La Sicurezza nei Sistemi Distribuiti

Protezione delle **Risorse** e delle **Informazioni**

Risorse di
- ***esecuzione***
- ***memorizzazione***
- ***comunicazione***

Aspetti del problema **Sicurezza**:

Autenticazione	*Authentication*
Autorizzazione	*Authorisation*
Riservatezza	*Privacy*
Disponibilità	*Availability*
Integrità	*Integrity*
Paternità	*Non-repudiability*

Autenticazione

Verifica dell'identità dell'utente attraverso:

- Possesso di un oggetto (es., smart card)
- Conoscenza di un segreto (password)
- Caratteristica personale fisiologica
  (impronta digitale, venature retina)

Problema della mutua autenticazione

Autenticazione ≠ Autorizzazione

Autorizzazione

Specifica le azioni concesse ad ogni utente

Riservatezza

Previene la lettura non autorizzata delle informazioni.
(es. un intruso che intercetta un messaggio non è in grado di interpretarlo, di risalire al contenuto informativo del msg)

Integrità

Previene la modifica non autorizzata delle informazioni
(es. un messaggio spedito dal mittente è ricevuto tale e quale dal destinatario)

Alcune modifiche dell'informazione possono
non alterare l'integrità

Disponibilità

Garantire in qualunque momento la possibilità di usare le risorse

Paternità

Il mittente non può ripudiare un messaggio

Esempio

Alice manda un messaggio a Bob dicendogli costruiscimi una casa per 1 miliardo

Problemi

Intrusione: Trudy intercetta il messaggio e cambia 1 miliardo in 100 miliardi (violata l'**integrità** del messaggio)
Soluzione: cifro il messaggio, Trudy può intercettare il messaggio ma non lo sa leggere

Risoluzione controversia tra i comunicanti: il messaggio è cifrato (**identità** del mittente) *ma* sia Bob che Alice conoscono come cifrare il messaggio e possono mentire, senza che l'altro possa dimostrarlo
due casi:

- Alice non ha più i soldi, ma Bob ha costruito la casa
- Bob costruisce una casa di costo 10 miliardi

In un regime di mutuo sospetto nessuno può provare ad una terza parte che l'altro sta mentendo.
Soluzione: Alice firma il messaggio, chiunque potrà verificare che Alice ha chiesto una casa da 1 miliardo e che così è scritto nel messaggio

Attacchi alla sicurezza del sistema

CHI può fare **QUALI** operazioni e su **QUALI** oggetti

Tentativi di intrusione

passivi

attivi

Tentativi passivi

- leggere informazioni di altri
- eseguire analisi del sistema, del traffico

Tentativi attivi

- modifica dell'informazione (es., messaggio, file ...)
- cancellazione dell'informazione (es., messaggio, file ...)
- impersonare un altro utente
- accesso non autorizzato alle risorse
- sistema impossibilitato a fornire i servizi

PRINCIPIO di SICUREZZA MINIMO

Proteggersi dagli attacchi passivi

Accorgersi degli attacchi attivi

Non si tengono nascoste le strategie di sicurezza

metodi per accedere impropriamente alle risorse:

Leaking acquisizione informazioni senza averne l'autorizzazione (confinare dominio dati)

Browsing lettura tutti i pacchetti

Inferencing deduzione informazioni dai dati stessi

Masquerading fingersi un utente diverso da quello reale

Criteri per valutare la sicurezza dei sistemi

Varie politiche di protezione dei dati:

- accesso ai dati sulla base dello user id e del gruppo di appartenenza (es. Unix)
- livelli di protezione crescenti associati ai dati, sviluppata principalmente in ambito militare, es. informazioni di tipo:
  unclassified, *confidential*, *secret*, *top secret*

Orange Book

redatto in ambito militare (U.S.) si concentra sulla riservatezza dei dati e sul mandatory access control (non sull'integrità dei dati)

Red Book (TNI)

Applica i criteri di sicurezza definiti nell'Orange Book ai sistemi interconnessi in rete

Lavender Book (TDI)

Applica i criteri di sicurezza definiti nell'Orange Book ai data base management system

Orange Book, Red Book e Lavender Book costituiscono la "Rainbow" series

Information Technology Security Evaluation Criteria (**ITSEC**) (Germania, Francia, UK, Olanda) ha redatto un rapporto per valutare la sicurezza dei sistemi diretti a gestire informazioni unclassified

Sforzo congiunto USA, Canada ed Europa per lo sviluppo di un "**Common Criteria**"

Relazione tra standard per Sistemi Aperti e requisiti per la sicurezza (livello implementativo)

Orange Book

Ogni livello ingloba gli elementi di sicurezza dei precedenti e vi aggiunge quelli caratteristici del livello stesso

Categoria D: **non sicuri**

Categoria C: **sistemi discrezionali**

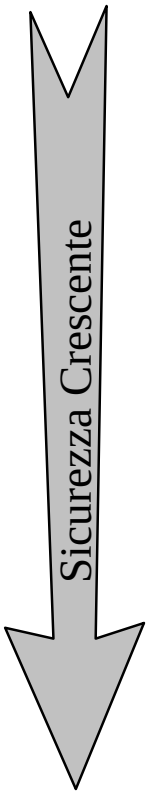
Protezione oggetti a discrezione dell'utente
il sistema non forza criteri
assunzione di base: buonafede utenti

(protezione dall'esterno)

Categoria B: **sistemi non discrezionali, protezione obbligatoria**

Oggetti protetti in modo obbligatorio dal sistema
assunzione di base: possibile malafede utenti

Categoria A: **modello formale di sicurezza**; **informazioni classificate**



Orange Book

Categoria D: **non sicuri**

Categoria C: **protezione possibile**

classe C1:

- autenticazione utenti via password
- semplici controlli degli accessi alle risorse (owner/group/other)
- accesso controllato degli utenti a certe parti della memoria

classe C2:

- controllo degli accessi alle risorse con granularità utente
- cancellazione memoria prima del suo assegnamento agli utenti
- auditing

Categoria B: **protezione obbligatoria**

classe B1:

- etichette di sicurezza attaccate a tutti gli utenti, i processi e le risorse, S.O. controlla i read-up e i write-down
- tutti i dispositivi (stampanti etc.) devono trattare opportunamente le etichette di sicurezza

classe B2:

- trusted path tra S.O. e utente, presenza cioè di meccanismi per garantire che un utente parli al S.O. (trojan horse)
- processi che cambiano livello di sicurezza notificano utente
- strutturazione del kernel: concentrazione delle funzionalità di sicurezza all'interno di un piccolo "security kernel"
- identificazione dei covert channel e delle loro bande di trasmissione
- strette procedure di controllo nella modifica delle parti del sistema sensibili per la sicurezza

classe B3:

- controllo accessi per utenti con overrule
- auditing attivo
- secure crashing e restarting

Categoria A: **modello formale di sicurezza**; **informazioni classificate**

classe A1:

- stessi requisiti precedenti ma è richiesta la verifica del progetto

POSIX.6

POSIX.6: Protection, Audit and Control Interface Standard

1. per realizzare applicazioni portabili, definisce le **interfacce** necessarie alla gestione delle informazioni di sicurezza.
2. per migliorare i meccanismi di sicurezza del POSIX.1

Aree funzionali indirizzate:

- auditing
- discretionary access control
- mandatory access control
- information labels
- privilege

Discretionary Access Control in POSIX.6

Il proprietario dell'oggetto stabilisce i diritti di accesso:

POSIX.1 adotta il meccanismo dei **permission bit**:

- ☺ semplice ed economico
- ☹ scarsa granularitá nel controllo degli accessi (non si possono indirizzare utenti individuali o specifici gruppi)

POSIX.6 adotta le **Access Control List** (ogni oggetto ha una ACL associata che specifica i diritti di accesso sulla risorsa per gruppi e utenti)

POSIX.6 NON specifica come implementare le ACL né la rappresentazione interna delle ACL.

Mandatory Access Control in POSIX.6

1. La protezione di un oggetto non è decisa dal proprietario dell'oggetto
2. Il sistema assicura e impone la protezione degli oggetti

POSIX.6 usa un labeling mechanism

soggetti = processi
oggetti = file (regular file, directory, device, etc.), processi

Tutti i soggetti e oggetti del sistema hanno una MAC label, in ogni momento (unclassified, confidential, secret, top secret).

Un soggetto senza privilegi non può rendere disponibile una informazione con label L1 ad un soggetto di label L2 se L2 < L1

Restrizioni nell'accesso ai file:

- **Lettura** file consentita se la label del processo lettore è maggiore della label del file (no read-up)
- **Scrittura** di un file consentita se la label del file è MAGGIORE della label del processo (no write-down)

Esempio:

- Un processo Secret non può leggere un file Top Secret
- Un processo Secret non può scrivere su un file con label Confidential

Privilegi in POSIX.6

Il **meccanismo dei privilegi** controlla i diritti di accesso alle risorse da parte degli utenti e dei gruppi.

Può fornire a specifici utenti la capacitá di eseguire operazioni “sensibili” ai fini della sicurezza, sotto certe condizioni e per un tempo limitato.

Esempio
Per eseguire un **backup** del sistema, l’amministratore di sistema deve acquisire il **privilegio di lettura** su tutti i file (**override** delle informazioni di controllo dell’accesso su tutti i file)

Esempio
UNIX **super user** è un meccanismo di overriding dei privilegi di tipo “tutto o niente”

POSIX.6 suggerisce un meccanismo di tipo “**least privilege**”, per permettere un override del minor numero possibile di informazioni di sicurezza per lo svolgimento di un determinato compito.

Caratteristiche di un meccanismo di overriding dei privilegi:
- granularitá
- durata temporale
- ereditarietá

Controllo di accesso di sistema

Modello

- **a matrice**
- **a reticolo**
- **a flusso di informazioni**
- **a security kernel**

Modello a matrice

controllo solo sintattico sulle azioni dei soggetti sugli oggetti

Modello a reticolo

soggetti ed oggetti con classificazione
anche commerciale

Modello a flusso

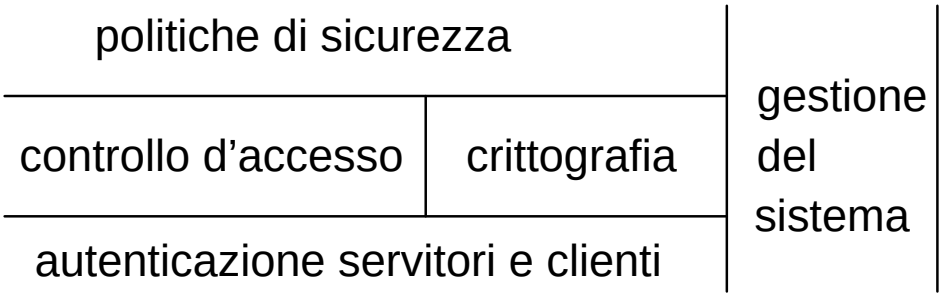
analisi del flusso di informazioni tra soggetti di classi diverse

Modello a kernel

definizione di un kernel centralizzato per la verifica di protezione

Sicurezza CLIENTE/SERVITORE

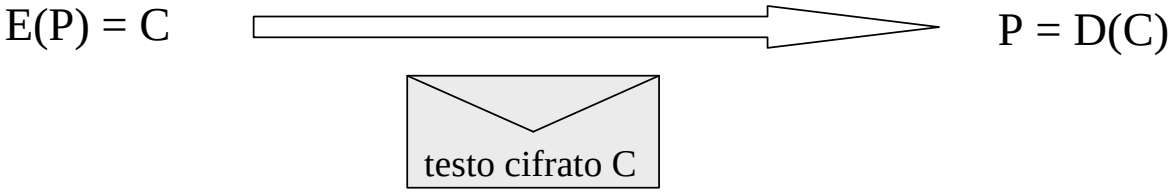
sicurezza sul **canale** con strumenti diversi
rapporto relazione cliente/servitore basato sul principio del **sospetto reciproco**
messaggi e autorizzazioni usati **una volta sola** per non avere problemi di ripetizioni da intrusori



Vediamo prima il problema della crittografia e poi passiamo all'inserimento di strumenti per un sistema **sicuro**

CRITTOGRAFIA

la scienza della crittografia fornisce gli strumenti per rendere un messaggio non intelligibile per chiunque non ne sia il destinatario



E = funzione di cifratura
D = funzione di decifrazione

P = plaintext, testo in chiaro
C = ciphertext, testo cifrato

CRITTOGRAFIA

di sostituzione
di trasposizione

Codici di sostituzione (monoalfabetica)

una permutazione sull'alfabeto del **testo chiaro**

se ogni simbolo corrisponde a un byte
alfabeto di 256 simboli ==>
ogni **simbolo** si trasforma in un altro simbolo dell'alfabeto:
- cambiano simboli ma non cambia il loro ordine

256! possibili **permutazioni** dell'alfabeto

La chiave di cifratura è la tabella con le corrispondenze di tutti i simboli (tabella 256x2)

Cifrario di Cesare è un codice di sostituzione monoalfabetica in cui ogni carattere viene sostituito con quello che lo segue di 3 posizioni nell'ordinamento alfabetico (la chiave in questo caso è 3).

Facilmente attaccabile

Esistono anche codici di sostituzione polialfabetica.

Codici di trasposizione

si fissa un numero intero P (periodo della trasposizione) e si sceglie una permutazione degli interi da 1 a P

se P=7 si può fare corrispondere alla successione
1 2 3 4 5 6 7 quella permutata 4 6 3 5 7 1 2

- i simboli dell'alfabeto sono gli stessi ma cambia l'ordine

Il testo chiaro viene considerato a **blocchi di lunghezza P** byte ed i **P byte di ciascun blocco** vengono anagrammati in base alla permutazione
Per la decifrazione si usa la permutazione inversa

Politiche miste

- *cifrario semplice:* unica trasformazione elementare
- *cifrario composto* o *di prodotto:* una serie di trasformazioni elementari in cascata

Crittoanalisi

Permette di ottenere il testo in chiaro da un testo cifrato senza conoscere la chiave di cifratura
Il crittoanalista deve avere a disposizione della conoscenza

Attacco	Conoscenza a disposizione
con solo testo cifrato (ciphertext only)	Proprietà statistiche del linguaggio in uso e parole probabilmente presenti nel testo in chiaro
con testo in chiaro noto (known plaintext)	Termini sicuramente presenti nel testo in chiaro e testi cifrati successivamente resi in chiaro e di pubblico dominio (coppie <plaintext, ciphertext>)
con testo in chiaro scelto (chosen plaintext)	Testi cifrati corrispondenti a qualsiasi testo in chiaro che si ritenga possa essere utile ai fini della crittanalisi (coppie <scelto plaintext, ciphertext>)
con testo cifrato scelto (chosen ciphertext)	Qualsiasi testo cifrato che si ritenga possa essere utile ai fini della crittanalisi ed il corrispondente testo in chiaro (coppie <plaintext, scelto ciphertext>)

La crittografia moderna

La crittografia moderna usa tre classi di algoritmi:

- a chiave segreta (simmetrici)
- a chiave pubblica (asimmetrici)

- funzioni hash

Un **sistema di crittografia** comprende:

- algoritmo
- chiave

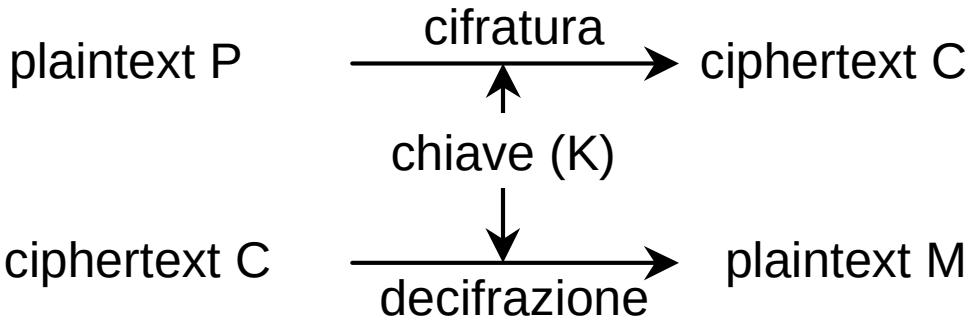
segretezza della chiave

segretezza dell'algoritmo ?

LA SICUREZZA DI UN SISTEMA DI CRITTOGRAFIA (CRITERI)

- L'algoritmo di cifratura noto (**principio di Kerchoffs**)
- Nessun sistema è *assolutamente* sicuro
  (sistemi teoricamente sicuri soluzione *non praticabile*)
- Si deve rendere *praticamente irrealizzabile* l'attacco
  (*tempo di vita delle info*)

La crittografia a chiave segreta (simmetrica)



funzione di cifratura

$$C = E(P, K) = E_K (P)$$

funzione di decifrazione

$$P = D (C, K) = D_K (C)$$

con funzioni di cifratura e decifrazione l'una inversa dell'altra

$$E_K (D_K (C)) == C$$

$$D_K (E_K (P)) == P$$

Impiego dei sistemi a chiave segreta

Trasmissione su un canale insicuro

il messaggio cifrato non rivela l'informazione anche se intercettato

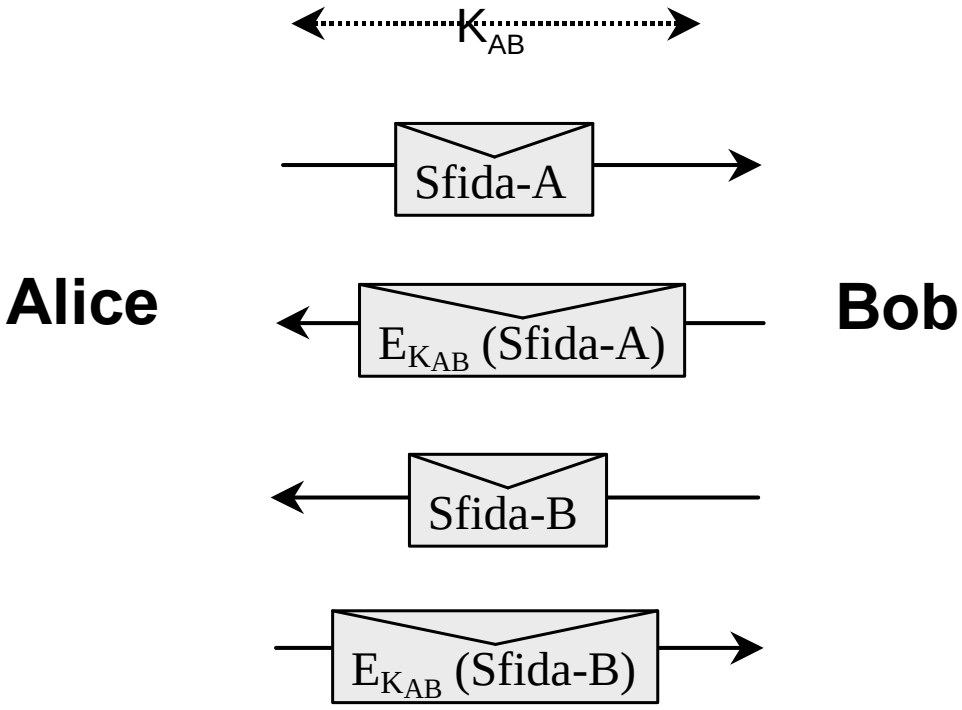
richiede: condivisione chiave tra mittente e destinatario

Immagazzinamento sicuro su un media insicuro

es. cifratura del file system

Autenticazione

es. strong authentication



Integrità dei messaggi

Message Integrity Code (MIC): il checksum cifrato del messaggio

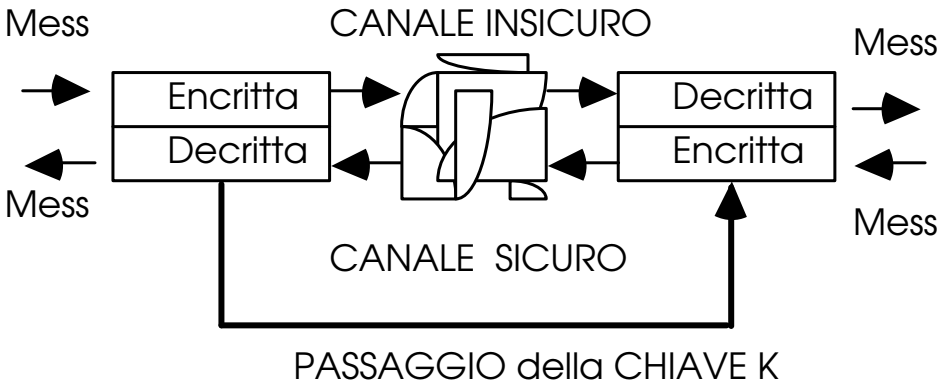
Sistemi a chiave segreta

Sistemi con una chiave unica per cifrare e decifrare (***K***) o con due chiavi diverse per cifrare e decifrare, ma comunque tra di loro collegate (***K_e, K_d***)

Algoritmo di pubblico dominio per la cifratura

IBM Data Encryption Standard (DES)

DES (1977) *cifrario composto* o *di prodotto*
con chiave di 56 bit con blocchi di 64 bit
algoritmo in 19 stage



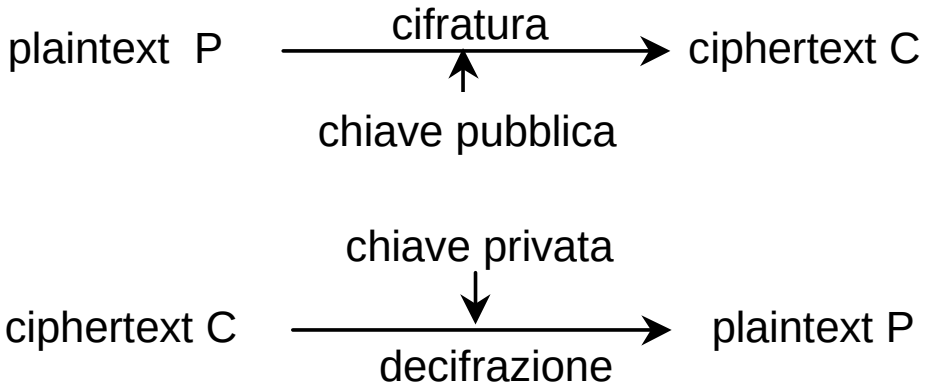
Critiche al DES
insufficiente la lunghezza della chiave (56 bit)
nel progetto originale pare fosse maggiore (128 bit)

IDEA (International Data Encryption Algorithm)
lunghezza di chiave a 128 bit

Possibilità di scorciatoie di decifrazione
(paura del **grande fratello**)

La crittografia a chiave pubblica (asimmetrica)

Usa due chiavi ed è molto difficile dedurre dall'una il valore dell'altra (chiave pubblica e chiave privata)



funzione di cifratura

$$C = E(P, K_{pub}) = Ek_{pub}(P)$$

funzione di decifrazione

$$P = D(C, K_{priv}) = Dk_{priv}(C)$$

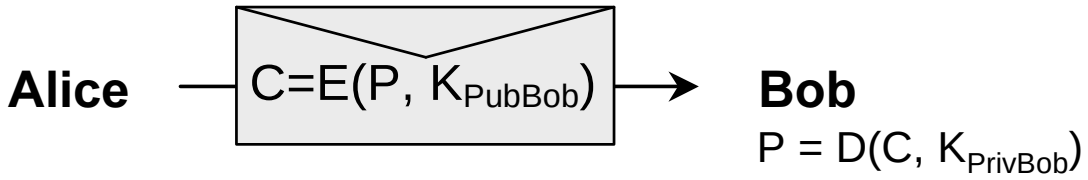
La chiave pubblica è resa di dominio pubblico

Costi computazionali per la cifratura e la decifrazione *molto* superiori al caso simmetrico

Impiego dei sistemi a chiave pubblica

Trasmissione su un canale insicuro

messaggio cifrato con la chiave pubblica del ricevente

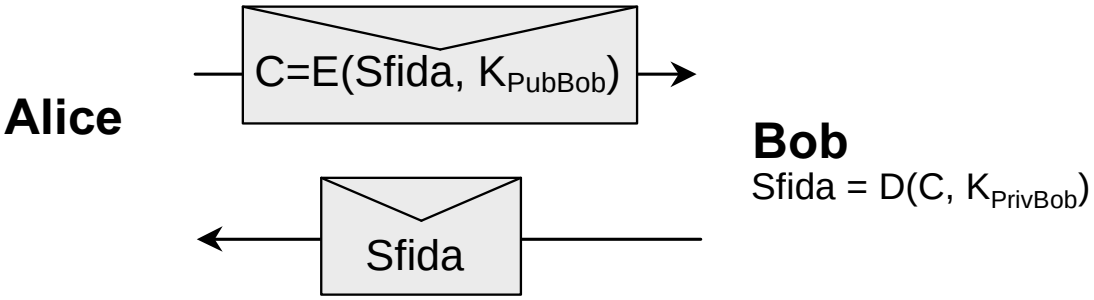


Solo Bob può decifrare C, ma non può sapere con certezza di chi sia il messaggio

Immagazzinamento sicuro su un media insicuro

i dati vengono cifrati con la propria chiave pubblica

Autenticazione



Solo Bob può decifrare C

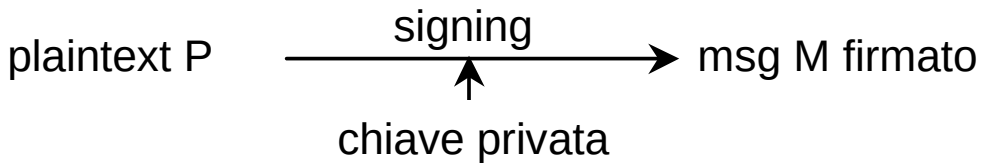
Questi problemi sono risolti anche con la crittografia a chiave segreta, ma nel caso di chiave pubblica:

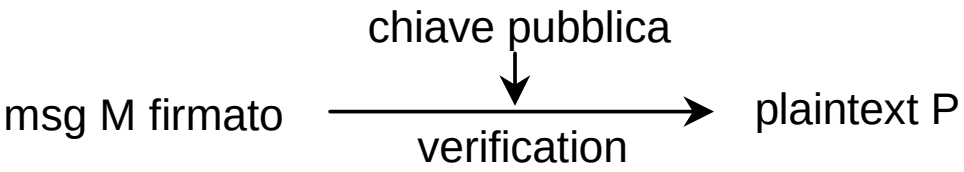
☺ non è richiesta condivisione chiave

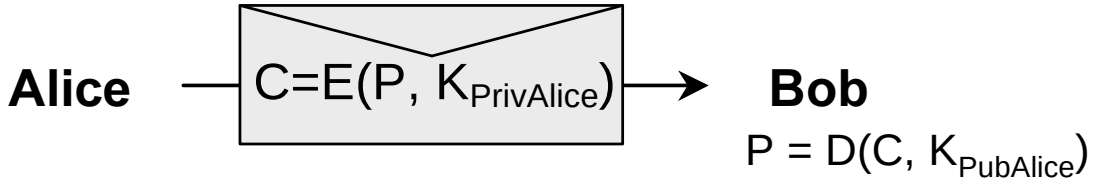
☹ costo computazionale **molto** superiore

Impiego dei sistemi a chiave pubblica

Firma digitale

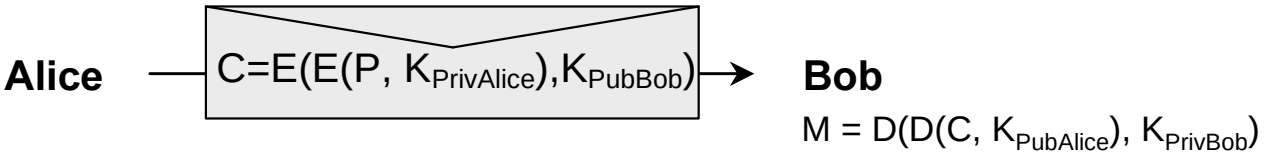






chiunque riconosce la *paternità* di P, perché solo Alice conosce la propria chiave privata

Caso composto



Doppia cifratura
Alice cifra M con la propria chiave privata poi cifra con la chiave pubblica di Bob
solo Bob può leggerlo e provare che arriva da Alice

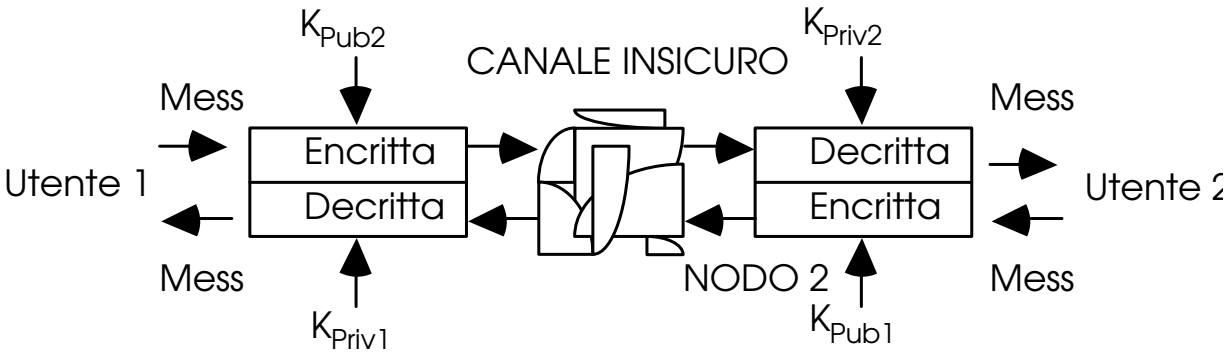
Sistemi a chiave pubblica

(Diffie Hellman 1976)
cifratura separata dalla decifrazione
chiavi diverse

Chiave pubblica **K_{Pub}**
e privata **K_{Priv}**

Invertibilità non facile

Con chiave pubblica e algoritmo non si riesce analiticamente a ricavare la chiave privata



<u>*Funzione unidirezionale*</u> ==> una funzione facilmente computabile ma la cui inversa non può essere computata a meno di informazioni sulla sua costruzione

Algoritmo RSA (Rivest, Shamir, Adelman)

Uso di prodotto di due numeri primi molto alti **p** e **q**
(> 10^{100})

N=p∗q **Z= (p-1)∗(q-1)**

d è un numero primo rispetto a Z

i i ∗ d= 1 (mod Z)
i è quindi il più piccolo elemento della serie Z+1, 2Z+1, 3Z+1, ... divisibile per d

Il messaggio preso a blocchi di **f** bit, con $\mathbf{2^f} < N$
(M per blocco di messaggio e C per forma cifrata)

$\mathbf{K_{pub}}$ == **(N, i)** cifratura ➔ $M= C^{\mathbf{i}} \bmod N$

$\mathbf{K_{priv}}$ == **(N, d)** decifrazione ➔ $C= M^{\mathbf{d}} \bmod N$

e le due chiavi sono inverse

$\mathbf{K_{pub}}$ è nota (N, i) e $\mathbf{K_{priv}}$ (N, d) è la chiave privata

E se il possessore delle chiavi se ne va:
il gestore deve conoscere entrambe le chiavi?

fattorizzazione di un numero di 200 cifre
richiede 4 miliardi di anni
l'operazione per un numero di 500 cifre 10^25 anni

Efficienza

Algoritmi a chiave pubblica (es., RSA) hanno un costo computazionale molto superiore al caso di chiave segreta (es., DES).

In genere si usa un metodo a chiave pubblica per scambiarsi chiavi segrete di sessione per la comunicazione degli utenti (es., SSL)

Inoltre, è fondamentale il **problema della gestione** delle chiavi

Gestore di sicurezza

fornisce le chiavi necessarie e controlla i partner che si devono impegnare in un contratto

problema della revoca delle chiavi

Problema della gestione delle chiavi

Crittografia a chiave pubblica

- una coppia di chiavi per ogni utente
- problema della Certification Authority
- le chiavi pubbliche sono di pubblico dominio
- infrastrutture per chiavi pubbliche (PKI)

Crittografia a chiave segreta

- ogni persona deve condividere una chiave con gli altri

$$N_c = (N_u^2 - N_u)/2$$

- Oppure utilizzo di un key distribution centre (KDC)

Dimensionamento delle chiavi

Attacco di *forza bruta*, cioè provare la decifrazione del messaggio con tutti i possibili valori di chiave

$$T_{max} = \tau * 2^n/N$$

T_{max} = tempo *massimo* richiesto per scoprire la chiave
τ = tempo richiesto per una verifica
n = lunghezza della chiave (numero di bit)
N = Numero di calcolatori in parallelo

Esempio:
DES, chiave a 56 bit $2^{56} \approx 10^{16}$
calcolatore che prova 10^6 chiavi al secondo,
T = 2000 anni
Macchina a parallelismo massiccio, costo 1 milione $
T = 3,5 ore
Macchina a parallelismo massiccio, costo 10 milioni $
T = 21 minuti

DES esportato fuori da USA a 40 bit

IDEA a 128 bit

Clipper

Riservatezza della comunicazione
Vs
possibilità intercettazioni telefoniche

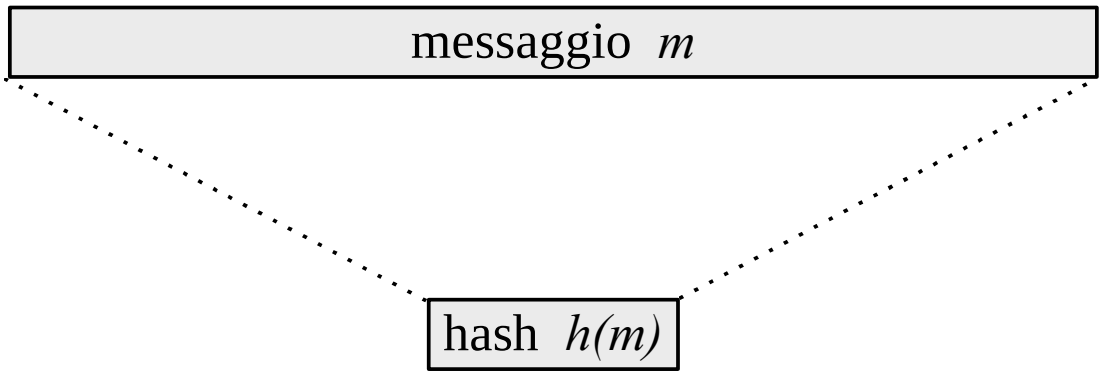
Clipper è un chip per la crittografia delle comunicazioni, l'algoritmo di crittografia è segreto (SKIPJACK) e usa una chiave segreta per ogni chip (utente)

Il possesso della chiave di un utente permette di decifrare le sue conversazioni

Negli USA la chiave è spezzata in due parti, ciascuna parte è custodita da un Dipartimento diverso e si concede la decifrazione solo su motivi gravi e fondati (problema della custodia delle chiavi, key escrow)

Funzioni Hash

Una funzione hash comprime un input *m* di lunghezza arbitraria in un output *h(m)* di lunghezza fissa e di piccole dimensioni



Caratteristiche funzione hash:

- per ogni messaggio *m* è facile calcolare *h(m)*
- dato *h(m)* è difficile trovare un *m* che lo fornisca
- deve essere difficile trovare due *m* con lo stesso *h(m)*

Impiego degli algoritmi Hash

Password hashing

password cifrata registrata su file
hash della password cifrata su file

Integrità dei messaggi

hash usata per generare MIC del messaggio
hash (messaggio+password)

Impronta dei file

MIC di un file per accertarsi della sua integrità

Efficienza delle firme digitali

si firma l’hash del messaggio

Kerberos

Kerberos è un protocollo per l'**autenticazione** in un sistema distribuito **cliente/servitore**

Kerberos fa parte del progetto **Athena MIT**
(e adottato anche in DCE e Andrew)

Assunzioni sull'ambiente:

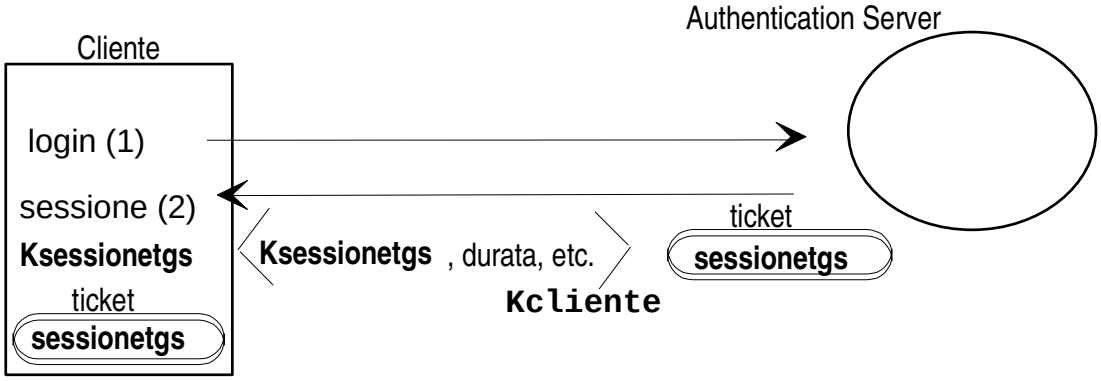
- utenti non fidati
- rete non sicura

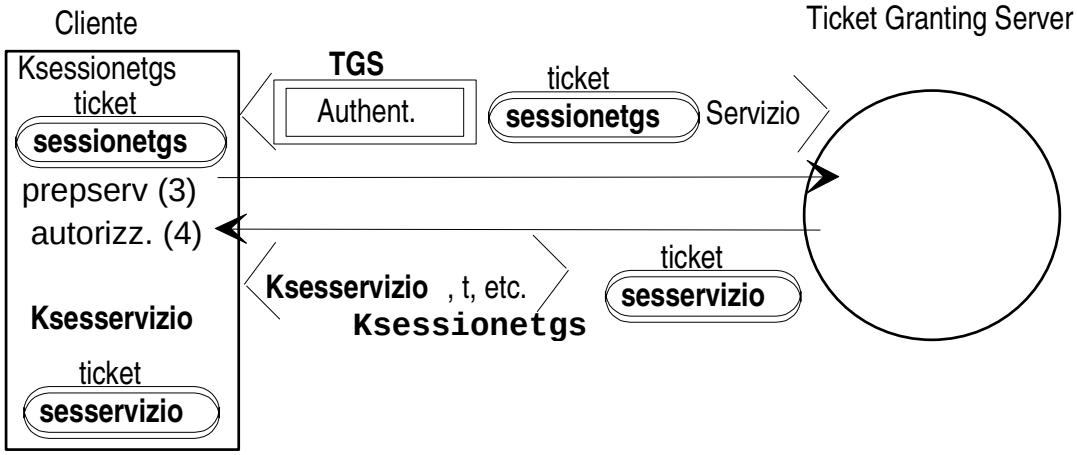
Kerberos usa chiavi segrete
Validità temporale limitata delle operazioni

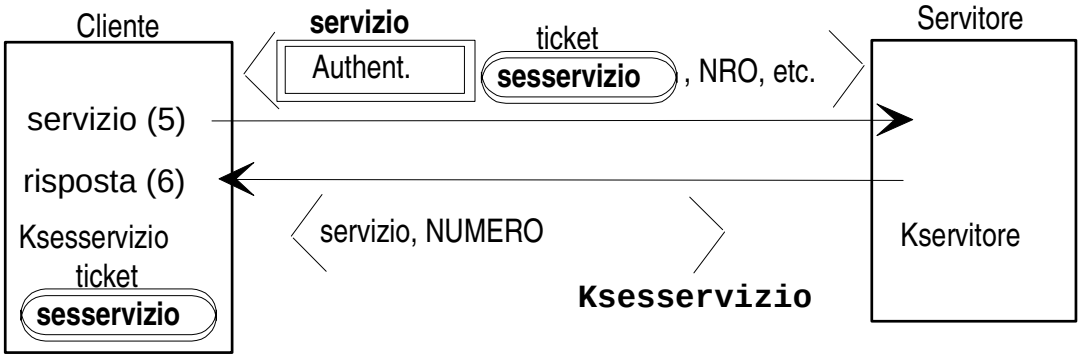
Servizi differenziati
- per il sistema:
  servizio di Autenticazione
  servizio di Ticket
- per ogni servizio:
  gestione dell'autenticazione

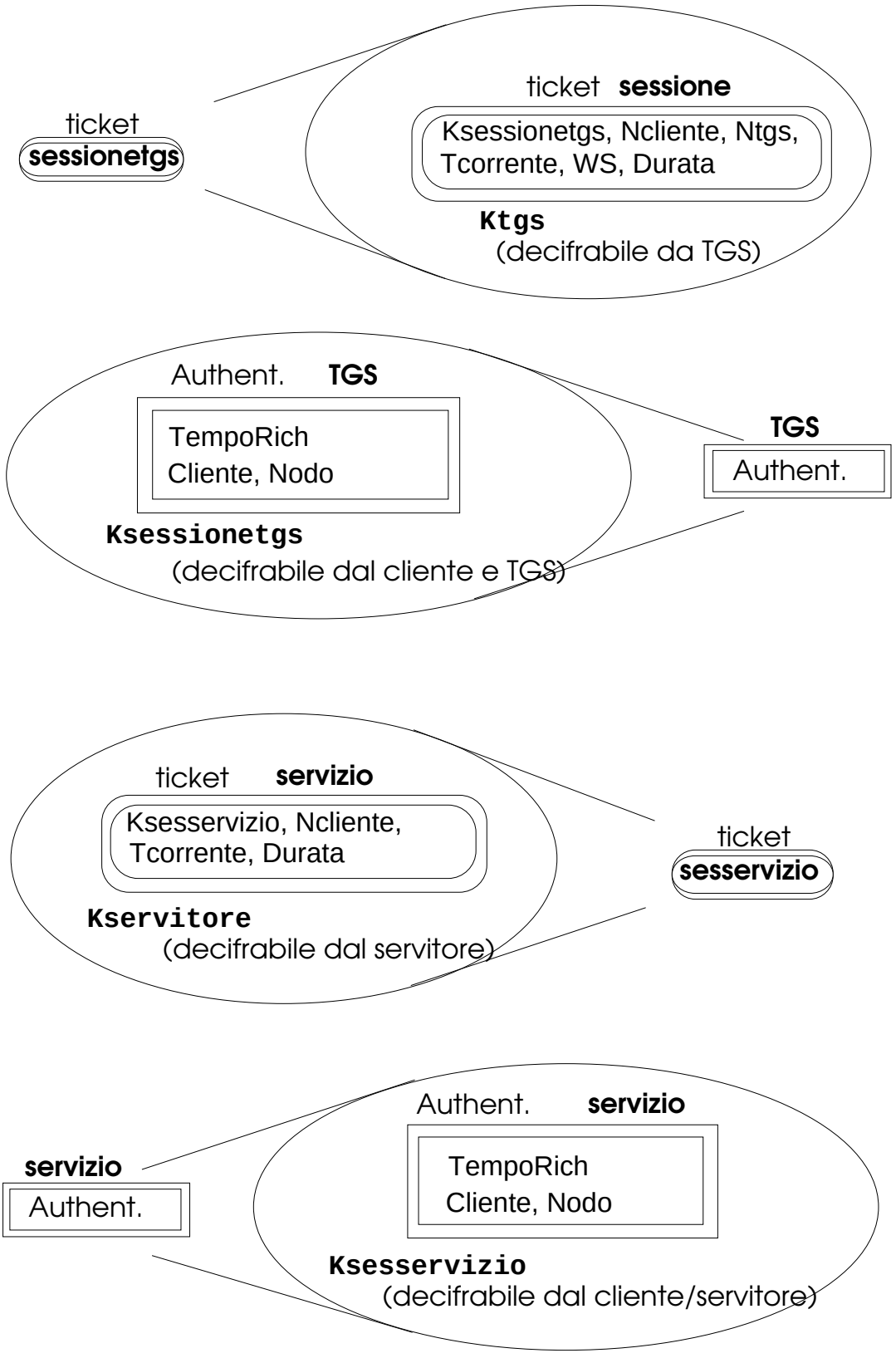
Passi ed entità
- **A**uthentication **S**erver **(AS)**
- **T**icket **G**ranting **S**erver **(TGS)**
- Ogni servitore deve essere noto al gestore dell'autenticazione

Circolano **token** diversi
- ***ticket***, dal TGS al cliente
- ***authenticator***, dal cliente al servitore
  con informazioni di tempo e usato una volta sola
- ***chiavi di sessione***, dal cliente al servitore


Authentication Server
Cliente
login (1)
sessione (2)
Ksessionetgs
ticket
sessionetgs
Ksessionetgs , durata, etc.
Kcliente
ticket
sessionetgs


Cliente
Ticket Granting Server
Ksessionetgs
ticket
sessionetgs
TGS
Authent.
ticket
sessionetgs
Servizio
prepserv (3)
autorizz. (4)
Ksesservizio
ticket
sesservizio
Ksesservizio , t, etc.
Ksessionetgs
ticket
sesservizio


Cliente
servizio
Authent.
ticket
sesservizio
, NRO, etc.
Servitore
servizio (5)
risposta (6)
Ksesservizio
ticket
sesservizio
servizio, NUMERO
Ksesservizio
Kservitore


ticket
sessionetgs
ticket sessione
Ksessionetgs, Ncliente, Ntgs,
Tcorrente, WS, Durata
Ktgs
(decifrabile da TGS)
Authent. TGS
TempoRich
Cliente, Nodo
Ksessionetgs
(decifrabile dal cliente e TGS)
TGS
Authent.
ticket servizio
Ksesservizio, Ncliente,
Tcorrente, Durata
Kservitore
(decifrabile dal servitore)
ticket
sesservizio
servizio
Authent.
Authent. servizio
TempoRich
Cliente, Nodo
Ksesservizio
(decifrabile dal cliente/servitore)

FASI

1) Cliente interagisce con **Authentication Server**

Cliente al *Login* inserisce lo *Username*
spedito in chiaro all'Authentication Server
con *Timestamp* con data ed ora

(1) Cliente ---->AS

$$\{N_{cliente}, N_{tgs}, T_{corrente}\}$$

I valori interi sono identificatori per l'unicità

2) TGS crea il ticket e lo restituisce al **Cliente**

Il **T**icket **G**ranting **S**ervice localizza la password corrispondente allo Username e crea per la sessione il $ticket_{sessionetgs}$

$$\{K_{sessionetgs}, N_{cliente}, T_{corrente}, WS, Lifetime\}^{Ktgs}$$

WS indirizzo della workstation richiedente

messaggio alla Workstation del Cliente

(2) AS (TGS) ---->Cliente

$$Ticket_{sessionetgs}\ , \{K_{sessionetgs}, N_{tgs}, Lifetime, T_{corrente}\}^{Kcliente}$$

l'utente inserisce la password
confrontata e cancellata subito dopo
Si verificano $T_{corrente}$ e N_{tgs} ==>
il Cliente è certo che arriva dal sistema di gestione
ed è l'unico a poterlo decifrare
memorizzazione di $K_{sessionetgs}$ e $Ticket_{tsessionegs}$ per dopo
Client e Authentication Server in reciproca fiducia

SERVIZIO VERO E PROPRIO

3) Cliente prima di una richiesta a un Servitore manda una richiesta al TGS (Ticket Granting Server)

quando il Client ha bisogno di un servizio
prepara un Authenticator

$$\mathbf{Authenticator_{tgs} => \{N_{cliente}, T_{corrente}, WS\}^{Ksessionetgs}}$$

ed spedisce al TGS

(3) Cliente ---->TGS

$$\mathbf{\{Authenticator_{tgs}, Ticket_{sessionetgs}, N_{servizio}\}}$$

Ticket-Granting Service decifra il Ticket
Se OK, decifra l'Authenticator mediante la chiave di Sessione cioé $K_{sessionetgs}$

4) TGS per il Cliente

Se tutto è regolare, TGS prepara il $\mathbf{Ticket_{sesservizio}}$

$$\mathbf{\{K_{sesservizio}, N_{cliente}, T_{corrente}, Lifetime\}^{Kservitore}}$$

$K_{sesservizio}$ una **Chiave di Servizio** per la sessione di comunicazioni tra Client e Servitore

(4) TGS ---->Cliente

$$\mathbf{Ticket_{sesservizio}, \{K_{sesservizio}, N_{servizio}, T_{corrente}\}^{Ksessionetgs}}$$

Il Client, possiede Ksessionetgs, decifra il pacchetto, ne verifica l'autenticità e memorizza $K_{sesservizio}$ e $Ticket_{sesservizio}$

5) Cliente al Servitore
il Cliente costruisce l'Authenticator$_{servizio}$

$$\mathbf{\{N_{cliente}, T_{corrente}, WS\}^{Ksesservizio}}$$

Il Cliente manda al Servitore

(5) Cliente ---->Servitore

$$\mathbf{\{Authenticator_{servizio}, Ticket_{sesservizio}\}}$$ **ed altro**

Il Servitore decifra il Ticket con chiave privata $K_{servitore}$
Se ci riesce e se Ticket/Authenticator non sono "scaduti", il Ticket proviene da *Kerberos*

Dopo una disconnessione, una riconnessione al Servitore può usare lo stesso Ticket
Authenticator deve essere ricreato ogni volta (Timestamp aggiornato)

6) Mutual Authentication
Garanzia che il Servitore sia quello corretto
attraverso una prova al Cliente

Messaggio di risposta del Servitore al Cliente

$$\mathbf{\{N_{servizio}, T_{corrente}+1, Server_{address}\}^{Ksesservizio}}$$

Il valore dimostra la autenticità del Servitore

Kerberos: considerazioni conclusive

Kerberos usa chiavi segrete

Kerberos sviluppato per un ambiente con:

- utenti non fidati
- rete non sicura

Le password degli utenti NON viaggiano sulla rete e sono note a:

- utenti
- AS del KDC

Chiavi segrete dei servizi sono note a:

- servitori
- TGS del KDC

Per assicurare una **validità temporale limitata** delle operazioni richiede una loose synchronization tra i nodi.

Replicazione KDC

Partizionamento rete in “realm”
(relazioni di fiducia tra diversi realm)

Problemi con NFS:
NFS è stateless ➔ autenticazione ogni richiesta costoso ➔ si autentica solo il mount

Intranet e Internet

Internet rete intrinsecamente insicura

Collegamento di Intranet aziendali ad Internet è un problema molto sentito, soprattutto per:

- economicità (*basso costo collegamenti*)
- mercato globale (*WWW*)
- supporto mobilità utenti

Principio fondamentale

se si vuole essere sicuri, è meglio non essere connessi

Possibilità
di accedere ai servizi di rete
accesso dall'interno verso l'esterno
accesso dall'esterno verso l'interno
senza compromettere il sistema interno

Per le **organizzazioni commerciali** o **bancarie**
diventa vitale trovare soluzioni accettabili

*A parte la **disconnessione***
possibili politiche e meccanismi di **separazione**
tra gli **ambienti** eventualmente **sicuri interni (*Intranet*)** e ***Internet*** *(uso di sistemi firewall e VPN)*

Fattori di perdita di sicurezza di Internet

TCP/IP come sistema aperto
- *uso di risorse esterne per routing*
- *vulnerabilità intrinseche dei servizi e protocolli*
- *estrema complessità di meccanismi di controllo*

Facilità di monitoraggio dell'attività di rete:
- *comunicazioni in chiaro*

Controllo degli accessi e autenticazione utenti
- *basati su password (statiche e riusabili)*

Connessioni di rete tramite risorse esterne
- *linee condivise e router di terzi*

Firewall

Un **firewall** garantisce la sicurezza del collegamento di una Intranet verso Internet.

Un firewall è un sistema costituito di molti componenti che:

- costituisce **l'unico punto di contatto** della rete **interna** con l'**esterna**
- filtra e **controlla** tutto il traffico tra le due reti.
- concentra i **meccanismi di sicurezza**
- impone la **politica di sicurezza** della organizzazione
- nasconde informazioni della rete interna
- registra eventi (**logging)** ed elabora statistiche sul traffico di rete (**auditing**)

Separazione politiche-meccanismi

Attenzione alla scelta dei servizi che devono transitare attraverso il firewall.

POLITICHE DI UN FIREWALL

il **firewall** deve implementare una **politica di accesso** in modo **separato** e **concentrato**

===>

☹ senza firewall le stesse funzioni vengono ottenute attraverso la cooperazione di tutti gli host

Problemi

un firewall non risolve tutti i problemi

☹ un **firewall** restringe la possibilità di accesso a servizi
☹ la topologia di rete può essere inadeguata a un firewall
☹ non protegge contro i ***virus***
☹ problemi di ***attacchi interni***
==> *giusto compromesso tra sicurezza e funzionalità*

😐 attenzione alle vie di accesso secondarie (**backdoor**)
==> *accesso tramite modem*

☹ come punto **concentrato** di affidabilità
ma può diventare un **collo di bottiglia** (bottleneck)

Aspetti progettuali da considerare in un firewall

- esigenze da soddisfare (quali servizi)
- architettura
- autenticazione
- politica di autorizzazione di rete

Metodologie di autenticazione

tecniche di autenticazioni robuste
anche basate su meccanismi differenziati

*password usa e getta (***one-time password***)*
carta magnetica
impronta digitale

==> caratteristica comune

utilizzo di password non riusabili

Politiche di autorizzazione opposte

tutto ciò che non è espressamente permesso è vietato

- maggiore sicurezza
- più difficile da gestire

tutto ciò che non è espressamente vietato è permesso

- minor sicurezza
- più facile da gestire

Problemi di efficienza

Il firewall comporta una inefficienza nei servizi che sono disponibili e un ritardo nei tempi di risposta

il firewall *potrebbe anche diventare il collo di bottiglia dell'intero sistema*

Considerazioni generali

- grossi oggetti sono difficili da verificare
  ('grande non è bello')
- se una risorsa dinamica non è attiva non preoccupa
- definire assunzioni di default => tutti sono sospettati
- usare risorse dedicate solo ai meccanismi di sicurezza

Esempi di configurazioni di sistemi firewall

- grado di controllo sugli errori e buchi di sicurezza
- zona di rischio (# di host esposti a possibili attacchi)
- politica di autorizzazione adottata

Router *un gateway che separa la rete interna dalla esterna*

Dual-Home gateway *macchina sicura con due accessi separati alla rete*

Bastion host *macchina sicura dedicata al controllo del sistema per la sicurezza* ==> ***auditing*** ossia verifica e traccia degli eventi nel sistema

Packet filtering (filtraggio livello rete)

il traffico filtrato sulla base dei campi contenuti nel datagramma IP

sourceIP *sourcePORT*
destinationIP *destinationPORT*

Si possono così
escludere alcuni host come mittenti o come destinatari
escludere alcuni servizi

azioni specificate a mano

tipo	Indirizzo destination	Indirizzo source	Porta destin.	Porta source	Azione
TCP	137.204.57.33	*	23	>1023	permit
TCP	137.204.57.32	*	25	>1023	permit
TCP	137.204.57.34	*	25	>1023	permit
TCP	137.204.57.31	137.2.5.30	119	>1023	permit
TCP	*	*	*	*	deny

vedi il file ***/etc/services***

problemi

- difficoltà in caso di
  - ☹ servizi RPC
  - ☹ più interfacce
- difficoltà di specificare in modo compatto regole
- mancanza di logging

Application gateway

i problemi del **packet filtering** si possono superare con proxy

cioé gestori ad-hoc per consentire il trattamento solo di uno specifico servizio

proxy server

applicazione software col compito di mediare il traffico tra rete esterna ed interna e consentire accesso a un servizio specifico

vantaggi

- ☺ filtra *servizi* e *protocolli*
- ☺ supporta *autenticazione* robusta e *logging*
- ☺ semplifica le regole del *filtering*
- ☺ garantisce riservatezza alla rete interna
- ☺ incide positivamente sul costo
  - i proxy devono essere concentrati sul solo firewall e non distribuiti su tutti gli host della rete

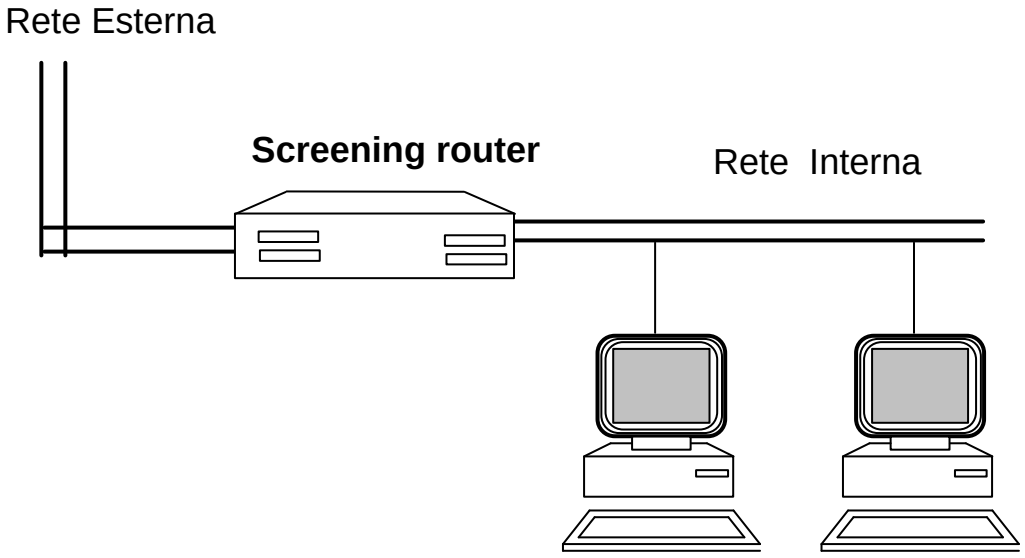
svantaggi

- ☹ connessioni con host interno a due passi
  perdita di trasparenza del firewall
  a meno di modificare i clienti per i servizi di rete più comuni

Screening router

*un **router** fa da filtro tra le **due reti** (interna ed esterna)*

- questo firewall usa il router per filtrare il traffico
- non necessita di proxy
- implementare la politica
  tutto ciò che non è espressamente permesso è proibito



problemi

☹ basso livello di sicurezza introdotto
ogni host necessita
di robuste misure di autenticazione==>
la zona di rischio è pari al numero di host della rete

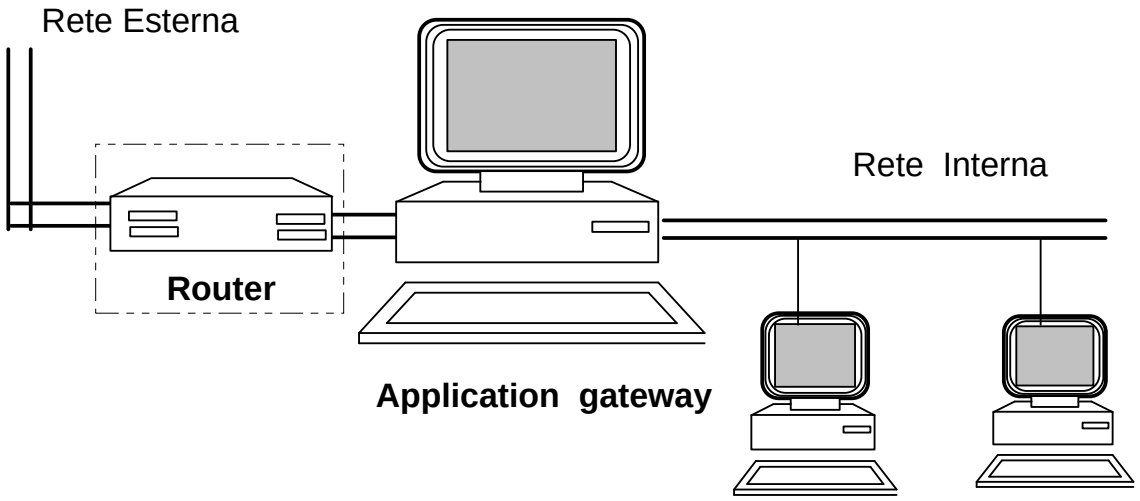
☹ regole di packet-filtering difficili da specificare sul router

☹ mancanza di logging

Dual homed gateway

stazione *dotata di due* ***interfacce*** *di rete con effettiva separazione fisica tra rete interna ed esterna*

a volte si aggiunge anche un ***router*** sulla connessione esterna per packet-filtering



☺ alto livello di privacy
☺ misure robuste di autenticazione
☺ logging facile

😐 implementa politica di accesso più rigida
😐 proxy per servizi standard: **telnet, ftp, e-mail**

☹ mancanza di flessibilità in caso di modifiche di servizi e sovraccarico di lavoro concentrato sul gateway

☹ gateway stesso come *zona di rischio* e come *collo di bottiglia*

Screened host firewall

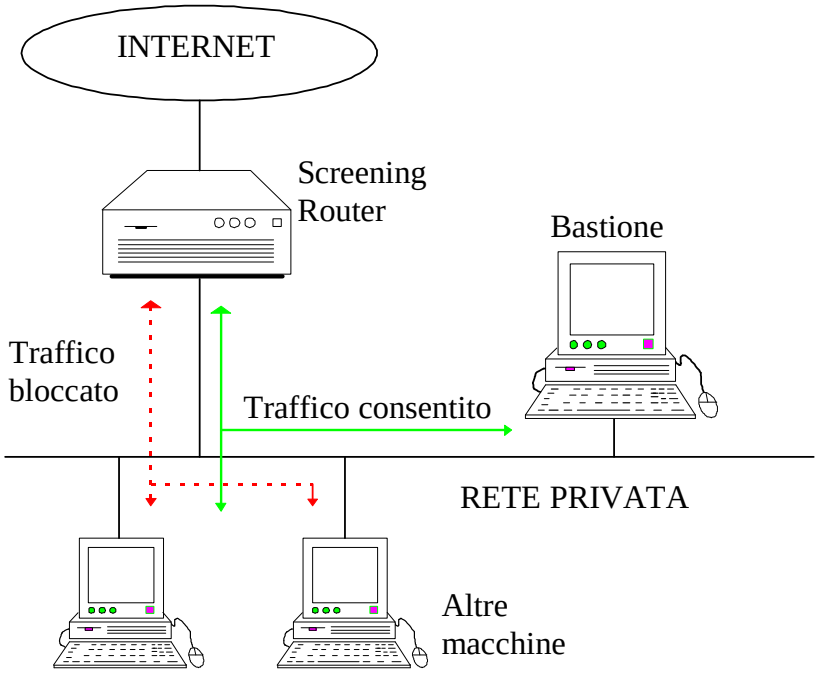
*costituito da un **bastion host** e da **una rete interna***

application-gateway sul bastione

==> si affaccia direttamente sulla rete interna e passa le informazioni all'interno

router

==> blocca i pacchetti dall'esterno/interno tranne quelli in arrivo/invio da application gateway



- flessibilità maggiore rispetto al dual-homed
- allenta il controllo su certi servizi/host

implementa entrambe le politiche di autorizzazione

problemi

☹ costo della soluzione

☹ la rete interna non presenta ulteriori barriere di protezione a parte il gateway,

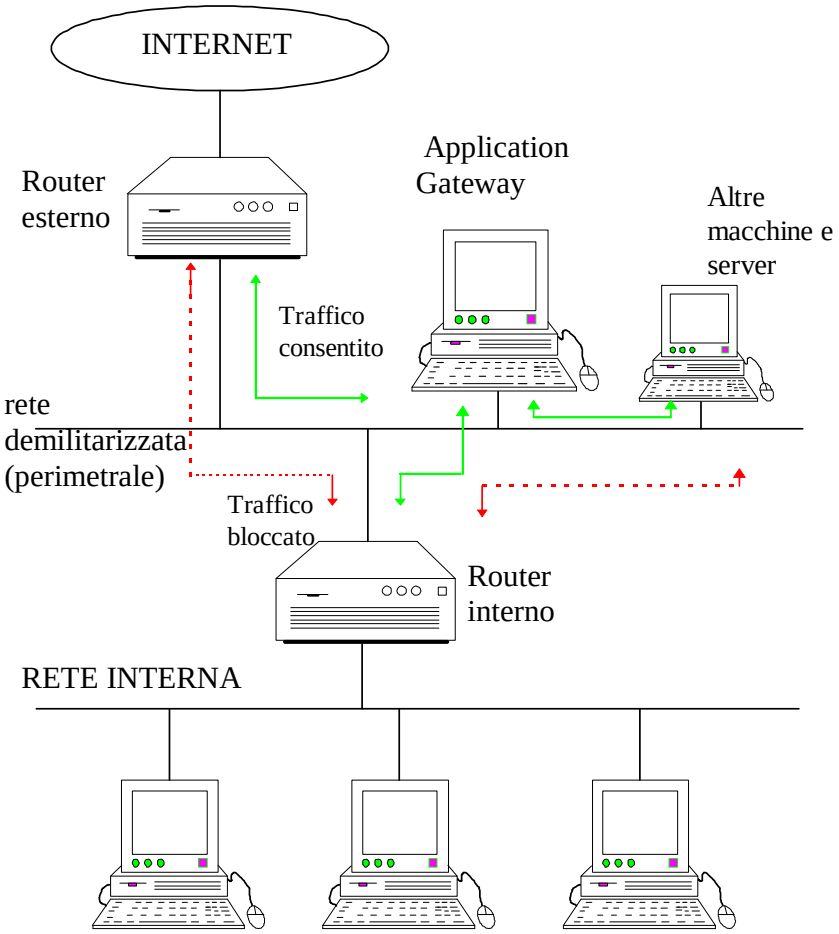
Screened Subnet Firewall

router per esterno

- inoltra traffico dall'esterno agli **application gateway**
  server e-mail e information server (amche host diversi)
- inoltra traffico dagli application gateway all'esterno
- *altro traffico rifiutato*

router per interno

- inoltra traffico dagli application gateway all'interno
- inoltra traffico ftp, gopher dal'interno agli information server della rete demilitarizzata
- *altro traffico rifiutato*



vantaggi:

☺ non c'è alcun accesso al **sistema interno**

☺ si garantisce elevato throughput considerando due router con intrinseca **ridondanza**

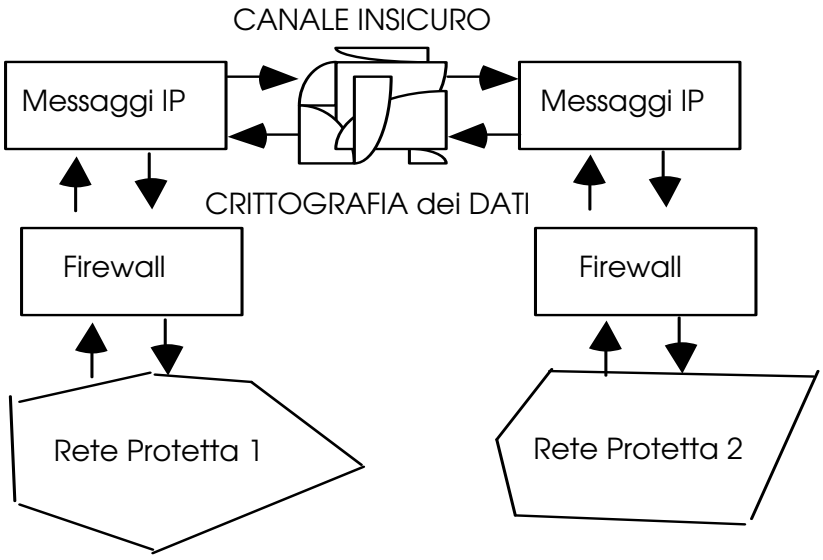
☺ anche **autenticazione avanzata** sugli application gateway

☹ gestione più complessa delle risorse

IP tunneling

Tecnologia con cui un pacchetto di un qualunque protocollo viene incapsulato in un datagramma IP.

Esempio: i pacchetti NetBeui incapsulati in un datagramma IP possono muoversi su Internet.



possibile applicazione della crittografia al sistema, con **chiavi note** solo all'interno dei due sistemi protetti

IPsec (secure IP)

Protocollo IP sicuro, fornisce la cifratura a livello IP, più in basso di SSL o VPN.

VPN (Virtual Private Network)

Una VPN realizza una Intranet privata virtuale al di sopra di una rete pubblica (Internet).

Macchine di sottoreti diverse all'interno di una stessa organizzazione possono cooperare direttamente.

Integrazione di diversi Firewall e di macchine mobili.

Vantaggi VPN:

- Trasparenza per utenti
- Supporto alla mobilità utenti
- Economicità del collegamento

Tecnologie:

PPTP (Point to Point Tunneling Protocol) tipicamente collegato al RAS (Remote Access Services) di Win NT (che esegue autenticazione e cifratura).

Altavista tunnel della Digital.

Cisco PIX Firewall, soluzione HW, veloce ed efficiente ma scarso supporto utenti mobili.

Riferimenti su Internet Security

(da http://www-lia.deis.unibo.it/Staff/CesareStefanelli/Security.htm)

Libri

Applied Cryptography - Protocols, Algorithms and Aource code in C, Bruce Schneier, John Wiley & Sons, 1995
(schneier@counterpane.com www.counterpane.com)
Network Security, Kaufman, Perlman, Speciner, Prentice Hall, 1995.
Security in Computing, C. Pfleeger, Prentice Hall
Practical Unix Security, Garfinkel, Spafford, O'Reilly
Java Security, S. Oaks, O'Reilly, 1998.
Virtual Private Network, C. Scott, P. Wolfe, M. Erwin, O'Reilly, 1998.
"Computer Communications Security: Principles, standards, protocols and techniques" di Warwick Ford, Prentice Hall
"Handbook of applied cryptography" di A.J.Menezes, P.C.van Oorschot, S.A.Vanstone, CRC Press
Firewalls and Internet Security, Cheswick, Bellovin, Addison Wesley
Building Internet Firewalls, Chapman, Zwicky, O'Reilly
"Protecting your Web Site with Firewalls" di Marcus Goncalves, Prentice Hall
Trusted Computer System Evaluation Criteria. DOD 5200.28-STD, National Computer Security Center, December 1985. (Orange Book)
Trusted Database Management System Interpretation. NCSC-TG 021, April 1991. (Lavender Book)
Trusted Network Interpretation. NCSC-TG 005, National Computer Security Center, August 1990. (Red Book)
Information Technology Security Evaluation Criteria (ITSEC). Department of Trade and Industry, London, June 1991. Harmonized Criteria of France, Germany, the Netherlands, and the United Kingdom.

Site

www.cert.org : CERT (computer emergency response team) charter is to work with the Internet community to facilitate its response to computer security events involving Internet hosts
www.crypto.com cifrare o non cifrare?
http://www.zurich.ibm.com/Technology/Security/ IBM
http://www.zurich.ibm.com/Technology/Security/extern/internet/white-paper.html
http://www.nist.gov/ NIST (National Institute of Standards and Technology)
http://www.itl.nist.gov/div893/ NIST, Computer Security Division

www.swcp.com/~iacr/proceedings/alldata_byname.html (IACR conference proceedings, by author name);
www.swcp.com/~iacr/jofc/jofc.html (Journal of cryptology bibliography and table of contents, from IACR)
http://web.mit.edu/security/www/iso1.html
news: alt.security
news: sci.crypt

University sites:
thery.lcs.mit.edu/~rivest/crypto.bib (Ron Rivest's Crypto and Security bibliography)
Cambridge www.cl.cam.ac.uk/Research/Security
Purdue COAST project www.cs.purdue.edu/coast/coast.html
Carnegie Mellon www.ini.cmu.edu/netbill
Ross Anderson www.cl.cam.uk/users/rja14
Carl Ellison (www.clark.net/pub/cme/home.html)

Algoritmi crittografici
www.rsa.com RSA Data Security, Inc.
www.cs.berkeley.edu/~daw/ David Wagner, collabora con Schneier, ha messo su web molti lavori.
Su Quantum cryptography si veda http://www-dse.doc.ic.ac.uk/~nd/surprise_97/index.html e i lavori di Gilles Brassard
http: //www.cs.hut.fi/ssh/crypto
http: //www.ifi.uio.no/pgp

Protocolli crittografici
www.rsa.com per le PKCS (Public-Key Cryptography Standards)
Certification Authorities, PKI (Pubblic Key Infrastructure), etc.
www.entrust.com/library.htm (contiene white paper sia sul prodotto specifico sia di carattere generale sulla gestione
della fiducia in Internet; contiene anche tutti gli IETF draft relativi alle PKI)
www.public-key.com
http://www.valicert.com/
www.xcert.com Esempio di CA
www.steinroe.com Esempio di CA

Sistemi Firewall
www.tis.com (Trusted Information System)

www.data.com (sito con rsultati di test di performance sui firewall commerciali)
www.clark.net/pub/mjr (pagina web di Ranum con tutti i suoi articoli)
Fred Avolio’s entire set of slides for his talk on "Securing the Perimeter"
http://www.tis.com/docs/products/gauntlet/fwovervw/index.htm

Internet Mail
www.imc.org Internet Mail Consortium, Informazioni su RFC e Internet Draft relativi alla posta elettronica.

Commercio Elettronico
http: //www.digicash.com/
www.forrester.com Informazioni e stime relative al commercio elettronico
Gail Grant, "Understanding Digital signatures: Establishing Trst over the Internet and other networks"
http://www.betabooks.mcgraw-hill.com/grant/