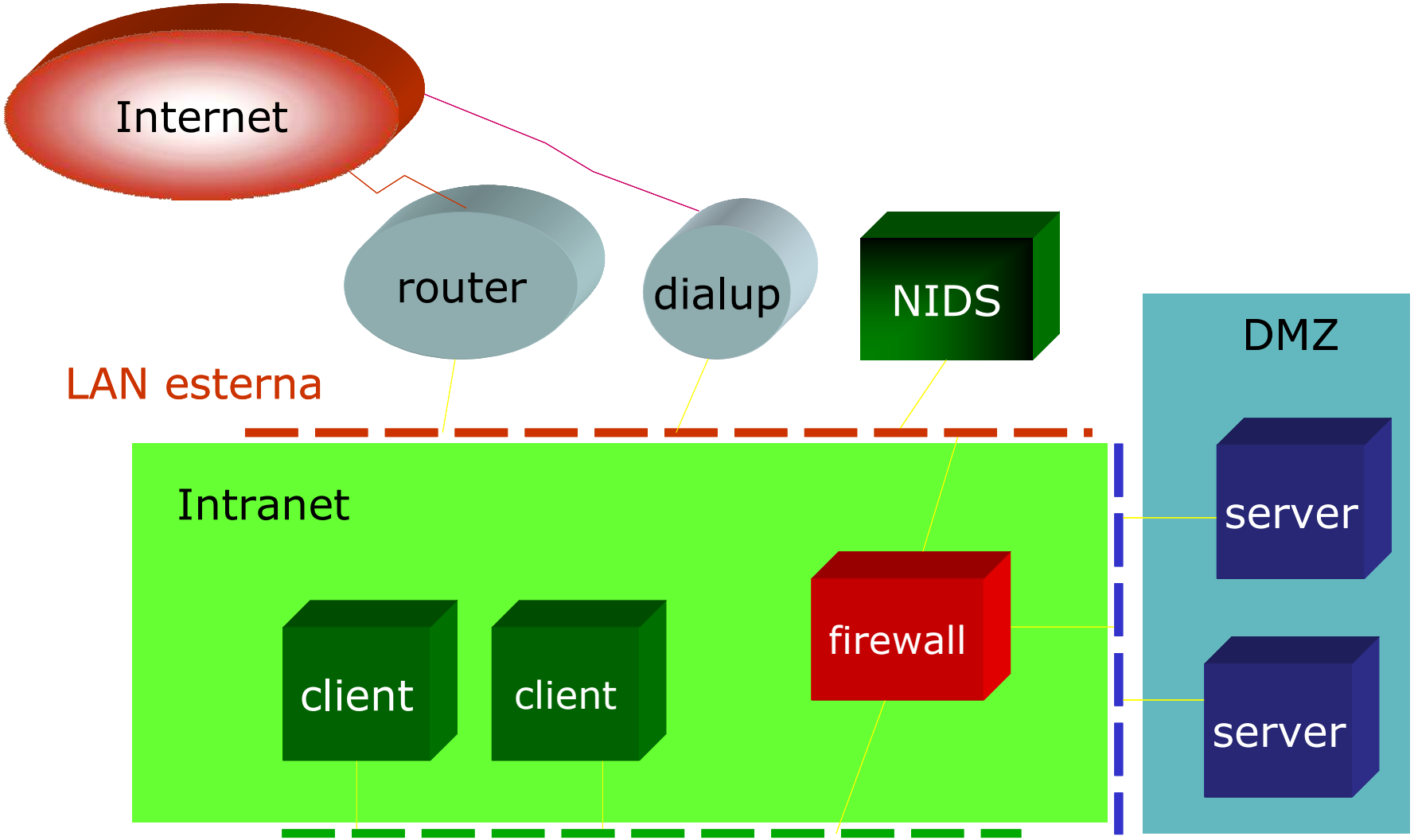
SSL/TLS

- Secure Sockets Layer (SSL) sviluppato da Netscape Communications
  - L'ultima versione (V3.0) è del Marzo 1996;
  - Netscape Communicator, Internet Explorer.
- Transport Layer Security (TSL) Working Group (IETF)
  - versione 1.0 del Gennaio 1999 (RFC 2246).
- Utilizza certificati X.509
- Può essere usato per ogni applicazione TCP (ad es. HHTP, Telnet, FTP, POP3, IMAP)
  - usato da praticamente tutti i server web "sicuri": https://…..
  - le vecchie applicazioni "insicure" possono essere usate in modalità tunnel (ad es. stunnel: http://www.stunnel.org)
- Non interagisce bene con firewall/proxy (man-in-the-middle)

S/MIME

- Permette di inviare e-mail MIME firmati e crittografati
- Supportato (tra gli altri) da *Communicator* e *Outlook*
- Utilizza certificati X.509 v.3

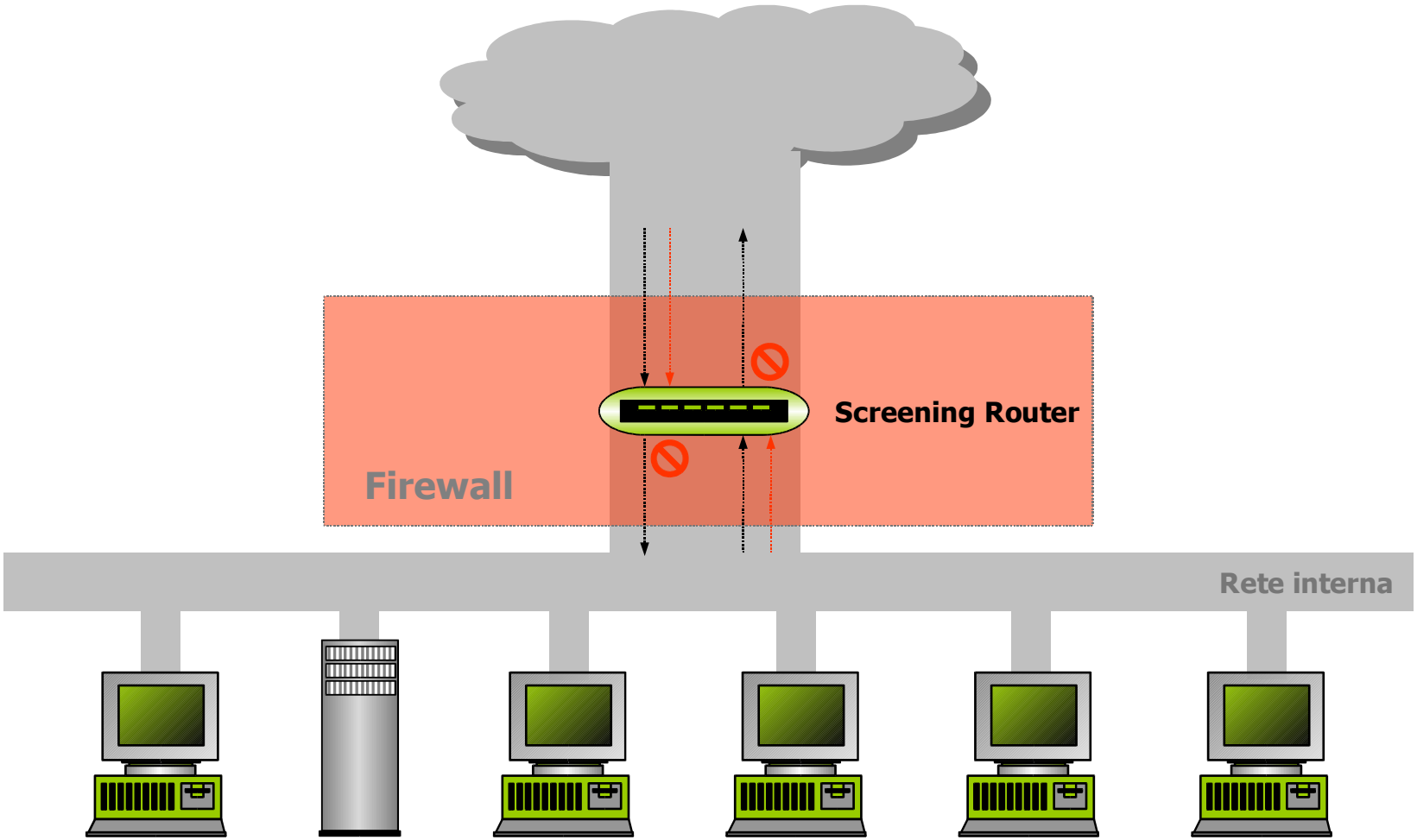
Internet e intranet



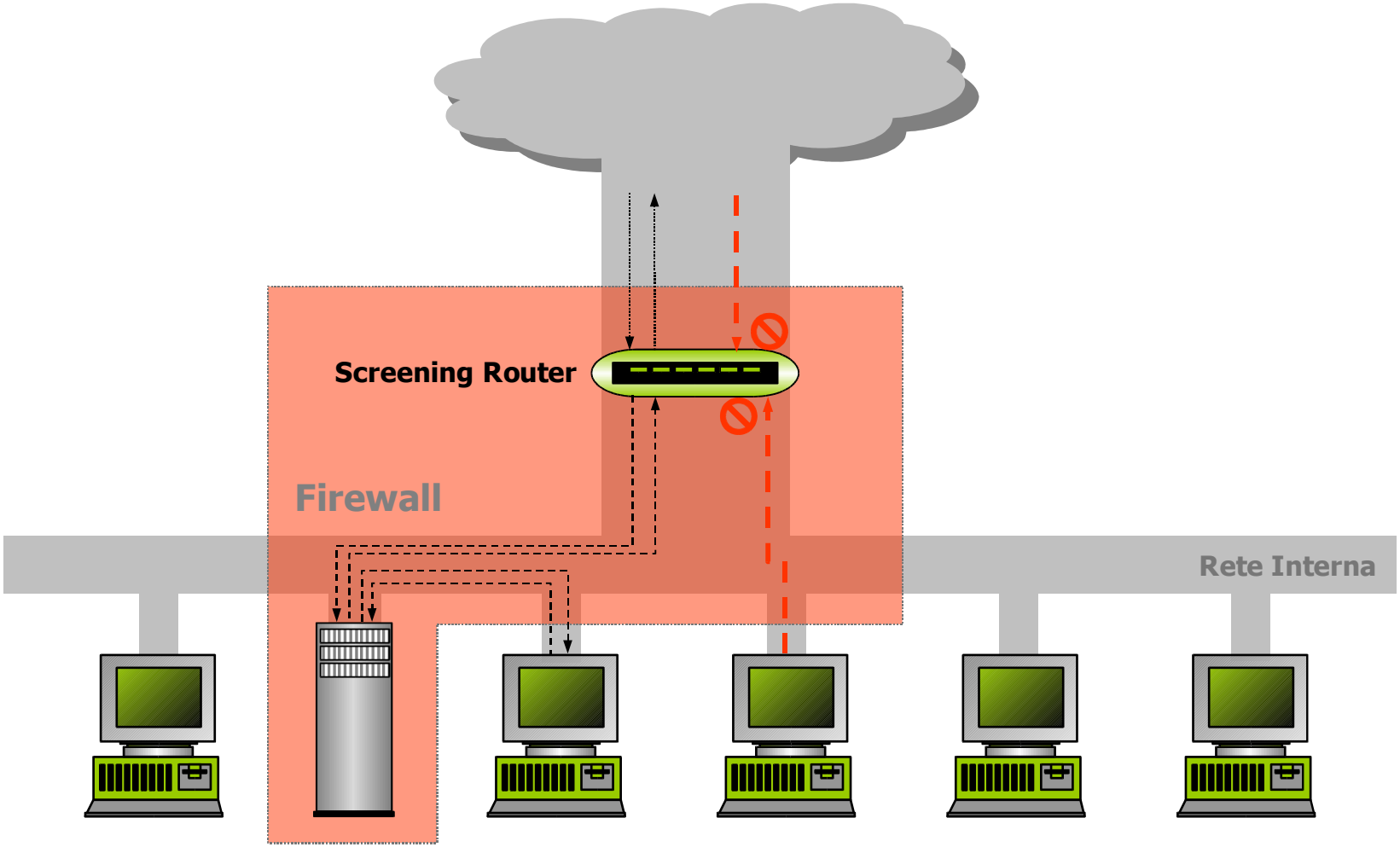
Firewall

- Network device con almeno 2 interfacce di rete
  - Router o hardware dedicato
- Separa zone amministrativamente diverse
  - LAN esterna (insicura)
  - DMZ
  - Intranet
- Effettua routing fra le diverse zone
- Può effettuare mascheramento indirizzi (NAT)
- Filtra traffico fra le diverse zone tramite regole predefinite
  - Router con filtri è un firewall!
- Può mediare accessi a specifiche applicazioni
  - Proxy server

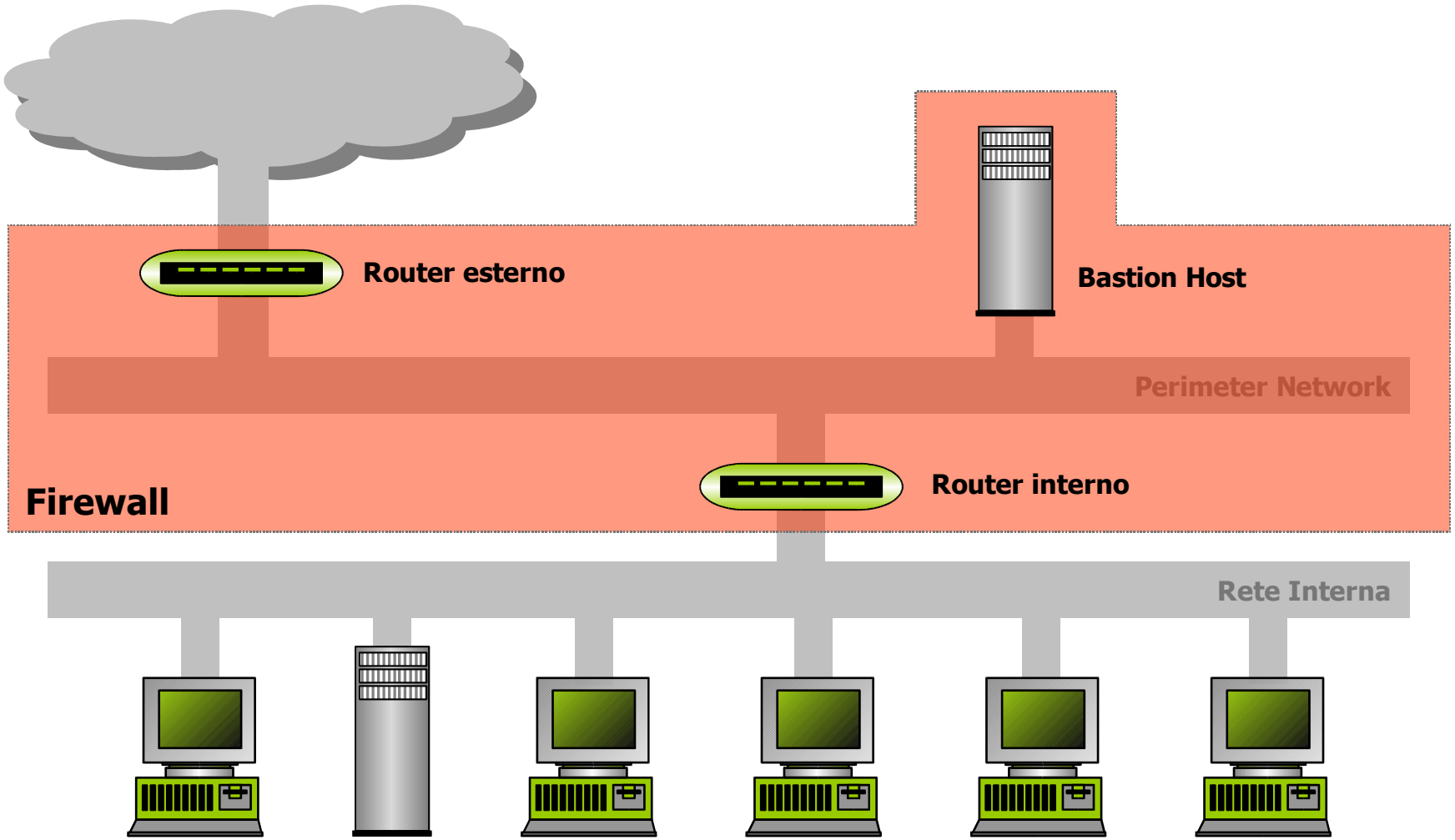
Architetture: screening router



Architetture: screened host



Architetture: screened subnet



Firewall: pro e contro

1. Sono troppo complessi
2. I nuovi servizi sono incompatibili con i FW
3. Sono di difficile gestione, troppo costosi e rapidamente obsoleti
4. Bloccano servizi utili e riducono l'utilità di Internet
5. Ad ogni nuovo servizio il FW deve essere riconfigurato
6. Il nuovo sw ha incluso il supporto per la sicurezza (ad es. SSL)
7. Riducono le prestazioni in modo eccessivo
8. I servizi che vengono fatti passare possono avere problemi di sicurezza
9. Non è in grado di bloccare i virus o applet ostili
10. IPV6 renderà i FW obsoleti

1. Il sw disponibile non è ancora di qualità adeguata
2. In molti casi è l'unica protezione possibile
3. Il costo di un incidente di sicurezza è ben maggiore
4. Se le applicazioni sono scritte bene i FW non sono di intralcio
5. In questo modo ogni novità rimane sotto controllo
6. Il nuovo sw spesso è meno sicuro di quello vecchio
7. È solo questione di comprare hw più veloce
8. Anche le macchine interne devono essere ben configurate
9. I FW non sono lo strumento adatto per bloccare virus e applet
10. IPV6 è ancora lontano.

Access control (1)

Altro aspetto della sicurezza: autorizzazione.

Non ci interessa se l'utente è chi dice di essere, ma se i suoi privilegi gli consentono di fare una operazione.

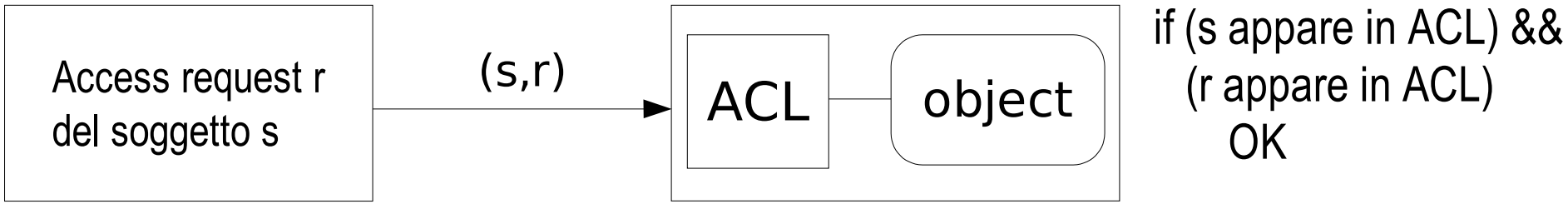
Modello “classico”: un *soggetto* vuole accedere ad un *oggetto*.....

Access control (2)

Si construisce una access control matrix: ogni soggetto è rappresentato da una riga, ogni oggetto da una colonna.

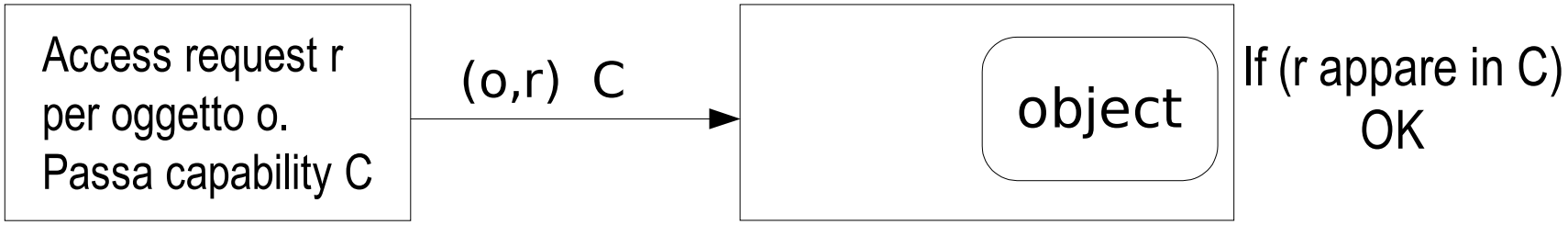
Migliaia di utenti, milioni di oggetti ⇒non pratico

Un possibile approccio: ogni oggetto mantiene una lista dei diritti dei soggetti (distribuire la matrice per colonne tra i vari oggetti e sopprimere gli zeri). Si ottiene una Access Control List per ogni oggetto.



Access control (3)

Un altro approccio: consideriamo la tabella per righe, dando a ciascun soggetto una serie di “capabilities” associate ai vari oggetti. Quando si fa una richiesta, si mandano anche le capabilities. Ovviamente le capabilities vengono assegnate in qualche modo e l'utente non può alterarle a piacimento.



Access control (4)

Come ridurre le liste (ACL o capabilities)?

Protection domains: un insieme di coppie (oggetto, access right). Quando un soggetto effettua una richiesta, lo fa sempre all'interno di un dominio. Il dominio può essere fatto mediante gruppi di utenti, gerarchico, ecc.
Il soggetto può usare un certificato per segnalare il proprio dominio.....

Invece che gruppi, si possono avere dei “ruoli”. Il modo in cui un utente si collega determina il ruolo, e l'utente può cambiare ruolo durante la sessione.

GRID AA (1)

Come esempio pratico, consideriamo il caso di GRID (o Globus).

Cos'è? Un sistema per applicazioni distribuite su larga scala, in cui (idealmente) molti computer, file, ecc. (in pratica molte “risorse”) sono necessarie per eseguire un programma, e queste risorse sono distribuite.

Risorse distribuite: appartengono a “administrative domains” diversi, con regole diverse, quindi il problema non è banale...

GRID AA (2)

Cercando di ridurre il problema al minimo....

1) La grid copre più dominii amministrativi
2) Operazioni locali (all'interno dello stesso dominio) sono soggette alla policy di quel dominio
3) Operationi globali richiedono i privilegi “giusti” nei vari dominii coinvolti
4) Operazioni tra entità diverse in dominii diversi richiedono mutua autenticazione
5) Al posto di una autenticazione locale si usa una autenticazione globale
6) L'accesso alle risorse di un dominio è un problema locale
7) Gli utenti possono delegare privilegi ai loro processi
8) Più processi in uno stesso dominio posso condividere le credenziali

GRID AA (3)

Autenticazione: questa è la parte “facile”. Tutto il sistema è fatto per funzionare a base di certificati (l'unico problema è stato definire il mutual trust)

Autorizzazione: questa è la parte non ovvia. Entrano in gioco almeno due tipi di entità: Resource Providers e Virtual Organizations.

Accordi tra i due tipi di entità:

- la VO decide chi ne fa parte, i ruoli degli appartenenti e le loro capabilities
- RP l'autorizzazione garantita dalla VO ad un utente, e mappa l'utente ad credenziali locali adeguate ad accedere alle risorse

GRID AA: mark 0

Sistema originale di Globus: grid-mapfile

Mapping tra le credenziali della grid (certificato dell'utente) e le credenziali locali (account unix)

Ogni rica contiene la coppia subject certificato, account unix esistente.

A parte la non scalabilità, qui esiste solo RP (non si parla di appartenenza a una VO, ruoli, ecc.)

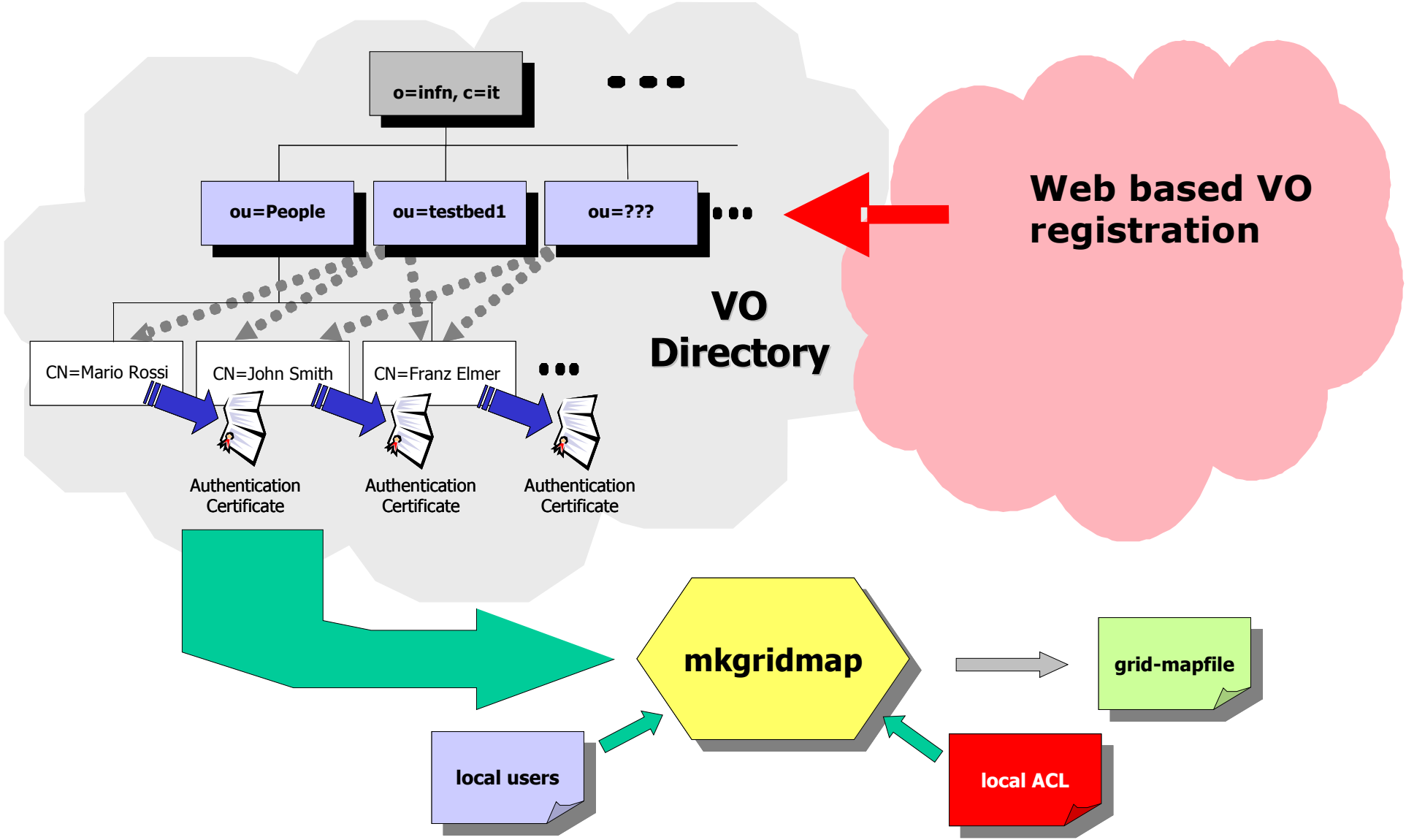
GRID AA: mark 1 (1)

Per ottenere una VO, il primo approccio è stato l'uso di LDAP (directory service)

Ogni CA pubblica i certificati su un server LDAP
Ogni VO utilizza una directory LDAP con:
•Un contenitore con elenco appartenenti
•Più contenitori (gruppi): ogni utente deve appartenere ad almeno un gruppo

Il grid-mapfile viene generato stabilendo quale VO (o quale gruppo) si accetta, mappando gli utenti globali ad utenti locali (pool di utenti)

GRID AA: mark 1 (2)



GRID AA: mark 2 (1)

Usare LDAP ha qualche inconveniente:

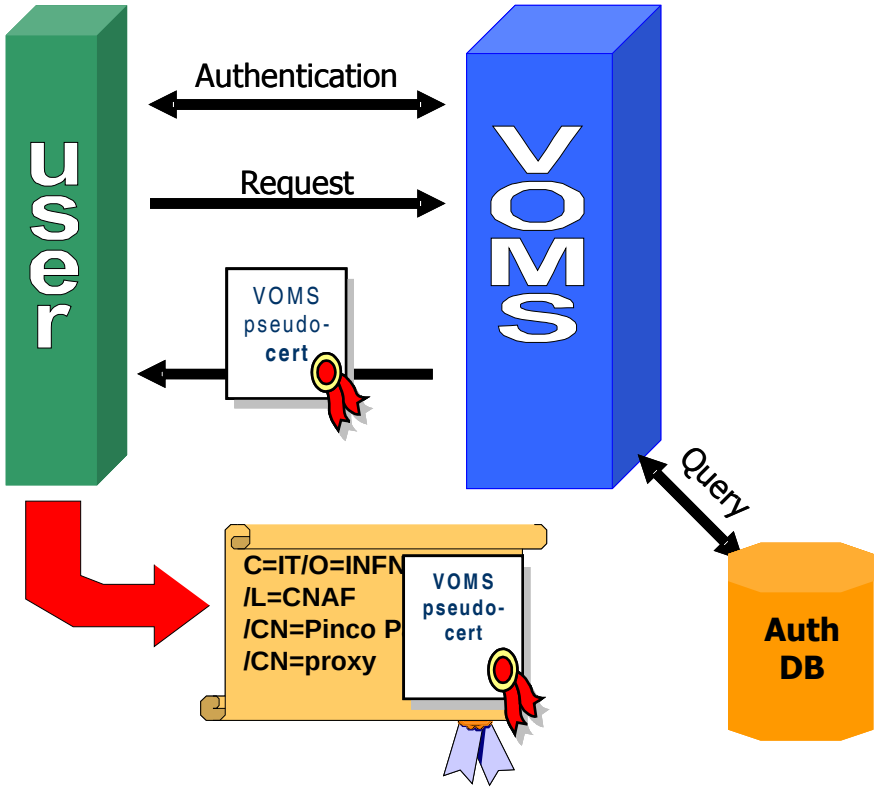
- poco flessibile
- problemi di scalabilità
- ci sono i gruppi ma non i ruoli

Per aggirare i vari problemi si passa a:

- VO Membership Service (VOMS) – associa a ogni utente i dati di autorizzazione a livello di VO
- Local Center Authorization Service (LCAS) – gestisce le richieste autorizzazione al sistema locale
- Local Credential Mapping Service (LCMAPS) – provvede alle credenziali locali di cui hanno bisogno i jobs in esecuzione localmente

GRID AA: mark 2 (2)

1) mutua autenticazione utente-VOMS (globus API)
2) cliente manda richiesta a server
3) server controlla richiesta
4) server rimanda le informazioni come pseudo-certificato (autofirmato)
5) cliente controlla validità pseudo-certificato
6) cliente ripete operazione con altri VOMS se necessario
7) cliente crea proxy certificate usando il suo certificato e aggiungendo le informazioni dei VOMS in estensioni non critiche



GRID AA: mark 2 (3)

Formato del pseudo certificato

è inserito in una estensione non critica di un proxy certificate

diventerà un attributo ufficiale

/C=IT/O=INFN/L=CNAF/CN=Vincenzo Ciaschini/Email=Vincenzo.Ciaschini@cnaf.infn.it

/C= IT/O=INFN/CN=INFN CA

user's identity

/C=IT/O=INFN/OU=gatekeeper/L=PR /CN=gridce.pr.infn.it/Email=alfieri@pr.infn.it

/C=IT/O=INFN/CN=INFN CA

VO: CMS URI: http://vomscms.cern.ch

server identity

TIME1: 020710134823Z
TIME2: 020711134822Z
GROUP: montecarlo
ROLE: administrator
CAP: "100 GB disk"

user's info

SIGNATURE:
.....L...B]....3H.....=".h.r...;C'..S.....o.g.=.n8S'x..\..A~.t5....90'Q.V.I. .../.Z*V*{.e.RP.....X.r.....qEbb...A...